14,- DM Ös 120,- sfr 14,-

...ende Schneider-Magazin

# SONDERHEFT Nr.: 5

# CPC Schneider INTERNATIONAL

Sonderheft

5/87

## Für Vortex-Anwender:
DOS-Erweiterung
Disktool

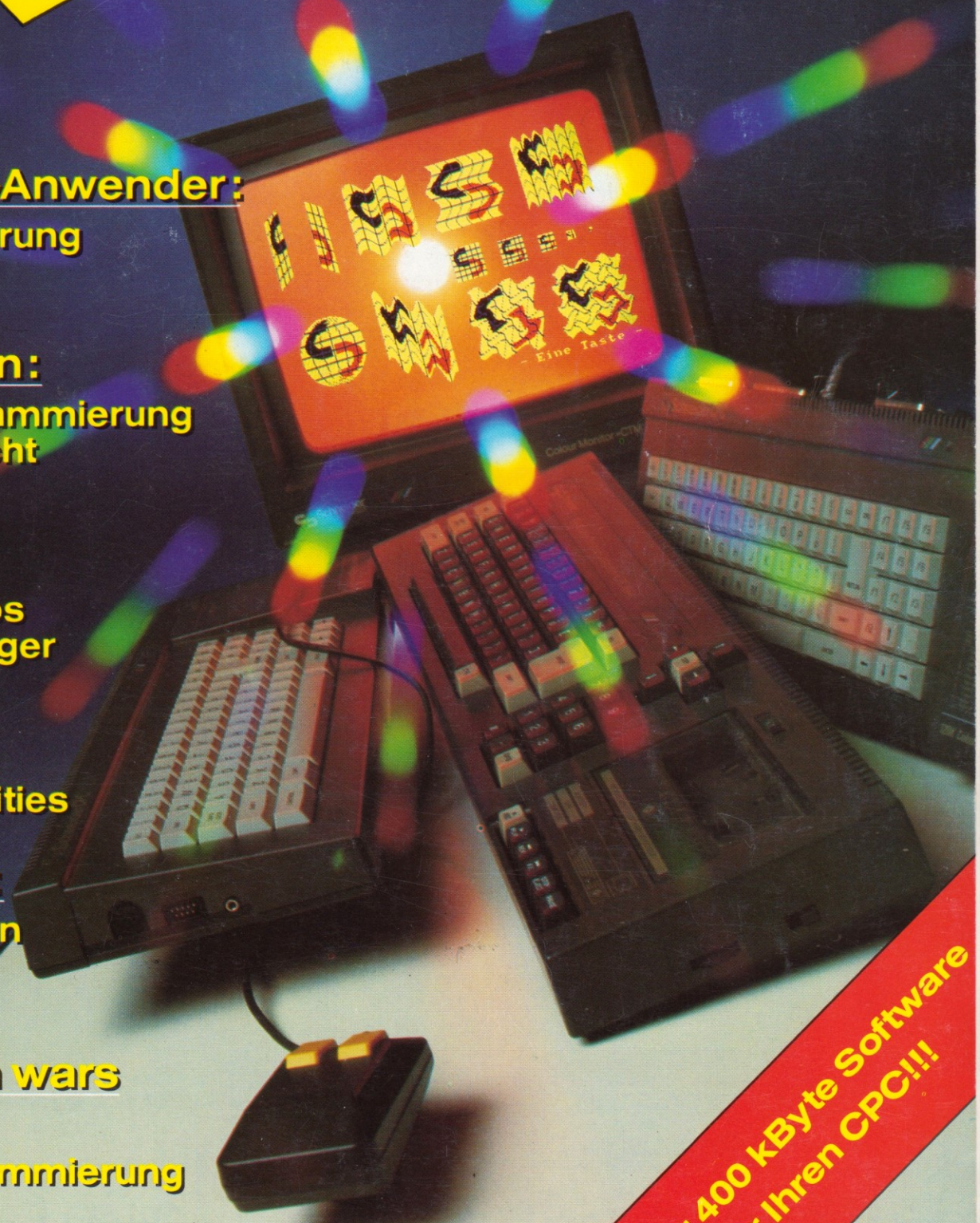
## Grundlagen:
Soundprogrammierung
leicht gemacht

## Spiele:
Crazy Stamps
Tennis Manager

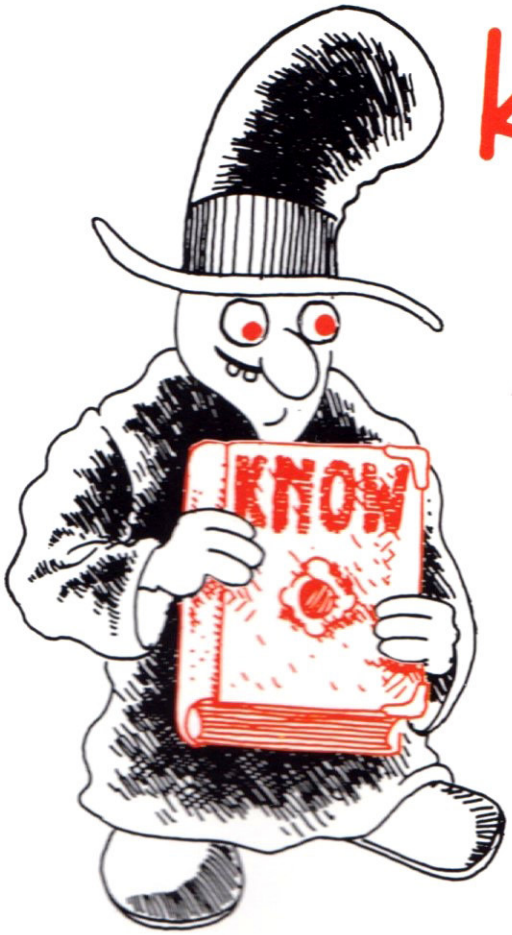
## Übersicht:
Diskettenutilities

## Spieletips:
Lösungshilfen
Lagepläne

## So einfach wars noch nie:
Grafikprogrammierung
mit GBasic




Über 400 kByte Software
für Ihren CPC!!!




# KNOW

# Wissen ist Macht...

- kann eine **unbegrenzte** Anzahl von Fragen und Antworten verwalten!
- hat ständig ca. **400** Fragen **plus** Antworten im Speicher!
- bietet die Möglichkeit **eigene** Fragen einzugeben, und zwar mit einem **komfortablen** Editor!
- besitzt eine **Supergrafik** mit Window-Technik!
- ist vollkommen **menuegesteuert**
- zeigt bis jetzt noch nie dagewesene Tricks mit dem **Videocontroller!**
- stellt zu jeder Frage **5** mögliche Antworten vor!
- kann man mit der **ganzen Familie** spielen, da jeder eine Mindest-Chance von 1–5 hat!
- muß ma mit **Strategie** spielen, da der Beste in jedem Wissensgebiet am Ende noch Zusatzpunkte bekommt!
- wurde mit äußerst schnellen **Suchroutinen** ausgestattet, die verhindern, daß eine schon richtig beantwortete Frage nochmals erscheint!
- ist spielbar mit **1–4** Einzelspielern oder in Gruppen mit einem Vielfachen davon!
- erkennt **automatisch,** welche Fragenblöcke auf der Diskette noch unbeantwortet sind!
- zeichnet sich nicht nur durch die o.g. Punkte als höchst **zukunftssicher** aus!




# KNOW

Das Spiel für die ganze Familie

**– ist das Spiel überhaupt**




für Schneider
CPC 464 · 664 · 6128

| | |
|---|---|
| Kassettenversion | 24,– DM |
| Diskette 3" | 29,– DM |

DMV Software

direkt zu beziehen von:

DMV-Verlag · Postfach 250 · 3440 Eschwege



# Darfs ein bißchen mehr sein?

*Liebe Leser,*

*die Reaktion auf unser letztes Sonderheft war gigantisch. An dieser Stelle sei all denen gedankt, die uns ihre Anregungen mitgeteilt und mit Lob, aber auch Kritik nicht hinterm Berg gehalten haben.*

*Aufgrund der großartigen Resonanz war die Konzeption dieses Sonderheftes eigentlich mühelos, lediglich die Auswahl der einzelnen Beiträge mußten wir mit einem lachenden und einem weinenden Auge vornehmen.*

*Zu viele, wirklich hervorragende Beiträge mußten aufgrund Platzmangels auf Eis gelegt werden, doch keine Angst: Nach und nach werden wir in der PC Schneider Int. diese Projekte aufgreifen.*

*Bei der Lektüre dieses Sonderheftes werden Sie feststellen, daß es uns wieder einmal gelungen ist, eine abwechslungsreiche Mischung diverser Themengebiete zusammenzustellen und Ihnen originelle Spiel- als auch leistungsfähige Anwenderprogramme anbieten können.*

*Daß das ein oder andere Programm den üblichen Listingumfang sprengt, sollte Sie erst recht zum Abtippen verleiten – es lohnt sich! Natürlich sind auch diesmal alle Programme wieder auf Datenträger erhältlich – unser Databox-Service bietet zwei beidseitig (!) bespielte Kassetten bzw. Disketten an.*

*Insgesamt erhalten Sie genügend Stoff für die bevorstehende Jahreswende, in Kilobyte ausgedrückt beinhaltet unser 5. Sonderheft ca. 500 KB!!*

*Zwei Themen, die bislang in unserer Monatszeitschrift nur kurz behandelt werden konnten und vielfach als Wunsch geäußert wurden, haben wir ausführlich für Sie ausgearbeitet.*

*Unsere Übersicht der Diskettenwerkzeuge macht Sie mit den Grundlagen dieser hilfreichen Tools bekannt und zeigt Ihnen gleichzeitig alle auf dem Markt befindlichen Programme.*

*Zu welchen musikalischen Leistungen ihr CPC in der Lage ist, erfahren Sie nach Studium des Kapitels »Sound relativ«. Abgerundet wird dieses Thema durch einige sehr interessante Anwendungen, die Ihnen das komfortable Erstellen und Einbinden von Soundroutinen in eigene Programme ermöglichen.*

*Die Freunde des Genres Computerspiel erhalten eine wahre Fundgrube an Informationen über neue Spiele und Abenteuer, viele hilfreiche Tips und Lösungshilfen zu bekannten Programmen sowie ganze Lagepläne einiger komplexer Spiele.*

*Alles in allem dürfte, so hoffen wir, für jeden das Richtige dabei sein.*

*Abschließend noch ein paar Geheimtips unserer Redakteure:*

*Für Spielefreunde:*
*Guazuela und Crazy Stamps*
*– zwei originelle Spiele, bei denen garantiert keine Langeweile aufkommt.*

*Für Grafikprogrammierer:*
*GBasic*
*– d a s Programm für verblüffende Effekte.*

*Für Vortex-Anwender:*
*Disktool und DOS-Erweiterung*
*– zwei Super-Hilfen für Speichererweiterung und Floppy.*

*Für PASCAL-Programmierer:*
*Deluxe Data System*
*– komfortable und universelle Dateiverwaltung unter Turbo-Pascal.*

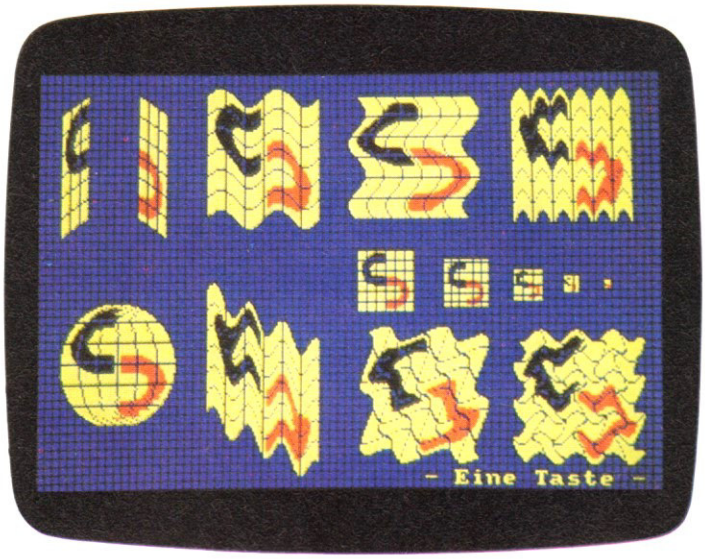
*und, und, und....*

*CPC pur – wir wünschen viel Spaß bei der Lektüre des Sonderheftes.*

*Herzlichst Ihr*

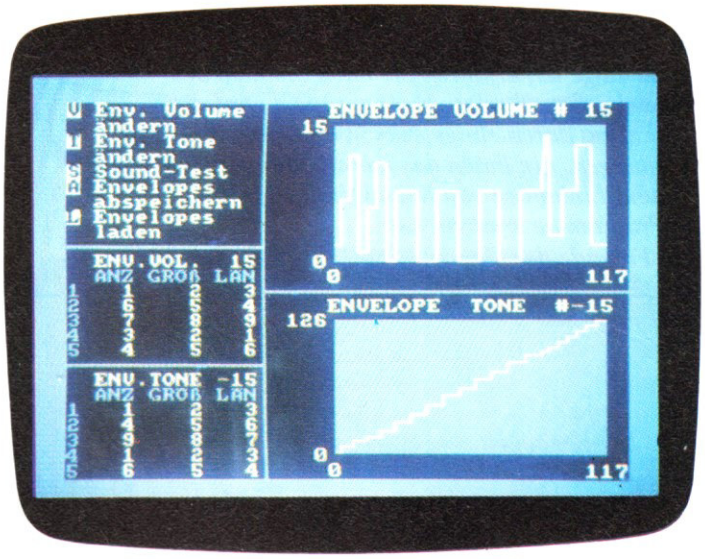
Stefan Ritter

*Stefan Ritter, Chefredakteur*




*GBasic – das neue Grafik-Basic. So tolle Grafiken sind nun einfach und schnell zu programmieren.*

*S. 6*




*Grundlagen der Soundprogrammierung. U.a. ermöglicht Ihnen ENVPRO die komfortable Generierung von Hüllkurven.*

*S.29*



*Die Axt im Haus erspart den Zimmermann. Diskettenwerkzeuge im Überblick*

*S. 76*

## Grundlagen:

**Soundprogrammierung für Jedermann** 22
– Sie erfahren alles über die effektive Soundprogrammierung der CPCs. Besonders interessant sind die zahlreichen Programmbeispiele, die es u.a. ermöglichen, die erstellten Routinen in eigene Programme einzubauen.
**PSG- der Sound-Programmer** 22
**ENVPRO- Hüllkurvengenerator** 29
**Sounddemo-tolle Beispiele** 32

## Anwendungen:

**GBasic** 6
– das neue,unschlagbare Grafik-BASIC!. Mit dieser Erweiterung zaubern Sie spielend die schönsten Grafiken auf den Bildschirm.
**Terminalprogramm** 17
– komfortables Programm für die Steuerung unserer RS-232 Schnittstelle.
**Colfunktion** 34
– Fraktale auf dem CPC! Die Grafiken können mit Musik untermalt werden.
**Morsetrainer** 36
– der CPC funkt SOS.
**Tilgungsplan** 43
– einfache Berechnung von Darlehenskosten und monatlichen Belastungen bei Kreditaufnahme.
**Defprint** 46
– Ausgabe von selbstdefinierten Zeichen auf Drucker.
**Timeout** 50
– und der Bildschirm wird dunkel. Dieses tolle Programm schaltet bei Bedarf den Bildschirm ab.
**Bundesliga** 54
**Unilab** 83
– Erstellen und Drucken von Etiketten in 4 verschiedenen Formaten.
**Wertpapierbilanz** 102
– archiviert und verwaltet Ihre Wertpapiere. Damit Sie ständig den Überblick behalten…

## Turbo-Pascal:

**Deluxe Data System** 116
– leistungsfähige und universelle Dateiverwaltung
**Windows** 122
– komfortable Fenstertechnik in Turbo-Pascal.

## Spiele:

**4 gewinnt** 124
– das beliebte Strategiespiel in einer einmaligen Version!
**Poker** 130
– blufft er oder nicht? Der CPC läßt sich in die Karten schauen…
**Secret Cavern** 136
– als Hubschrauberpilot sind Sie unterwegs in den undurchdringlichen Höhlensystemen.
**Tennis Mangager** 146
– schlüpfen Sie in die Rolle von Ion Tiriac und managen unseren BUM-BUM-Boris.
**Guazuela** 153
– dieses Strategiespiel um Macht wird jeden fesseln. Als Herrscher von Guazuela können Sie auf Ihr Land Einfluß nehmen,Kriege führen,ihr Nummernkonto in der Schweiz ausbauen u.v.m.

## Spiele:

**Mensch ärgere den CPC nicht** 166
– das älteste Gesellschaftsspiel jetzt in einer Neuauflage.

**Schlumpis Horror-Trip** 170

**Runner** 174

**Memory** 178
– eine Super-Version in Maschinensprache! Trainieren Sie Ihr Gedächtnis auf spielerische Weise.

**Crazy Stamps** 188
– das verrückteste Computerspiel seit es Briefmarken gibt. Als Postbeamter haben Sie die Aufgabe, eingehende Briefe ordnungsgemäß zu frankieren und abzustempeln. Aber es warten einige, nur schwer überwindbare Probleme auf Sie...

## Tips & Tricks:

**CPC Listing-Service** 92

**Convert** 94
– setzt Bilder des Profi-Painter so um, daß diese mit anderen Programmen weiter verarbeitet werden können.

**Vortex-Disktool** 97
– komfortable Verwaltung und Steuerung der Vortex-Floppy.

**Sieb des Erasthones** 100
– Ready to Use-Tips in Assembler

**MC-Software Clock** 101

**Slime** 101

**DOS-Erweiterung F1-X** 106
– leistungsstarke Befehlserweiterung für Vortex-Anwender.

**Drucken mit 8.bit** 108

**Edit-List** 111
– neue Editiermöglichkeiten verbessern die Fehlersuche in Programmen.

**Trace** 112
– diese kurze Routine verlangsamt den Programmablauf um einen Faktor x.

**Tonkopfjustage** 115
– nie wieder Read-Errors ! Dieses tolle Programm bietet Kassettenbenutzern die Möglichkeit, den Tonkopf ihres Rekorders neu zu justieren.

## Hardware:

**Ready-Signal** 87

**Zeitsignal** 88

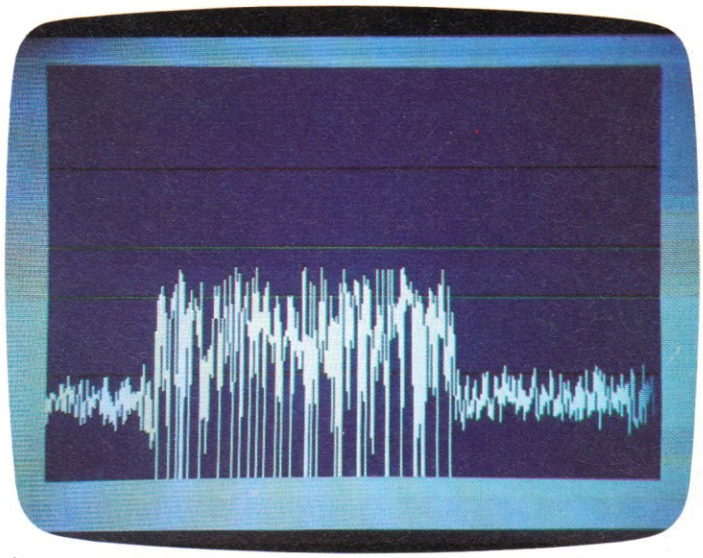
**Zwei 51/4"Laufwerke am CPC** 91
– wie das zu realisieren ist, zeigt unsere kurze Bauanleitung.

## Spieletips/Abenteuer:

**Diskettenutilities** 76
– die unabläßlichen Werkzeuge für den Anwender. Jetzt in einer großen Marktübersicht alle Tools auf einen Blick.

## Rubriken:

**Editorial** 3
**Impressum** 194
**Inserentenverzeichnis** 194
**Umfrage** 195



*Tonkopf justieren-unser Programm machts möglich. Nie wieder Read Errors auf Kassette...*

***S.115***




*Tips und Lösungshilfen sowie Lagepläne bietet unsere Spiele-Rubrik in Hülle und Fülle. Informieren Sie sich...*

***S.124***




*Crazy Stamps – als Postbeamter haben Sie einiges zu tun, denn die Briefmarken spielen verrückt.*

***S. 188***




# GBASIC – 30 neue Grafikbefehle

**Diese Programme ermöglichen die Generierung und Verwaltung von 30 neuen, als RSX eingebundenen Grafikbefehlen.**

Bei der Programmierung dieser Befehlserweiterung wurde auf einen universellen Einsatz Wert gelegt. Knackpunkt bei dieser Erweiterung ist das einfache Verändern von Grafiken (z.B. spiegeln usw.). Vor allem sollten wirklich neue Befehle geschaffen werden.

**Zunächst ihre Fähigkeiten:**
- Einlesen von Linien
- Ausgabe von Linien
- Variable Strichbreite
- Verwaltung eines zweiten Bildschirmspeichers im RAM
- Raster-Fill

Hier nun die Erklärung der Befehle. Zu erwähnen bleibt nur, daß eventuelle Fehler als abfangbarer ( → ON ERROR GOTO) SYNTAX-ERROR gemeldet werden. In der Anleitung steht ein ! für den geraden Strich vor einem RSX-Befehl und ein # für einen Klammeraffen. Buchstaben wie x/y hinter den Befehlen stehen als Platzhalter für Ausdrücke.

**!TDRAW,x,y**
Zieht in Gedanken einen Strich von der aktuellen Grafikposition zur angegebenen Position x/y ( →DRAW). Nun holt sich der Computer die Farbinformationen der Punkte, die auf dieser gedachten Linien liegen, codiert sie und legt sie zur weiteren Verarbeitung in einen Buffer (Speicherplatz, →BUFFER), der 640 Bytes umfaßt, ab.
1. GBASIC legt die Farbinformation (entspricht den Farbnummern) ab der Stelle 0 im Buffer ab.


Fuer alle CPC's
Fuer alle CPC's
Fuer alle CPC's

1. Die Variable muß vom Typ INTEGER sein (→DEFINT oder a%) und schon einen Wert haben. (z.B.1).
Bsp.: v%=0:|BEEK,1,#v%:? v%

**|BOKE,<Bufferpos.>,#Integervariable**
Legt die Farbnummer an der angegebenen Stelle im Buffer ab.
Bsp.: |BOKE,1,1

**|LEN,#<INT-Variable>**
Dieser Befehl übergibt eine Zahl (0-640) in die Integervariable. Für den Wert gibt es zwei Bedeutungen:
- Nach TDRAW: Gibt die Bufferstelle
- 1 an, an der die letzte Farbinformation abgelegt wurde.
- Nach WDRAW: Gibt die Bufferstelle -1 der zuletzt ausgegebenen Farbinfomation aus.

Beachte: Richtige Übergabe der Variablen.
Bsp.: 1%=0:|LEN,#1%:? 1%

**|ENG,<Eng-Zahl>,<Ink-Nummer>**
Dieser Befehl läßt die Farbinformation im Buffer zusammenrücken. Wie stark dies geschehen soll, hängt vom 1. Parameter ab. Je kleiner er ist, desto stärker wird der Buffer zusammengeschoben. Bei 1 als <ENG-Zahl> wird z.B. jede 2. Farbinformation verschluckt. Allgemein ergibt sich folgender Zusammenhang: Jede <Eng-Zahl>+1 te Farbinformation wird verschluckt.
Hierbei wird hinten Bufferplatz frei, der mit der <INK-Nummer> aufgefüllt wird.
Bsp.: |ENG,2,0

**|BREIT**
Hier geschieht das Gegenteil wie bei ENG. Der Bufferinhalt wird – vorgeschriebenermaßen – auf das doppelte aufgeblasen. Jeder Farbpunkt wird also verdoppelt.

**|WANDEL,<1.Farbe>,<2.Farbe>,A,L**
Dieser Befehl bearbeitet einen Teil des Buffers. Dieser Teil wird durch A und L beschrieben – dem »A«nfang und der »L«änge des Teilstücks. Diese Art der Teildefinition taucht auch bei anderen Befehlen wieder auf.
Der o.g. Bereich wird nach der 1. Farbe (INK-Farbnummer) durchsucht. Findet er die 1. Farbe, wird diese durch die 2. ausgetauscht und GBASIC sucht weiter.
Beachte: A und L dürfen zusammen nicht >640 (&280) sein.
Bsp.: |WANDEL,1,2,0,640
→ Alle Farben 1 werden im gesamten Buffer zu Farbe 2.

**|DREHE,A,L**
Dreht den Teilbereich des Buffers (Anfang, Länge) um.
Bsp.: |DREHE,0,640
Dreht den gesamten Bufferinhalt.

**|RECHTS,<Wieviel Punkte>,A,L**
Der Teilbereich wird um <Wieviel Punkte> nach rechts kopiert/verschoben.
- Einen Befehl »LINKS« braucht man wegen des 3. Parameters bei WDRAW nicht.
- Alle der Parameter müssen zusammen <640 (&280) sein.

Bsp.: |RECHTS,10,0,630
→ Verschiebt den gesamten Bufferinhalt um 10 Punkte nach rechts.

**|FILL,<Farbe>,A,L**
Dies ist nicht der RASTER-FILL! Hier wird nur der Teilbereich (A,L) mit der <Farbe> aufgefüllt. Löschen des Buffers: |FILL,0,0,640

**|VORNE,Anfang,Farbe,Flagge,#INT-Var**
Dieser Befehl durchsucht den Bufferinhalt von VORNE nach hinten, am ANFANG angefangen nach der FARBE. Die Stelle wird in der INTeger VARiablen übergeben. Wurde nichts gefunden, steht dort −1. Die FLAGGE gibt an, wie gesucht wird. Wenn:

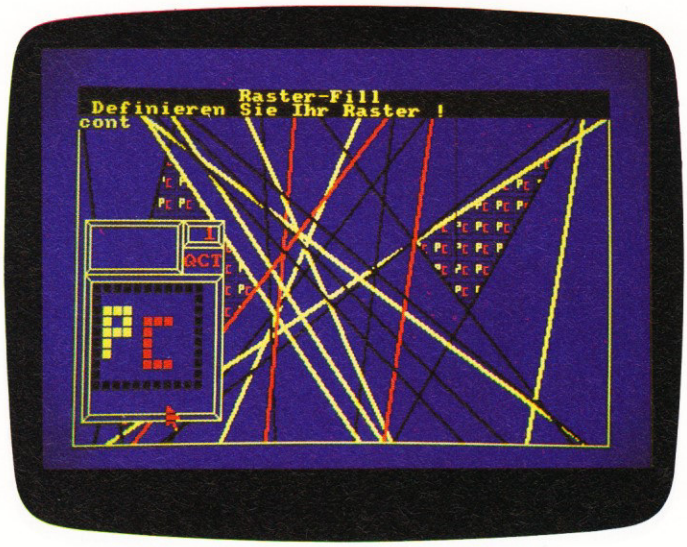



FLAGGE=1: Die Farbe gefunden wurde
FLAGGE=0: Eine andere Farbe gefunden wurde.
Bsp.: v%=0:|VORNE,0,1,0,#v%:? v%
Der Buffer wird von VORNE ab Stelle 0 nach einer FARBE <>1 (FLAGGE=0) gesucht:
Gefunden: v%=Stelle
Nicht gefunden: v%=-1

**|HINTEN,Anfang,Farbe,Flagge,#IN-Var**
Durchsucht den Buffer ab ANFANG von HINTEN aus. Die anderen Parameter sind gleich wie bei VORNE.
Bsp.: v%=0:|HINTEN,400,1,1,#v%:? v%
Der Buffer wird ab Stelle 400 von HINTEN nach vorne nach der Farbe 1 (FLAGGE=1) durchsucht.

**|SCRDEF,<Adresse>**
GBASIC erlaubt es, einen 2.Bildschirmspeicher im RAM zu verwalten. Dieser Befehl legt die Adresse für diesen fest.
- Die Adresse muß >&4000 sein.
- Der 2.Bildspeicher sollte durch MEMORY (→Handbuch) geschützt sein.

**|SCR1 |SCR2**
Diese beiden Befehle erlauben es, zwischen dem normalen Bildschirmspeicher (1,Anfang &C000,immer auf Monitor) und einem 2. Bildschirmspeicher (→SCRDEF, nie von alleine sichtbar) hin und her zu schalten.
|SCR1: Normalbildschirmspeicher
|SCR2: 2. Bildschirmspeicher
Was heißt hier aber »schalten«?
Folgende Befehle beziehen sich nun nicht mehr auf den normalen Bildschirmspeicher, sondern auf den mit |SCR<X> selektierten:
|TDRAW(R); |WDRAW(R); |TWDRAW;
|PRASTER und auch:
DRAW(R);PLOT(R);Textausgabe im TAG-Modus (→Handbuch, TAG, TAGOFF)
Bsp.: MODE 1
MEMORY & 4FFF:|SCRDEF,&5000
|SCR2:MOVE 0,0:DRAWR 640,400
|SCR1
So wurde nun ein Strich im 2. Bildspeicher (ab &5000=20480) gemacht. Auf dem Monitor hat sich nichts geregt. Mit |REIN (→) kann nun der 2. Speicher hereingeholt und die Linie bewundert werden.




1. Es darf keine Grafikausgabe mit |WIDTH auf dem 2. Bildspeicher geben!
2. Ein Hardware-Scroll (verschieben des gesamten Bildschirmes nach oben oder unten) sollte vermieden werden. Es können Fehler auftreten.

**|REIN (,L,R,O,U,X,Y)**

Mit diesem Befehl kann man ein Bild (oder nur ein Teil davon) aus dem 2. Bildschirmspeicher in den normalen Speicher herREINkopieren. Hier gibt es zwei verschiedene Formen des Befehls:

a) REIN ohne Parameter: der ganze Bildschirmspeicher wird kopiert.

b) REIN mit 6 Parametern: Hier wird nur ein Teil-Rechteck kopiert. Aus einem beliebig gelegten Rechteck im 2. Bildschirmspeicher werden die Daten in ein gleich großes, aber auch beliebig platziertes Rechteck im 1. Speicher kopiert. GBASIC braucht also Größe und die beiden Positionen der Rechtecke oder Fenster (WINDOWs). Das Fenster wird wie bei der WINDOW Angabe im Mode 2 definiert:

»L«inks,»R«echts,»O«ben,»U«nten

Dies definiert Größe und Platz des Windows im 1. (normalen) Bildspeicher. Für das Fenster, aus dem die Daten kommen (das für den 2. Bildschirmspeicher), brauchen wir nur noch die Koordinaten der linken oberen Ecke:

'Y'-Koor.(1-25)
'X'-Koor.(1-80)

Beachte: Bei Hardware-Scroll gibt es Fehler und Verschiebungen.

Bsp.: |REIN,1,40,1,25,40,1

Holt die rechte Hälfte des Bildschirms aus dem 2. Speicher und kopiert ihn an die linke Hälfte im normalen Speicher.

**|RAUS (L,R,O,U,X,Y)**

Dieser Befehl macht genau das gleiche wie REIN, nur wird hier vom normalen in den 2. Speicher RAUS- kopiert.

**|CHANGE (L,R,O,U,X,Y)**

Dieser Befehl ist wieder ähnlich den Befehlen REIN/RAUS; so ist auch die Parameterauswertung gleich. Hier werden aber die Bildschirminhalte gegenseitig ausgetauscht. So wechselt z.B. CHANGE in Reinform die beiden Bildschirmspeicher aus. Das was im 1. Speicher sichtbar war, ist nun im 2. und umgekehrt ist der 2. Speicher nun sichtbar. Bei CHANGE gehen also keine Daten verloren, es wird nichts überschrieben.

**|WIDTH,x,y**

Legt die Größe eines Grafikpunktes bei PLOT(R), DRAW(R), W/TDRAW(R), PRASTER und bei Textausgabe mit TAG fest. Die Größe des Grafikpunktes kann in X-(Breite) und Y-(Höhe) Koordinaten festgelegt werden. Hiermit kann also die Strichstärke bei DRAW festgelegt werden (!). Wird als X-Koordinate 0 angegeben, so wird die WIDTH-Routine ausgeschaltet (Grafik mit |WIDTH,1,1 ist langsamer als abgeschaltete mit |WIDTH,0,0).

Beachte:

1. Die WIDTH-Ausgabe darf nicht auf dem 2. Bildschirm erfolgen.
2. Wenn der Grafikpunkt rechts oder unten über den Bildschirm hinausgeht, wird er links oder oben weitergeführt.
3. Die Koordinaten sind **physikalisch** und modeabhängig:
- Nullpunkt ist immer links unten
- Y-Koordinaten von 0 bis 199
- X-Koordinaten modeabhängig:

| | | |
|---|---|---|
| MODE 0 | 0-159 | |
| MODE | 0-319 | 1 |
| MODE | 0-639 | 2 |

4. X- und Y-Koordinaten dürfen bei WIDTH 255 nicht überschreiten.

Bsp.:|WIDTH,160,199:PLOT 0,399,0
löscht im MODE 1 die linke Hälfte des Bildschirms.

|WIDTH,3,3:MOVE 0,0:DRAW 640,400
Zieht einen diag., dicken Strich

**|ENCODE,<Farbnummer>,#<INTeger-Var>**

Kodiert die Farbnummer in eine Bitmaske und übergibt diese in die Variable. Die Bitmaske wird bei den Befehlen PRASTER und RFILL besonders wichtig bei der RASTER-Definition. Auch bei W/TDRAW's werden die Farben in dieser, übrigens in jedem Mode anderen Bitmaske, in den Buffer eingetragen.

**|GMODUS,<0-4>**

Die Schneider haben verschiedene Grafikmodi bei PLOT/DRAW ...:

| | |
|---|---|
| 0: FORCE | Mode Normalmodus |
| 1: XOR | Mode Invertiert |
| 2: AND | Mode |
| 3: OR | Mode |

und bei GBASIC einen neuen:

| | |
|---|---|
| 4: »NORCE« | Mode |

Die Grafikmodi müssen unter GBASIC mit diesem Befehl angewählt werden! Veränderungen mit Steuerzeichenfolgen setzen GBASIC außer Gefecht. Der 4. Modus setzt Punkte wie bei FORCE (also normal), außer es wird mit der Farbe 0 geplotet, dann wird nämlich nichts getan. (→»SHOW«)

Bsp.: |GMODUS,2 AND-Mode einstellen

Nun ein umfangreicher Befehl:

**|PFEIL**,fl,fr,fo,fu,ascii-links,-rechts,-oben,-unten,wait, x-step,y-step,@X,@Y,@F

Der Befehl läßt einen Pfeil in einem definierten Fenster durch Tastendruck bewegen. Die Routine braucht 3 INT-Variablen. Die Variablen X und Y geben die aktuelle Position an. Sie sind mit **physikalischen** Koordinaten anzugeben (→WIDTH). Wird die Routine durch Drücken einer falschen Taste, die nicht zur Pfeilsteuerung dient, beendet, so enthalten X und Y die akt. Koordinaten und F den ASCII- Code der zuletzt gedrückten Taste. Welche Tasten den Pfeil steuern, wird auch angegeben:

ASCII-Links: ASCII-Code für Pfeil nach links bewegen
ASCII-Rechts ...

Das Fenster, in dem sich der Cursor bewegen darf, wird durch 4 Parameter angegeben. Sie beinhalten physikalische Koordinaten:

fl: Fensterbegrenzung links
fr: rechts ; fo: oben ; fu: unten
<Wait>, ein Wert von 0–255 erzeugt
bei höheren Werten eine Verzögerung des Pfeils.

<X-step> & <Y-step> sagen, in wie großen Schritten der Pfeil in X- bzw. Y-Richtung bewegt wird.
Bsp.: MODE 1:DEFINT x,y,z:x=160:y=100:speed key 20,1:
|PFEIL,0,319,199,0,242,243,240,241,0,2,2,#x,#y,#f
läßt den Pfeil mit den Cursortasten ab der Bildschirmmitte auf dem ganzen Bildschirm bewegen. Eine andere Taste als die Cursorsteuerung bringt den Abbruch des Befehls.

**|ZEIGE,x,y**
Bildet den Pfeil an der physikalischen Koordinate (→WIDTH) ab. Der Pfeil wirkt invertierend, d.h. einmal an einer Stelle dargestellt, kann er ohne den Hintergrund zu verändern, durch nochmaliges Setzen wieder gelöscht werden.
Bsp.: |ZEIGE,160,100:WHILE INKEY$=""
malt einen Pfeil und löscht ihn auf Tastendruck wieder.

**|RFILL,#raster$**
Dieser leistungsfähige Befehl, abgeleitet aus der FILL-Routine im Schneider International (4/86), füllt eine beliebig begrenzte Fläche ab der akt. Grafikposition (mit »MOVE« gesetzt) mit einem bestimmten mehrfarbigen (!) Muster aus. Das Muster, auch Raster genannt, ist im String (in diesem Fall) raster$ enthalten. Das Raster baut normalerweise auf einer 12×12 großen Matrix auf. Die Matrixgröße kann in weiten Grenzen mit dem Programm »GTUNER« verändert werden. Einzige Grenze bildet die Stringlänge von 255 Zeichen, in denen die Farbinformationen des Rasters ja abgelegt werden müssen. Und so werden die Daten in den String abgelegt:
Zuerst 12 Mal (oder X-Breite bei anderer Größe) CHR$(1), dann Farbdaten, hier 144 Stellen (oder XY Stellen bei anderer Größe) dann wieder 12 Mal CHR$(1) (→ entspr.: STRING$(12,1)
Wie sind aber die Farbdaten (die 144 Stellen in der Mitte) genauer definiert? Wir gehen einmal davon aus, daß Sie ihr Raster kennen und evtl. schon auf einem karierten Stück Papier notiert haben. Die Farbdaten werden nun aus dem Raster von oben links nach unten rechts zeilenweise geLESEN (so, wie man liest!). Als Farbdaten meine ich die Farbnummer (oder INK-Stift) eines Punktes. Nun habe ich die Farbnummer meines Punktes, muß diese mit |ENCODE in eine Bitmaske umwandeln und den erhaltenen Wert mit CHR$ an den String anhängen:
a%=1(Farbnr.=1):|ENCODE,a%,#a%:r$=r:$+chr$(a%)
(→ RESIGNER)
Beachte:
1. Stringübergabe mit Klammeraffen
2. Ein in einem Mode generiertes Raster ist in anderen Modi nicht gleich.
3. Bei Speicherknappheit oder sehr komplizierten Figuren läßt RFILL Lücken, die durch nochmaliges Füllen geschlossen werden können.
4. Die Platzierung des Rasters ist nicht von der Startkoordinate beim Füllen abhängig!
5. RFILL bleibt beim Füllen im Grafikfenster (→ ORIGIN; CLG)

**|PRASTER,x,y,#raster$**
Dieser Befehl platziert das Raster an die physikalischen Koordinaten x/y. Raster$ gibt das Raster an. (→ RFILL).
Nun aber zu den Listings (Li.):
Li.1:**(GDATA)** Dieses Programm liest GBASIC aus den Data's in den Speicher und paßt es automatisch an den Schneider, den Sie haben, an. GBASIC wird dann als Binärdatei »PROBE« abgespeichert. Auf diese Datei greifen nun Listing 2 und 3 zurück.
Li.2:**(Gtuner):**Mit diesen Listing kann GBASIC noch verändert werden. Hier die einzelnen Menüpunkte:
1 – GBASIC verschieben: Nach Eingabe der End- oder Anfangsadresse wird GBASIC dorthin verschoben. Die akt. End- und Anfangsadresse wird auch oben im Hilfsfenster angezeigt.




2 – Merkzettel: Hier werden, entweder auf Drucker oder Bildschirm, die wichtigsten Daten der aktuellen GBASIC-Version ausgegeben. So auch das Stringformat bei RFILL/PRASTER.
3 – Rastergröße: Hier kann die Rastergröße verändert werden. (→RFILL).
4 – GBASIC abspeichern: Nach Eingabe des Dateinamens wird GBASIC auf Disk oder Kassette abgespeichert. Wenn man nun GBASIC in einem Programm benutzen will, ist folgendes ins Programm zu schreiben
MEMORY <akt. Anfangsadresse>-1
LOAD"<Dateiname>",<akt. Anfangsadr>
CALL <Akt. Anfangsadresse>
So kann sich jeder sein persönlich gebrauchtes GBASIC zusammenstellen.
Li.3:**(SHOW)** dieses Programm demonstriert die Fähigkeiten von GBASIC recht eindrücklich. Es muß nicht unbedingt abgetippt werden, dient nur zur Analyse von Programmtechniken mit GBASIC
Li.4:**(RESIGNER)**. Zeigt nochmals den Aufbau eines Rasters als Programm.
GBASIC – INTERN
Hier möchte ich die grobe Funktionsweise des Programmes etwas erläutern. Das Programm entstand schrittweise. Zuerst T/W-DRAW, dann die Verwaltung des 2. Bildschirmspeichers und am Ende der Raster- Fill. Anfangs dachte ich nicht, daß GBASIC so groß werden würde, aber mir fiel schrittweise immer etwas neues ein.
GBASIC arbeitet bei fast allen Funktionen mit der SCR-Write In- direction. Wenn der CPC einen Grafikpunkt (oder mehrere) ausgeben will, so springt er an eine Adresse im RAM, die normalerweise auf den aktuellen Grafikmodus zeigt. Wird nun diese Adresse verbogen, was GBASIC des öfteren tut, so läuft jeder Punkt, der bei: DRAW / PLOT / Textausgabe mit TAG ausgegeben wird, über die eigene Routine.
1. **TDRAW.** Der Punkt wird nicht gedruckt, sondern im Buffer festgehalten. Dies, während er eine Linie zieht.
2. **WDRAW.** Der Punkt wird gesetzt, nur die Farbe wird aus dem Buffer geholt.
3. Das Betriebssystem gibt den Punkt u.a. mit der Bildschirmspeicheradresse an. Beim Arbeiten mit dem 2. Bildspeicher wird also nur von dieser Adresse ein Wert abgezogen. Schon landet er im 2. Speicher.
4. **WIDTH.** Anstelle eines kleinen Punktes, wird ein entsprechend großer gemacht.
Durch geschicktes Verbiegen der Adressen können diese Funktionen auch gemischt werden (z.B. WIDTH & WDRAW; TDRAW & 2. RAM usw.). Dann kann ich Ihnen nur noch gutes Programmieren und viel Spaß wünschen! ■

```
10 REM Raster-Designer                                  [825]
20 REM                                                  [272]
30 REM Uebergibt das Raster im Data-Feld i              [4045]
n r$
40 REM Die Zahlen in den Datas geben die F              [3227]
arbnummer
50 REM in Hexadezimalen Ziffern an.                     [2078]
60 REM                                                  [272]
70 REM                                                  [272]
80 REM                                                  [272]
90 INPUT "Mode (0-2) ";a:IF a<0 OR a>2 THE              [1816]
N 90
100 MODE a:DEFINT v:v=0:READ x,y:r$=STRING              [3487]
$(x,1)' obere 1-er Reihe
110 RESTORE 150                                         [749]
120 REM Zuerst die Groesse des Rasters:                 [2437]
130 DATA 12,12                                          [413]
140 REM dann die Farbdaten                              [1300]
150 DATA 123456789ABC                                   [894]
160 DATA 23456789ABCD                                   [668]
170 DATA 3456789ABCDE                                   [757]
180 DATA 456789ABCDEF                                   [1360]
190 DATA 56789ABCDEF0                                   [695]
200 DATA 6789ABCDEF01                                   [901]
210 DATA 789ABCDEF012                                   [1228]
220 DATA 89ABCDEF0123                                   [228]
230 DATA 9ABCDEF01234                                   [815]
240 DATA ABCDEF012345                                   [326]
250 DATA BCDEF0123456                                   [1001]
260 DATA CDEF01234567                                   [814]
270 FOR n=1 TO y:READ a$:IF LEN(a$)<>x THE              [3303]
N PRINT"DATA ERROR !":END
280 FOR m=1 TO x:a=VAL("&"+MID$(a$,m,1)):!              [2706]
ENCODE,a,@v
290 r$=r$+CHR$(v):NEXT m:NEXT n                         [2134]
300 r$=r$+STRING$(x,1)                                  [972]
310 !PRASTER,160,100,@r$:END                            [1474]
```

```
10 MEMORY &8FFF                                         [207]
20    DATA 21,00,00,22,12,90,01,14,90,21,0              [3184]
8,91,CD,D1,BC,C3,D5,95,1739
30    DATA 00,00,70,90,C3,35,91,C3,3F,91,C              [2783]
3,49,91,C3,57,91,C3,65,2188
40    DATA 91,C3,A9,91,C3,BE,91,C3,D6,91,C              [2523]
3,EF,91,C3,04,92,C3,4A,2931
50    DATA 92,C3,63,92,C3,8B,92,C3,A9,92,C              [2830]
3,D4,92,C3,E9,92,C3,02,2900
60    DATA 93,C3,0D,93,C3,26,93,C3,2D,93,C              [2566]
3,32,93,C3,3F,93,C3,44,2329
70    DATA 93,C3,53,93,C3,92,93,C3,AA,93,C              [2554]
3,13,9C,C3,D3,9D,C3,39,2757
80    DATA 9E,C3,88,9E,54,44,52,41,D7,54,4              [2584]
4,52,41,57,D2,57,44,52,1994
90    DATA 41,D7,57,44,52,41,57,D2,54,57,4              [3229]
4,52,41,D7,42,55,46,46,1771
100   DATA 45,D2,42,45,45,CB,42,4F,4B,C5,4              [2791]
C,45,CE,45,4E,C7,42,52,1948
110   DATA 45,49,D4,57,41,4E,44,45,CC,44,5              [3556]
2,45,48,C5,52,45,43,48,1703
120   DATA 54,D3,46,49,4C,CC,56,4F,52,4E,C              [4230]
5,48,49,4E,54,45,CE,53,1905
130   DATA 43,52,44,45,C6,53,43,52,B1,53,4              [2658]
3,52,B2,52,45,49,CE,52,1815
140   DATA 41,55,D3,43,48,41,4E,47,C5,57,4              [3232]
9,44,54,C8,45,4E,43,4F,1716
150   DATA 44,C5,47,4D,4F,44,55,D3,50,46,4              [3826]
5,49,CC,5A,45,49,47,C5,1852
160   DATA 50,52,41,53,54,45,D2,52,46,49,4              [2621]
C,CC,00,00,00,00,00,00,1178
170   DATA 00,00,C3,96,C3,96,6B,0C,72,0C,7              [2199]
7,0C,7D,0C,F7,93,C3,00,1792
180   DATA 00,00,00,00,00,00,00,00,00,00,0              [2194]
0,00,94,CA,80,FC,16,FE,1006
190   DATA 95,0B,FE,CD,52,94,CD,F6,BB,CD,C              [2541]
7,95,C9,CD,52,94,CD,F9,3130
200   DATA BB,CD,C7,95,C9,FE,03,C2,DD,93,C              [3054]
D,01,94,CD,F6,BB,C3,C7,3146
210   DATA 95,FE,03,C2,DD,93,CD,01,94,CD,F              [3848]
9,BB,C3,C7,95,FE,09,C2,2963
220   DATA DD,93,DD,5E,10,DD,56,11,DD,6E,0              [3152]
E,DD,66,0F,CD,C0,BB,CD,2495
230   DATA 8C,95,CD,48,95,DD,6E,0A,DD,66,0              [2875]
B,DD,5E,0C,DD,56,0D,CD,2242
240   DATA F6,BB,CD,C7,95,DD,6E,06,DD,66,0              [3249]
7,DD,5E,08,DD,56,09,CD,2497
250   DATA C0,BB,CD,01,94,CD,F6,BB,C3,C7,9              [1903]
5,FE,01,C2,DD,93,DD,6E,3062
260   DATA 00,DD,66,01,CD,67,95,22,10,91,2              [3203]
2,12,91,C9,FE,02,C2,DD,2045
270   DATA 93,DD,6E,02,DD,66,03,CD,59,95,C              [2982]
D,6E,95,7E,CD,2F,BC,C3,2474
280   DATA 9D,93,FE,02,C2,DD,93,DD,6E,02,D              [2625]
D,66,03,CD,59,95,CD,6E,2539
290   DATA 95,DD,7E,00,CD,2C,BC,77,C9,FE,0              [2948]
1,C2,DD,93,2A,12,91,CD,2480
300   DATA 76,95,EB,DD,6E,00,DD,66,01,C3,8              [3493]
0,95,FE,02,C2,DD,93,DD,2668
310   DATA 66,02,2E,00,22,27,91,2A,10,91,5              [3306]
4,5D,01,80,02,ED,A0,E2,1502
320   DATA 31,92,D5,ED,5B,27,91,1C,7A,BB,2              [2986]
0,1B,23,1E,00,0B,78,B1,1689
330   DATA 20,13,D1,B7,ED,52,44,4D,62,6B,1              [2977]
3,DD,7E,00,CD,2C,BC,77,2034
340   DATA ED,B0,C9,ED,53,27,91,D1,18,CF,2              [3637]
1,80,02,CD,6E,95,EB,21,2453
350   DATA 40,01,CD,6E,95,01,80,02,ED,A8,2              [3066]
3,ED,A8,EA,5A,92,C9,FE,2430
360   DATA 04,C2,DD,93,DD,7E,04,CD,2C,BC,5              [4139]
F,DD,7E,06,CD,2C,BC,F5,2484
370   DATA D5,CD,1A,94,D1,F1,04,BE,CC,89,9              [3081]
2,23,0D,20,F8,05,20,F5,2333
380   DATA C9,73,C9,FE,02,C2,DD,93,CD,1A,9              [2057]
4,E5,50,59,19,EB,E1,CB,2800
390   DATA 38,CB,19,1A,ED,A0,1B,1B,2B,77,2              [2042]
3,EA,9D,92,C9,FE,03,C2,2147
400   DATA DD,93,DD,5E,00,DD,56,01,DD,6E,0              [3707]
2,DD,66,03,19,E5,4B,42,2045
410   DATA DD,5E,04,DD,56,05,19,CD,59,95,C              [2968]
D,6E,95,EB,E1,CD,6E,95,2487
420   DATA 03,ED,B8,C9,FE,03,C2,DD,93,CD,1              [2437]
A,94,54,5D,13,DD,7E,04,2370
430   DATA CD,2C,BC,77,ED,B0,C9,CD,32,94,D              [3031]
5,E5,21,80,02,59,50,B7,2530
440   DATA ED,52,44,4D,E1,D1,CB,CA,CD,6A,9              [3345]
4,C3,FA,91,CD,32,94,CB,2958
450   DATA 8A,CD,6A,94,C3,FA,91,FE,01,C2,D              [3351]
D,93,DD,6E,00,DD,66,01,2659
460   DATA CD,67,95,EB,21,00,C0,B7,ED,52,2              [2908]
2,0E,91,C9,21,00,00,22,1880
470   DATA 0C,91,C9,2A,0E,91,18,F7,21,3B,9              [2934]
3,22,AF,94,C3,96,94,EB,2154
480   DATA ED,B0,C9,21,3C,93,18,F1,21,49,9              [3796]
3,18,EC,1A,ED,A0,2B,77,2217
490   DATA 23,EA,49,93,C9,FE,02,C2,DD,93,D              [3199]
D,7E,02,B7,28,24,3A,29,2215
500   DATA 91,B7,20,11,2A,E8,95,22,FA,95,2              [3234]
1,EC,95,22,E8,95,3E,FF,2383
510   DATA 32,29,91,DD,7E,00,32,2B,91,DD,7              [3439]
E,02,32,2A,91,C9,3A,29,1707
520   DATA 91,B7,C8,2A,FA,95,22,E8,95,AF,3              [2826]
2,29,91,C9,FE,02,C2,DD,2667
530   DATA 93,DD,7E,02,CD,2C,BC,5F,16,00,D              [3052]
D,6E,00,DD,66,01,CD,80,2038
540   DATA 95,C9,FE,01,C2,DD,93,DD,7E,00,E              [3145]
6,07,2A,14,91,28,16,FE,2274
550   DATA 02,2A,16,91,38,0F,2A,18,91,28,0              [2587]
A,FE,03,2A,1A,91,28,03,1056
560   DATA 2A,1C,91,3A,29,91,B7,28,04,22,F              [3336]
A,95,C9,22,E8,95,C9,1E,1966
570   DATA 02,DF,2C,91,CB,09,D0,23,3A,C8,B              [3148]
1,FE,01,3E,88,28,06,3E,1865
580   DATA AA,38,02,3E,80,4F,C9,78,B7,C8,7              [2746]
E,A8,B1,A9,A8,77,C9,DD,2550
590   DATA 6E,00,DD,66,01,CD,4F,95,CD,A7,9              [3601]
5,DD,6E,02,DD,66,03,DD,2268
600   DATA 5E,04,DD,56,05,C9,DD,6E,02,DD,6              [4310]
6,03,DD,5E,00,DD,56,01,1893
610   DATA D5,E5,19,CD,59,95,E1,CD,6E,95,C              [2526]
1,C9,FE,04,C2,DD,93,DD,3034
620   DATA 6E,06,DD,66,07,44,4D,CD,59,95,C              [4391]
D,6E,95,DD,5E,04,DD,7E,2164
630   DATA 02,16,00,B7,C8,CB,C2,C9,FE,02,C              [2161]
2,DD,93,CD,8C,95,CD,48,2594
640   DATA 95,DD,6E,00,DD,66,01,DD,5E,02,D              [3626]
D,56,03,C9,7B,CD,2C,BC,2192
650   DATA 5F,7B,CB,42,28,03,BE,18,08,BE,2              [3320]
0,04,3D,C3,7F,94,AF,28,1724
660   DATA 11,CB,4A,28,03,23,18,01,2B,08,7              [2376]
8,B1,20,E1,21,FF,FF,C9,1749
670   DATA CD,76,95,C9,FE,06,28,18,B7,C2,D              [3112]
D,93,21,00,C0,E5,ED,5B,2524
680   DATA 0E,91,B7,ED,52,EB,E1,01,FF,3F,C              [4515]
D,00,00,C9,21,C8,B1,7E,2382
690   DATA 32,21,91,3E,02,77,DD,7E,08,DD,9              [3267]
6,0A,DA,39,95,DD,86,02,1928
700   DATA FE,52,D2,39,95,DD,7E,04,DD,96,0              [2565]
6,DA,39,95,DD,86,00,FE,2513
```

```
710   DATA 1B,D2,39,95,DD,6E,00,DD,66,02,2 [3324]
D,25,CD,1A,BC,ED,5B,0E,1942
720   DATA 91,B7,ED,52,E5,DD,66,0A,DD,6E,0 [2016]
6,DD,56,08,DD,5E,04,25,2217
730   DATA 2D,15,1D,DF,32,91,ED,53,23,91,D [3435]
1,E5,B7,ED,52,22,25,91,2169
740   DATA E1,0E,00,3A,23,91,47,C5,41,3A,2 [3769]
4,91,4F,E5,CD,AE,94,E1,2109
750   DATA CD,26,BC,E5,ED,5B,25,91,B7,ED,5 [2921]
2,EB,E1,C1,10,E5,3A,21,2661
760   DATA 91,32,C8,B1,C9,CD,32,95,C3,DD,9 [2713]
3,B7,21,00,00,C8,2A,0E,2212
770   DATA 91,C9,2A,10,91,22,12,91,C9,CD,5 [2816]
9,95,CD,6E,95,22,12,91,2051
780   DATA C9,7C,FE,02,D8,C2,DD,93,7D,FE,8 [4841]
1,D8,C3,DD,93,7C,FE,40,3088
790   DATA D0,C3,DD,93,D5,ED,5B,10,91,19,D [4052]
1,C9,D5,ED,5B,10,91,B7,2793
800   DATA ED,52,D1,C9,2B,7E,FE,01,C2,DD,9 [4013]
3,23,73,23,72,C9,CD,C0,2612
810   DATA 95,3A,C8,B1,21,4E,96,FE,02,28,0 [3257]
A,21,6B,96,FE,01,28,03,1739
820   DATA 21,94,96,22,E8,95,C9,CD,C0,95,3 [4281]
A,C8,B1,21,12,96,FE,02,2385
830   DATA 28,EF,21,3E,96,FE,01,28,E8,21,4 [3640]
6,96,18,E3,2A,E8,95,22,2012
840   DATA 1F,91,C9,2A,1F,91,22,E8,95,C9,2 [2849]
1,DD,95,22,CD,B1,C9,2A,2273
850   DATA CD,B1,22,E8,95,18,F1,E5,D5,ED,5 [4229]
B,0C,91,B7,ED,52,D1,CD,2905
860   DATA 00,00,E1,C9,D5,3A,2B,91,5F,3A,2 [4521]
A,91,57,E5,C5,E5,C5,CD,2369
870   DATA 00,00,CD,E2,93,15,20,F7,C1,E1,C [2991]
D,26,BC,3A,2A,91,57,1D,2088
880   DATA 20,E9,C1,E1,D1,C9,D5,C5,3E,80,0 [4074]
6,08,ED,5B,12,91,D5,FD,2664
890   DATA E1,E5,5F,CB,01,30,0B,C5,4B,FD,4 [3359]
6,00,FD,23,CD,1E,91,C1,2268
900   DATA CB,0B,10,ED,FD,E5,E1,22,12,91,E [2508]
1,C1,D1,C9,D5,C5,3E,88,2807
910   DATA 06,04,18,D2,D5,C5,3E,AA,06,02,1 [3619]
8,CA,D5,E5,C5,56,1E,80,2003
920   DATA 06,08,CB,01,30,09,7A,A3,28,02,3 [2296]
E,FF,CD,BA,96,CB,0B,10,1690
930   DATA EF,C1,E1,D1,C9,D5,C5,E5,56,1E,8 [2894]
8,06,04,CB,01,30,15,7A,2363
940   DATA A3,6F,0F,B5,0F,B5,0F,B5,6B,CB,0 [3677]
3,38,03,07,18,F9,5D,CD,1812
950   DATA BA,96,CB,0B,10,E3,E1,C1,D1,C9,D [3303]
5,C5,E5,56,1E,AA,06,02,2554
960   DATA CB,01,30,12,7A,A3,6F,0F,B5,6B,C [2453]
B,03,38,04,CB,07,18,F8,1717
970   DATA 5D,CD,BA,96,CB,0B,10,E6,E1,C1,D [2450]
1,C9,2A,12,91,77,23,22,2315
980   DATA 12,91,C9,00,00,00,00,00,00,00,0 [2857]
0,00,00,00,00,00,00,00,364
990   DATA 00,00,00,00,00,DF,CB,9B,C9,DF,C [2493]
E,9B,C9,8C,37,FE,50,37,2151
1000  DATA FE,00,00,00,00,C3,96,00,00,00,0 [3933]
0,00,00,00,FC,16,FE,CC,1331
1010  DATA B1,FE,FF,16,FE,1A,16,FE,00,00,0 [2908]
0,00,00,00,00,00,00,00,1264
1020  DATA 00,00,00,00,00,00,00,00,00,00,0 [1682]
0,00,00,00,00,00,00,C0,192
1030  DATA E0,F0,F8,FC,FE,FF,FF,FC,F8,CC,0 [3088]
C,06,06,FE,0E,C2,DD,93,3286
1040  DATA 21,A0,00,3A,C8,B1,B7,28,0C,FE,0 [3705]
2,28,05,21,40,01,18,03,1289
1050  DATA 21,80,02,22,FD,9B,26,00,DD,6E,1 [2475]
6,CD,26,9E,DA,DD,93,7D,2108
1060  DATA 32,EA,9B,DD,6E,14,CD,26,9E,DA,D [3727]
D,93,7D,32,EB,9B,DD,6E,2673
1070  DATA 16,BD,D2,DD,93,CA,DD,93,DD,5E,1 [2549]
A,DD,56,1B,CD,1C,9E,DA,2643
1080  DATA DD,93,ED,53,EC,9B,DD,5E,18,DD,5 [3191]
6,19,CD,1C,9E,DA,DD,93,2727
1090  DATA ED,53,EE,9B,2A,EC,9B,CD,2A,9E,D [3558]
2,DD,93,DD,6E,12,DD,66,2801
1100  DATA 10,DD,5E,0E,DD,56,0C,22,F0,9B,E [3159]
D,53,F2,9B,DD,7E,0A,32,2217
1110  DATA F4,9B,DD,7E,08,32,F5,9B,DD,7E,0 [4364]
6,32,F6,9B,DD,6E,04,DD,2564
1120  DATA 66,05,CD,30,9E,22,F7,9B,7E,23,6 [2391]
6,6F,EB,CD,1C,9E,DA,DD,2393
1130  DATA 93,ED,53,FF,9B,DD,6E,02,DD,66,0 [2492]
3,CD,30,9E,22,F9,9B,6E,2495
1140  DATA 26,00,CD,26,9E,DA,DD,93,22,01,9 [3974]
C,DD,6E,00,DD,66,01,CD,2076
1150  DATA 30,9E,22,FB,9B,ED,5B,FF,9B,2A,0 [3394]
1,9C,CD,A8,9D,DA,DD,93,2699
1160  DATA CD,E7,9D,ED,53,FF,9B,22,01,9C,C [1947]
D,18,BB,E5,21,F0,9B,BE,2777
1170  DATA 28,2E,23,BE,28,38,23,BE,28,3F,2 [2381]
3,BE,28,47,E1,F5,CD,E7,1977
1180  DATA 9D,F1,E5,2A,F7,9B,CD,2E,9D,E1,E [3624]
B,2A,F9,9B,CD,2E,9D,5F,2888
1190  DATA 16,00,2A,FB,9B,C3,2E,9D,73,23,7 [3759]
2,C9,3A,F5,9B,6F,26,00,1940
1200  DATA EB,B7,ED,52,EB,E1,18,23,3A,F5,9 [3701]
B,6F,26,00,19,EB,E1,18,2372
1210  DATA 18,E1,D5,3A,F6,9B,5F,16,00,19,D [3130]
1,18,0C,E1,D5,3A,F6,9B,2205
1220  DATA 5F,16,00,B7,ED,52,D1,CD,A8,9D,3 [3155]
8,15,D5,ED,5B,01,9C,CD,2338
1230  DATA 2A,9E,D1,20,14,E5,2A,FF,9B,CD,2 [3805]
A,9E,E1,20,0A,ED,5B,FF,2397
1240  DATA 9B,2A,01,9C,C3,FA,9C,D5,E5,2A,0 [2777]
1,9C,ED,5B,FF,9B,CD,E7,2770
1250  DATA 9D,E1,D1,CD,E7,9D,3A,F4,9B,B7,2 [2936]
8,07,47,AF,3D,20,FD,10,2479
1260  DATA FA,C3,F3,9C,D5,E5,23,13,E5,2A,E [2983]
C,9B,CD,2A,9E,30,1A,2A,2523
1270  DATA EE,9B,CD,2A,9E,38,12,E1,3A,EA,9 [2568]
B,BD,38,0C,3A,EB,9B,BD,2438
1280  DATA 30,06,37,3F,E1,D1,C9,E1,37,18,F [3831]
9,FE,02,C2,DD,93,DD,6E,2509
1290  DATA 00,DD,66,01,DD,5E,02,DD,56,03,C [3816]
3,E7,9D,D5,E5,CD,1D,BC,2398
1300  DATA 11,05,9C,ED,53,03,9C,11,0E,08,E [3232]
5,2A,03,9C,46,23,22,03,1268
1310  DATA 9C,E1,E5,C5,CB,00,30,03,7E,A9,7 [2920]
7,CD,E2,93,15,20,F3,C1,2542
1320  DATA E1,CD,26,BC,16,08,1D,20,DD,E1,D [2917]
1,C9,3A,FE,9B,BA,C0,3A,2506
1330  DATA FD,9B,BB,C9,3E,C8,BD,C9,7C,BA,C [3587]
0,7D,BB,C9,2B,7E,FE,01,2887
1340  DATA C2,DD,93,23,C9,CD,64,A0,D0,CD,7 [2487]
E,A0,79,32,D1,9B,32,D2,2757
1350  DATA 9B,06,0C,13,10,FD,01,0C,0C,ED,4 [3435]
3,27,91,E5,3A,D1,9B,4F,1704
1360  DATA 78,32,28,91,1A,47,13,DF,E1,9B,C [3424]
D,E2,93,79,32,D1,9B,3A,2245
1370  DATA 28,91,47,10,E5,E1,3A,D2,9B,32,D [2310]
1,9B,4F,CD,26,BC,ED,4B,2385
1380  DATA 27,91,06,0C,0D,ED,43,27,91,20,C [3272]
C,C9,CD,80,9F,D8,CD,C6,2251
1390  DATA BB,CD,F0,BB,CD,2C,BC,32,DD,9B,D [3330]
F,E7,9B,DF,E4,9B,D2,35,3160
1400  DATA 9F,E5,FD,E1,D5,DD,E1,CD,A4,9F,C [4027]
D,B4,9F,CD,1D,BC,3A,DD,3298
1410  DATA 9B,47,FD,23,CD,29,BC,CD,E5,9F,3 [4011]
A,34,B3,FD,95,38,05,78,2413
1420  DATA AE,A1,28,EC,FD,2B,CD,26,BC,CD,0 [3260]
0,A0,51,59,CD,1C,A0,E5,2495
1430  DATA C5,CB,01,DC,23,BC,78,42,AE,A1,5 [3092]
7,20,0F,B0,28,0C,CD,5B,2023
1440  DATA A0,38,07,DD,2B,CD,45,9F,DD,23,C [3263]
1,E1,E5,C5,CB,09,DC,20,2484
1450  DATA BC,78,43,AE,A1,5F,20,0F,B0,28,0 [3508]
C,CD,3F,A0,38,07,DD,23,1827
1460  DATA CD,45,9F,DD,2B,C1,E1,E5,2A,D7,9 [3059]
B,7E,E1,AE,A1,AE,77,3A,2793
1470  DATA 36,B3,3C,FD,95,30,0D,FD,2B,CD,2 [2958]
6,BC,CD,00,A0,78,AE,A1,2303
1480  DATA 28,A3,CD,2D,A0,2A,DB,9B,7D,B4,C [3453]
8,2B,22,DB,9B,CD,71,9F,2462
1490  DATA C3,9B,9E,D5,2A,89,AE,01,03,00,0 [4277]
9,38,07,EB,2A,8D,B0,CD,1949
1500  DATA 2A,9E,38,17,EB,22,89,AE,DD,E5,D [3420]
1,FD,E5,C1,2B,72,2B,73,2508
1510  DATA 2B,71,2A,DB,9B,23,22,DB,9B,D1,C [3642]
9,2A,89,AE,2B,56,2B,5E,2044
1520  DATA 2B,22,89,AE,6E,26,00,C9,FE,01,C [3825]
2,DD,93,CD,7E,A0,2E,0C,2103
1530  DATA 26,00,19,22,D3,9B,2A,D5,9B,01,8 [3123]
0,02,AF,77,23,3D,77,23,1548
1540  DATA 0B,78,B1,20,F5,37,3F,C9,E5,D5,6 [3083]
2,6B,19,EB,2A,D5,9B,19,2246
1550  DATA 22,D9,9B,D1,E1,C9,E5,D5,FD,E5,E [3024]
1,EB,21,C8,00,B7,ED,52,3160
1560  DATA 1E,0C,16,00,CD,C3,9B,2E,0C,26,0 [3213]
0,CD,C7,9B,E5,DD,E5,E1,2178
1570  DATA 1E,0C,16,00,CD,C3,9B,E1,19,ED,5 [2018]
B,D3,9B,19,22,D7,9B,18,2016
1580  DATA 34,E5,D5,2A,D7,9B,1E,0C,16,00,B [3403]
7,ED,52,7E,FE,01,20,05,1890
1590  DATA 1E,90,16,00,19,22,D7,9B,18,19,E [2868]
5,D5,2A,D7,9B,1E,0C,16,1592
```

```
1600  DATA 00,19,7E,FE,01,20,07,1E,90,16,0 [2838]
0,B7,ED,52,22,D7,9B,D1,1756
1610  DATA E1,C9,E5,D5,FD,E5,D1,2A,D9,9B,7 [3334]
B,BE,30,02,18,EF,77,18,2742
1620  DATA EC,E5,D5,FD,E5,D1,2A,D9,9B,23,7 [4187]
B,BE,38,02,18,DD,77,18,2577
1630  DATA DA,E5,D5,2A,D9,9B,23,23,FD,E5,D [3495]
1,7B,3D,BE,30,09,23,7B,2424
1640  DATA 3C,BE,38,03,37,18,C2,37,3F,18,B [3116]
E,E5,D5,2A,D9,9B,2B,2B,1856
1650  DATA 18,E2,FE,03,C2,DD,93,DD,6E,02,D [2918]
D,66,03,DD,5E,04,DD,56,2354
1660  DATA 05,DF,DE,9B,D0,CD,1D,BC,37,C9,E [2798]
5,DD,6E,00,DD,66,01,2B,2418
1670  DATA 3E,02,BE,C2,DD,93,23,23,5E,23,5 [2352]
6,E1,C9,00,00
1680 MODE 2                                 [513]
1690 RESTORE 20:sp=36864:bz=18:zz=20:spe=4  [2344]
1108
1700 z=0:ps=0:zen=980                       [1239]
1710 READ a$:a=VAL("&"+a$):POKE sp,a:ps=ps  [2652]
+a:z=z+1:sp=sp+1:IF z=bz THEN 1730
1720 IF sp=spe+1 THEN 1760 ELSE 1710        [1166]
1730 READ p:zz=zz+10:LOCATE 10,12:PRINT US  [4740]
ING" Zeile : #####  :  Soll : #####   Ist
: #####";zz-10,p,ps;
1740 IF p<>ps THEN PRINT" Data-Error in";z  [5045]
z-10:END ELSE PRINT"  ->  Alles OK !"
1750 IF zz-10=zen THEN 2160 ELSE 1700       [1316]
1760 sp=36864                               [395]
1770 a=PEEK(6)                              [1184]
1780 IF a=128 THEN 1830                     [1015]
1790 IF a=&7B THEN GOSUB 1840:GOTO 1810     [1012]
1800 GOSUB 1940                             [855]
1810 FOR n=1 TO 28:READ ad$,we$:ad=VAL("&"  [4043]
+ad$):wh=VAL("&"+LEFT$(we$,2))
1820 wl=VAL("&"+RIGHT$(we$,2)):POKE ad,wl:  [3638]
POKE ad+1,wh:NEXT
1830 GOTO 2040                              [355]
1840 REM Zeiger auf Datas fuer 664  stelle  [2397]
n
1850 RESTORE 1870:POKE &9BCD,253:POKE &9BD  [2350]
0,253:RETURN
1860 REM Datas fuer CPC 664                 [1131]
1870 DATA 9114,0C70,9116,0C76,9118,0C7B,91  [2699]
1A,0C81
1880 DATA 912C,CB4C,912F,1690,9132,0B97,9C  [2314]
1C,B7C3
1890 DATA 93E7,B7C3,94B3,B7C3,9536,B7C3,95  [2111]
90,B7C3
1900 DATA 95AB,B7C3,95D2,B7C8,95D6,B7C8,9B  [2324]
CB,DDB3
1910 DATA 9BCE,DD77,9BDE,1690,9EBD,B69F,9F  [1849]
1E,B6A1
1920 DATA 9BE1,B7C7,9BE4,1693,9BE7,1620,9F  [1390]
47,AE6C
1930 DATA 9F51,B071,9F5A,AE6C,9F72,AE6C,9F  [1920]
7A,AE6C
1940 REM Zeiger auf Datas fuer 6128 stelle  [2719]
n
1950 RESTORE 1970:POKE &9BCD,253:POKE &9BD  [2221]
0,253:RETURN
1960 REM Datas fuer CPC 6128                [1270]
1970 DATA 9114,0C74,9116,0C7A,9118,0C7F,91  [2083]
1A,0C85
1980 DATA 912C,CB49,912F,1694,9132,0B9B,9C  [2196]
1C,B7C3
1990 DATA 93E7,B7C3,94B3,B7C3,9536,B7C3,95  [2111]
90,B7C3
2000 DATA 95AB,B7C3,95D2,B7C8,95D6,B7C8,9B  [2144]
CB,DDAE
2010 DATA 9BCE,DD72,9BDE,1694,9EBD,B69F,9F  [2692]
1E,B6A1
2020 DATA 9BE1,B7C7,9BE4,1697,9BE7,1624,9F  [1637]
47,AE6C
2030 DATA 9F51,B071,9F5A,AE6C,9F72,AE6C,9F  [1920]
7A,AE6C
2040 SAVE"probe",b,&9000,4244,0:END         [1854]
2050 REM ****************************       [1471]
2060 REM *                          *       [403]
2070 REM *     DATAs fuer GBASIC    *       [1290]
2080 REM *                          *       [403]
2090 REM *   generiert "PROBE.BIN"  *       [1769]
2100 REM *                          *       [403]
2110 REM * COPYRIGHT SEPTEMBER 1986 *       [1753]
2120 REM *                          *       [403]
2130 REM *      BY DIETER BRAUN     *       [2264]
2140 REM *                          *       [403]
2150 REM ****************************       [1471]
```

```
2160 REM 1260 Nullen einfuegen              [1214]
2170 FOR q=1 TO 1260:POKE sp,0:sp=sp+1:NEX  [3258]
T
2180 zen=-1:GOTO 1700                       [868]
```

```
10 DEFINT a-z                               [553]
20 INK 0,1:PAPER 0:PEN 1:CLS:BORDER 1:INK   [3616]
2,0:INK 3,6
30 IF HIMEM<40000 THEN 60' zweites Laden a  [1417]
bfangen
40 MEMORY &8FFF:LOAD"probe"                 [1455]
50 init=&9000:CALL init:MEMORY 20000:|SCRD  [3760]
EF,20000
60 MODE 1:x=68:y=300:GOSUB 2960 'Schneider  [3158]
 auf 1. Schirm
70 |RAUS,1,16,3,10,1,1' in 2. Schirm ableg  [2476]
en
80 FOR n=240 TO 360 STEP 2:MOVE 8,n:|TDRAW  [3947]
R,120,0' 2 kopieren
90 MOVE 136,n:|WDRAWR,120,0,0:NEXT          [1731]
100 FOR n=360 TO 240 STEP-2:MOVE 8,n:|TDRA  [3035]
WR,120,0' 3 kopieren
110 MOVE 264,n:|WDRAWR,120,0,0:NEXT         [1748]
120 FOR n=392 TO 512 STEP 2:MOVE n-384,360  [4459]
:|TDRAWR,0,-120' 4 kopieren
130 MOVE n,360:|WDRAWR,0,-120,0:NEXT        [1371]
140 FOR n=640 TO 520 STEP -2:MOVE n-512,36  [2742]
0:|TDRAWR,0,-120' 5 kopieren
150 MOVE n,360:|WDRAWR,0,-120,0:NEXT        [1371]
160 LOCATE 14,2:PEN 1:PRINT"Super-Grafik"   [2025]
170 REM Schriftzug                          [1176]
180 |GMODUS,0:f=1:y=220:FOR i=0 TO 2        [1731]
190   FOR n=382 TO 368 STEP-2               [699]
200     |FILL,0,0,640:' Puffer leeren       [2117]
210     MOVE 208,n:|TDRAWR,192,0:|RECHTS,1  [3361]
0,0,630:|FILL,0,0,10
220     |WIDTH,1,2^i:FOR m=1 TO i:|BREIT:|  [3334]
ENG,5,0:NEXT
230     FOR j=0 TO 610 STEP 30:|WANDEL,1,f  [3396]
,j,30
240       f=f+1:IF f=4 THEN f=1             [855]
250     NEXT                                [350]
260     MOVE 0,y:|WDRAWR,620,0,0:y=y-(2^i)  [2147]
*2:NEXT
270   y=y+2^i-32:NEXT i                     [1418]
280 LOCATE 14,2:PEN 1:PRINT SPACE$(12)      [1698]
290 LOCATE 6,25:PEN 1:PRINT"< Bitte eine T  [3997]
aste druecken >"
300 |REIN,62,77,11,18,1,1                   [1492]
310 WHILE INKEY$=""                         [1095]
320 FOR n=1 TO 15:CALL &BD19:FOR m=0 TO 3:  [3356]
NEXT
330   |REIN,47+n,61+n,11,18,1,1' jeweils a  [4445]
us 2. Schirm holen
340 NEXT n                                  [366]
350 WEND                                    [390]
360 REM                                     [272]
370 REM verschiedene Schriften              [1043]
380 REM                                     [272]
390 MODE 1:INK 1,24:INK 2,18:INK 3,6        [1945]
400 WINDOW#1,1,40,1,2:PEN#1,1:PAPER#1,3:CL  [2655]
S#1
410 PRINT#1,TAB(14);"Schriftarten"          [1664]
420 PRINT#1,"Schriftzug :";:LINE INPUT#1,s  [2987]
$
430 s$=LEFT$(s$,15):CLS:PRINT s$:|WIDTH,0,  [2167]
0
440 REM spiegelnde Schrift                  [1571]
450 x=10:y=350:FOR n=384 TO 398 STEP 2      [1792]
460   nn=n-384:MOVE 0,n:|TDRAWR,240,0       [2403]
470   MOVE x,y+nn:|WDRAWR,240,0,0:|WANDEL,  [2256]
1,3,0,240
480   MOVE x-(nn/2),y-nn:|WDRAWR,240,0,0    [2851]
490 NEXT                                    [350]
500 REM 3-D Schrift                         [910]
510 |GMODUS,4 ' NORCE - Modus !             [1443]
520 x=10:y=300:FOR n=384 TO 398 STEP 2      [1601]
530   nn=n-384:MOVE 0,n:|TDRAWR,240,0       [2403]
540   MOVE x,y+nn:|WDRAWR,240,0,0           [1432]
550   |WANDEL,1,3,0,240:MOVE x-2,y+nn-2:|W  [3779]
DRAWR,240,0,0
560 NEXT:|GMODUS,0                          [1119]
570 REM gestreift, senkrecht                [2378]
580 f=1:x=10:y=270:FOR n=0 TO 240 STEP 2    [2148]
590   MOVE n,382:|TDRAWR,0,18:|WANDEL,1,f,  [3593]
0,20
600   MOVE x+n,y:|WDRAWR,0,18,0             [840]
610   IF f=1 THEN f=3 ELSE f=1              [1088]
```

```
620 NEXT                                          [350]
630 REM gestreift, waagerecht                     [2474]
640 f=1:x=10:y=240:FOR n=384 TO 398 STEP 2        [1248]
650   MOVE 0,n:|TDRAWR,240,0:|WANDEL,1,f,0        [2351]
,240
660   MOVE x,y+n-384:|WDRAWR,240,0,0              [1727]
670   IF f=1 THEN f=3 ELSE f=1                    [1088]
680 NEXT                                          [350]
690 REM fett                                      [726]
700 |GMODUS,4:x=10:y=210:FOR n=384 TO 398         [2276]
STEP 2
710   MOVE 0,n:|TDRAWR,240,0                      [1702]
720   MOVE x,y+n-384:|WDRAWR,240,0,0              [1727]
730   MOVE x+2,y+n-384:|WDRAWR,240,0,0            [1504]
740 NEXT:|GMODUS,0                                [1119]
750 REM umgedreht                                 [863]
760 x=10:y=180:FOR n=384 TO 398 STEP 2            [2041]
770   MOVE 0,n:|TDRAWR,240,0                      [1702]
780   |DREHE,0,120                                [1880]
790   MOVE x,y+n-384:|WDRAWR,240,0,0              [1727]
800 NEXT                                          [350]
810 REM auf dem Kopf                              [364]
820 x=10:y=166:FOR n=384 TO 398 STEP 2            [3039]
830   MOVE 0,n:|TDRAWR,240,0:MOVE x,y-(n-3        [3677]
84):|WDRAWR,240,0,0
840 NEXT                                          [350]
850 REM kursiv                                    [977]
860 x=10:y=120:FOR n=384 TO 398 STEP 2            [2446]
870   MOVE 0,n:|TDRAWR,240,0                      [1702]
880   MOVE x+n-384,y+n-384:|WDRAWR,240,0,0        [2002]
890 NEXT                                          [350]
900 REM eng                                       [500]
910 x=10:y=90:FOR n=384 TO 398 STEP 2             [3378]
920   MOVE 0,n:|TDRAWR,240,0:|ENG,1,0             [2081]
930   MOVE x,y+n-384:|WDRAWR,120,0,0              [1599]
940 NEXT                                          [350]
950 REM breit                                     [618]
960 x=10:y=60:FOR n=384 TO 398 STEP 2             [2773]
970   MOVE 0,n:|TDRAWR,240,0:|BREIT               [1869]
980   MOVE x,y+n-384:|WDRAWR,240,0,0              [1727]
990 NEXT                                          [350]
1000 REM hoch                                     [820]
1010 x=10:y=10:FOR n=384 TO 399                   [1989]
1020   MOVE 0,n:|TDRAWR,240,0                     [1702]
1030   MOVE x,y+(n-384)*2:|WDRAWR,240,0,0         [1494]
1040 NEXT                                         [350]
1050 REM nach oben                                [995]
1060 x=330:y=350:FOR n=384 TO 398 STEP 2          [2040]
1070   MOVE 0,n:|TDRAWR,240,0                     [1702]
1080   MOVE x,y+n-384:|WDRAWR,240,29,0            [2064]
1090 NEXT                                         [350]
1100 REM nach unten                               [972]
1110 x=330:y=330:FOR n=384 TO 398 STEP 2          [2486]
1120   MOVE 0,n:|TDRAWR,240,0                     [1702]
1130   MOVE x,y+n-384:|WDRAWR,240,-28,0           [1878]
1140 NEXT                                         [350]
1150 REM mini                                     [802]
1160 |FILL,0,0,640:x=330:y=290:FOR n=384 T        [3508]
O 398 STEP 4
1170   MOVE 0,n:|TDRAWR,240,0:|ENG,1,0            [2081]
1180   MOVE x,y+(n-384)/2:|WDRAWR,240,0,0         [2166]
1190 NEXT                                         [350]
1200 REM kariert + gross                          [1677]
1210 f=1:ff=3:x=330:y=250:FOR n=384 TO 399        [2293]
1220   MOVE 0,n:|TDRAWR,240,0                     [1702]
1230   FOR m=0 TO 120 STEP 8                      [1164]
1240     |WANDEL,1,f,m,4:|WANDEL,1,ff,m+4,       [3009]
4:NEXT
1250   MOVE x,y+(n-384)*2:|WDRAWR,240,0,0         [1494]
1260   IF n MOD 4=1 THEN c=f:f=ff:ff=c            [2148]
1270 NEXT                                         [350]
1280 REM diagonal                                 [336]
1290 |FILL,0,0,240:x=330:y=230:FOR n=384 T        [3638]
O 398 STEP 2
1300   MOVE 0,n:|TDRAWR,240,0                     [1702]
1310   MOVE x+(n-384)/2,y+n-384:|WDRAWR,24        [3066]
0,-120,0
1320   MOVE x+(n-384)/2+1,y+n-384+1:|WDRAW        [3314]
R,240,-120,0
1330 NEXT                                         [350]
1340 REM nach unten                               [972]
1350 |FILL,0,0,240:x=280:y=300:FOR n=384 T        [3684]
O 398 STEP 2
1360   MOVE 0,n:|TDRAWR,240,0:|DREHE,0,120        [2281]
1370   MOVE x+n-384,y:|WDRAWR,0,-240,0            [2813]
1380 NEXT                                         [350]
1390 REM geschwungen                              [2564]
1400 f=2:|FILL,0,0,240:|GMODUS,4:RESTORE 1        [2882]
410
```

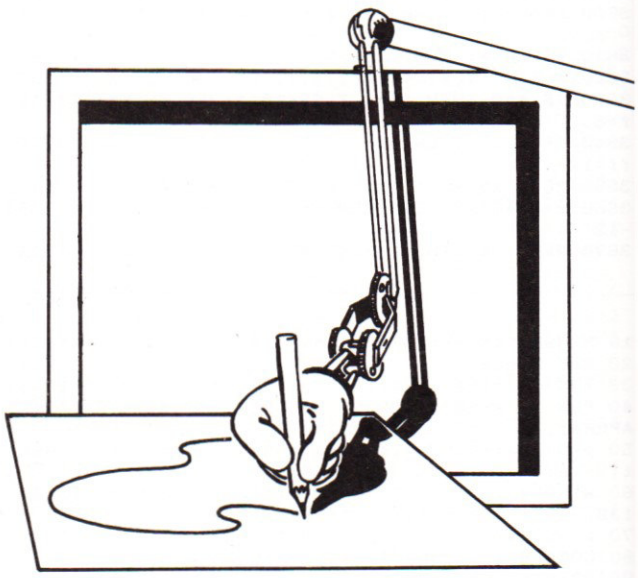
```
1410 DATA 2,3,5,10,0                              [550]
1420 DEG:FOR m=300 TO 285 STEP -5:READ mm         [2107]
1430   x=SIN(m)*300+640:y=COS(m)*100+40           [2141]
1440   f=f+1:IF f=3 THEN f=1                      [1134]
1450   FOR n=384 TO 398 STEP 2                    [1180]
1460     MOVE 0,n:|TDRAWR,240,0:|ENG,mm,0:        [3449]
|WANDEL,1,f,0,240
1470     MOVE x,y+n-384:|WDRAWR,240,0,0           [1727]
1480   NEXT:NEXT:PLOT -2,-2,3:TAG:MOVE 344        [3617]
,73:PRINT s$;:TAGOFF
1490 REM invers                                   [1145]
1500 f=1:x=330:y=10:FOR n=382 TO 398 STEP         [2311]
2
1510   MOVE 0,n:|TDRAWR,240,0:|WANDEL,0,3,        [4185]
0,240:|WANDEL,1,0,0,240
1520   |RECHTS,1,0,240:|BOKE,0,3                  [1002]
1530   |WANDEL,3,f,0,240:IF f=1 THEN f=3 E        [3978]
LSE f=1
1540   MOVE x,y+n-382:|WDRAWR,240,0,0:NEXT        [2737]
1550 PLOT x,y+n-382:DRAWR 240,0,3                 [1614]
1560 WHILE INKEY$="":WEND                         [1607]
1570 REM                                          [272]
1580 REM  Flaechenveraenderung                    [1278]
1590 REM                                          [272]
1600 DEG:MODE 1:x=80:y=318:INK 1,1:INK 2,1        [3335]
:INK 3,1
1610 BORDER 0:GOSUB 2960:|GMODUS,0                [2180]
1620 FOR n=0 TO 140 STEP 20:MOVE 0,400-n:D        [3967]
RAWR 200,0,0:NEXT
1630 FOR n=0 TO 140 STEP 20:MOVE n,0:DRAWR        [4153]
 0,400:NEXT
1640 |RAUS:CLS:INK 1,24:INK 2,0:INK 3,6           [1842]
1650 FOR n=6 TO 396 STEP 8:MOVE 0,n:DRAWR         [3394]
640,0,2:NEXT
1660 FOR n=6 TO 636 STEP 8:MOVE n,0:DRAWR         [3001]
0,400:NEXT
1670 |GMODUS,4:REM Extra - Grafikmodus ein        [4523]
schalten ("NORCE")
1680 REM Tuer                                     [493]
1690 FOR n=0 TO 60 STEP 4:|SCR2:MOVE n+20,        [3308]
379:|TDRAWR,0,-120
1700   |SCR1:MOVE n/2+20,360+n/3:|WDRAWR,0        [2555]
,-120,0:NEXT
1710 FOR n=60 TO 120 STEP 4:|SCR2:MOVE n+2        [3851]
0,379:|TDRAWR,0,-120
1720   |SCR1:MOVE 70+n/2,360+(120-n)/3:|WD        [4537]
RAWR,0,-120,0:NEXT
1730 REM Flagge                                   [562]
1740 FOR n=0 TO 120 STEP 2:|SCR2:MOVE n+20        [3082]
,379:|TDRAWR,0,-120
1750   |SCR1:MOVE n+170,378+SIN(n*6)*9:|WD        [3101]
RAWR,0,-120,0
1760 NEXT                                         [350]
1770 REM Flagge senkrecht                         [1448]
1780 FOR n=0 TO 120 STEP 2:|SCR2:MOVE 20,3        [2915]
78-n:|TDRAWR,120,0
1790   |SCR1:MOVE 330+SIN(n*6)*9,378-n:|WD        [3373]
RAWR,120,0,0
1800 NEXT                                         [350]
1810 REM Pos. SINUS                               [690]
1820 FOR n=0 TO 120 STEP 2:|SCR2:MOVE n+20        [3082]
,379:|TDRAWR,0,-120
1830   |SCR1:MOVE n+490,370+ABS(SIN(n*9)*1        [2976]
6):|WDRAWR,0,-120,0
1840 NEXT                                         [350]
1850 v%=0:REM Kugel                               [848]
1860 FOR n=-58 TO 58 STEP 2:|SCR2:MOVE 20,        [4220]
318+n:|TDRAWR,120,0
1870   |SCR1:|LEN,@v%:|FILL,1,v%-1,200:x=S        [5788]
QR(3600-n*n):f=(60-x)*2:GOSUB 2310
1880   |FILL,0,x,200:MOVE 80-x,120+n:|WDRA        [2996]
WR,120,0,0
1890 NEXT                                         [350]
1900 REM Scheren + Flagge                         [2129]
1910 FOR n=0 TO 120 STEP 2:|SCR2:MOVE n+20        [3082]
,379:|TDRAWR,0,-120
1920   |SCR1:MOVE n+170,(190-n/2)+SIN(n*10        [3376]
)*12:|WDRAWR,0,-120,0
1930 NEXT                                         [350]
1940 |FILL,0,0,640:REM Diagonal - Sinus           [2461]
1950 FOR n=0 TO 240 STEP 2                        [999]
1960   IF n<120 THEN |SCR2:MOVE 20,380-n:|        [3786]
TDRAWR,200,200:GOTO 1980
1970   |SCR2:MOVE n-100,260:|TDRAWR,200,20        [3933]
0
1980   s=SIN(n*6)*7                               [602]
1990   IF n<120 THEN |SCR1:MOVE 340+s,150-        [3566]
n+s:|WDRAWR,200,200,0:GOTO 2010
2000   |SCR1:MOVE 220+n+s,30+s:|WDRAWR,200        [3045]
```

```
,200,0
2010 NEXT                                          [350]
2020 REM Zick-Zack-Muster ("Eierkarton")           [4327]
2030 FOR n=0 TO 120 STEP 2:|SCR2:MOVE 20,3         [2915]
78-n:|TDRAWR,120,0
2040   x=SIN(n*10)*6+6                             [1582]
2050   MOVE 490+x,182-n:|WDRAWR,120,0,0 'X         [3970]
-Zick-Zack im 2. Bildschirm
2060 NEXT                                          [350]
2070 FOR n=0 TO 132 STEP 2:|SCR2:MOVE 490+         [3222]
n,182:|TDRAWR,0,-120
2080   |SCR1:x=SIN(n*10)*6+6                       [1878]
2090   MOVE 490+n,152-x:|WDRAWR,0,-120,0 '         [3402]
Y-Zick-Zack auf'm 1.
2100 NEXT                                          [350]
2110 REM Klein in der Mitte                        [1256]
2120 FOR n=2 TO 122 STEP 4:|SCR2:MOVE 20,3         [3440]
78-n:|TDRAWR,120,0
2130   |SCR1:|ENG,1,0                             [1340]
2140   MOVE 330,230-n/2:|WDRAWR,60,0,0             [1734]
2150 NEXT:FOR n!=0 TO 120 STEP 6.6:|SCR2:M         [4204]
OVE 20,378-n!:|TDRAWR,120,0
2160   |SCR1:|ENG,1,0:|ENG,3,0                     [2061]
2170   MOVE 420,220-n!/2.6:|WDRAWR,60,0,0          [2106]
2180 NEXT:FOR n=2 TO 122 STEP 8:|SCR2:MOVE         [3956]
 20,378-n:|TDRAWR,120,0
2190   |SCR1:|ENG,1,0:|ENG,1,0                     [2005]
2200   MOVE 490,210-n/4:|WDRAWR,30,0,0             [1908]
2210 NEXT:FOR n=0 TO 120 STEP 16:|SCR2:MOV         [6266]
E 20,378-n:|TDRAWR,120,0
2220   |SCR1:|ENG,1,0:|ENG,1,0:|ENG,1,0           [2543]
2230   MOVE 540,204-n/8:|WDRAWR,15,0,0             [987]
2240 NEXT:FOR n=0 TO 120 STEP 32:|SCR2:MOV         [5260]
E 20,378-n:|TDRAWR,120,0
2250   |SCR1:|ENG,1,0:|ENG,1,0:|ENG,1,0:|E        [2972]
NG,1,0
2260   MOVE 580,200-n/16:|WDRAWR,15,0,0            [870]
2270 NEXT                                          [350]
2280 PLOT -2,-2,1:MOVE 400,16:TAG:PRINT"-          [3151]
Eine Taste -";:TAGOFF
2290 WHILE INKEY$="":WEND:|GMODUS,0:GOTO 2         [2570]
370
2300 REM Buffer schrumpfen lassen  (f Punk         [3504]
te wegfallen lassen !)
2310 p=120:IF f=0 THEN RETURN                      [1827]
2320 a=p/f:IF a<2 THEN |ENG,1,0:p=p/2:f=f-         [4175]
p:GOTO 2320
2330 |ENG,a-1,0:RETURN                             [1806]
2340 REM                                           [272]
2350 REM Fill - Demo                               [422]
2360 REM                                           [272]
2370 WINDOW#1,1,40,1,2:PAPER#1,2:PEN#1,1:B         [3264]
ORDER 1:|WIDTH,2,2
2380 CLS:FOR n=0 TO 20:x1=RND*640:x2=RND*6         [4398]
40:MOVE x1,2:DRAW x2,368,RND*2+1:NEXT
2390 |WIDTH,0,0:MOVE 0,0:DRAWR 640,0,1:MOV         [2830]
E 0,0:DRAWR 0,400:MOVE 638,0
2400 DRAWR 0,400:|RAUS:x=10:y=250:CLS#1:PR         [3179]
INT#1,"              Raster-Fill"
2410 PRINT#1," Definieren Sie Ihr Raster !         [4698]
":GOSUB 3270:x=160:y=100:f=0
2420 |REIN:CLS#1:PRINT#1," Wohin soll es k         [4727]
ommen ?"
2430 |PFEIL,0,320,185,0,242,243,240,241,0,         [3314]
3,3,@x,@y,@f
2440 MOVE x*2,y*2:|RFILL,@a$                       [1147]
2450 CLS#1:PRINT#1," Noch einmal (J/N) ?":         [2375]
GOSUB 3220
2460 IF f<>1 THEN |RAUS:GOTO 2420                  [869]
2470 CLS#1:PRINT#1," Naechster Demo (J/N)          [2927]
 ?":GOSUB 3220
2480 IF f=1 THEN 2400                              [917]
2490 REM                                           [272]
2500 REM Zoom - Demo                               [1161]
2510 REM                                           [272]
2520 |REIN:CLS#1:PRINT#1,"                         [3387]
 Zoomeditor"
2530 PRINT#1," Selektieren Sie: untere, li         [3852]
nke Ecke"
2540 DEFINT x,y,f:x=160:y=100:f=0:ORIGIN 0         [3169]
,0
2550 |PFEIL,0,320,185,0,242,243,240,241,0,         [3314]
3,3,@x,@y,@f
2560 PRINT#1,"                    obere, rec       [3311]
hte Ecke"
2570 x1=x*2:y1=y*2:|PFEIL,0,320,185,0,242,         [3653]
243,240,241,0,3,3,@x,@y,@f
2580 x2=x*2:y2=y*2:IF x1>=x2 OR y1>=y2 THE         [3262]
N 2500
2590 xa=x2-x1:ya=y2-y1:IF 630/xa<=2 OR 390         [4861]
/ya<=2 THEN 2500
2600 z=0:zz=xa:WHILE zz<640:zz=zz*2:z=z+1:         [4409]
WEND:xz=z-1:ax=2^xz
2610 z=0:zz=ya:WHILE zz<400:zz=zz*2:z=z+1:         [3057]
WEND:yz=z-1:ay=2^yz
2620 IF xa*ax+ax>=600 THEN xz=xz-1:ax=2^xz         [4409]
' ueberpruefen ob width am Rand!
2630 IF ya*ay+ay>=360 THEN yz=yz-1:ay=2^yz         [2269]
2640 REM Kopieren                                  [869]
2650 xx=320-(xa*ax+ax)/2:yy=200-(ya*ay+ay)         [2870]
/2
2660 MODE 1:|WIDTH,1,ay:x=xx:y=398-yy:FOR          [4877]
n=y2 TO y1 STEP-2
2670   |FILL,0,0,640                              [1554]
2680   |SCR2:MOVE x1,n:|TDRAW,x2,n:|SCR1          [3774]
2690   FOR m=1 TO xz:|BREIT:NEXT:|BOKE,0,0        [2407]
2700   MOVE x,y:|WDRAWR,640-ax*2,0,0:y=y-a         [3872]
y*2:NEXT
2710 REM editieren                                 [794]
2720 BORDER 2:x=ax/2+xx/2:y=((200-ay/2)-yy         [4185]
/2)-1:xg=xa*ax/2+xx/2
2730 yg=(200-ya*ay/2-ay)-yy/2                      [1980]
2740 |WIDTH,2,2:PLOT xx-4,404-yy,1:DRAW  x         [7348]
g*2+4,404-yy:DRAW xg*2+4,yg*2-6:DRAW xx-4,
yg*2-6:DRAW xx-4,404-yy:|WIDTH,ax,ay
2750 IF ax=2 THEN x=x-1                            [956]
2760 IF ay=2 THEN y=y-1                            [514]
2770 |PFEIL,xx/2,xg,200-yy/2,yg,242,243,24         [3046]
0,241,0,ax,ay,@x,@y,@f
2780 IF UPPER$(CHR$(f))="Q" THEN 2810              [1968]
2790 t=TEST(x*2,y*2):t=t+1:IF t=4 THEN t=0         [1339]
2800 PLOT x*2-ax+2,y*2+ay,t:GOTO 2770              [1466]
2810 REM Zurueckbringen                            [683]
2820 |WIDTH,0,0:x=xx+2:y=398-yy:FOR n=y2 T         [2844]
O y1 STEP-2
2830   |FILL,0,0,640                              [1554]
2840   |SCR1:MOVE x,y                             [917]
2850   |TDRAWR,640,0                              [564]
2860   |SCR2                                      [922]
2870     FOR m=1 TO xz:|ENG,1,0:NEXT              [1775]
2880   MOVE x1+2,n:|WDRAW,x2,n,0                  [2026]
2890 y=y-ay*2:NEXT:|SCR1                          [1124]
2900 |REIN:WINDOW #1,1,40,1,2:PEN #1,3:PAP        [2676]
ER#1,2
2910 PRINT#1,"Noch einmal (J/N)  ?                 [3105]
":PRINT#1
2920 GOSUB 3220:IF f=0 THEN 2520                   [1379]
2930 REM Programmende                              [984]
2940 MODE 1:PEN 1:LOCATE 17,12:PRINT"E N D         [3600]
 E"
2950 GOTO 2950                                     [381]
2960 REM Unterprogramme                            [718]
2970 REM Schneider-Zeichen an x/y Position         [3519]
 auf den Schirm bringen
2980 REM                                           [272]
2990 ORIGIN x,y:DEG:|WIDTH,0,0                     [1241]
3000 PLOT -60,-56,1:DRAWR 120,0:DRAWR 0,11         [2773]
6:DRAWR -120,0:DRAWR 0,-116
3010 PLOT -4,10,2:DRAWR -16,0                      [949]
3020 FOR n=180 TO 360 STEP 10                      [1285]
3030   DRAW SIN(n)*14-20,COS(n)*14+24:NEXT         [2308]
3040 DRAWR 16,0:DRAWR 0,16:DRAWR -16,0             [1368]
3050 FOR n=360 TO 180 STEP-10                      [942]
3060   DRAW SIN(n)*30-20,COS(n)*30+24:NEXT         [730]
3070 DRAWR 16,0:DRAWR 0,16                         [940]
3080 PLOT 4,-6,3:DRAWR 16,0                        [467]
3090 FOR n=0 TO 180 STEP 10                        [1232]
3100   DRAW SIN(n)*14+20,COS(n)*14-20:NEXT         [1907]
3110 DRAWR -16,0:DRAWR 0,-16:DRAWR 16,0            [1517]
3120 FOR n=180 TO 0 STEP-10                        [1047]
3130   DRAW SIN(n)*30+20,COS(n)*30-20:NEXT         [616]
3140 DRAWR -16,0:DRAWR 0,-16                       [777]
3150 REM jetzt auffuellen                          [1491]
3160 a1$=STRING$(12,1)+STRING$(144,240)+ST         [2196]
RING$(12,1): REM ink 1
3170 a2$=STRING$(12,1)+STRING$(144,15)+STR         [2927]
ING$(12,1) : REM ink 2
3180 a3$=STRING$(12,1)+STRING$(144,255)+ST         [4000]
RING$(12,1): REM ink 3
3190 MOVE 0,0:|RFILL,@a1$:MOVE -10,20:|RFI         [2868]
LL,@a1$:MOVE 10,-20:|RFILL,@a1$
3200 MOVE -10,0:|RFILL,@a2$:MOVE 10,0:|RFI         [2605]
LL,@a3$
3210 ORIGIN 0,0:RETURN                             [456]
3220 REM Abfrage j/n                               [746]
3230 r$=UPPER$(INKEY$):IF r$="" THEN 3230          [2409]
3240 IF r$="J" THEN f=0:RETURN                     [1280]
3250 IF r$="N" THEN f=1:RETURN ELSE 3230           [2255]
```

```
3260 x=320:y=300                                        [998]
3270 REM Raster - Designer (an Stelle x/y               [2498]
in String a$)
3280 IF LEN(a$)<>168 THEN a$=STRING$(12,1)              [3231]
+STRING$(144,0)+STRING$(12,1)
3290 mx=x:my=y:DEFINT x,y,f:|WIDTH,0,0                  [2600]
3300 |WIDTH,82,112:PLOT mx,my,0:|WIDTH,0,0              [4449]
 'Fenster loeschen (!)
3310 REM Bildschirmaufbau                               [2442]
3320 PLOT mx,my,1:DRAWR 164,0:DRAWR 0,-224              [4146]
:DRAWR -164,0:DRAWR 0,224
3330 PLOT mx,my:DRAWR 6,-6:DRAWR 104,0:DRA              [4672]
WR 6,6:DRAWR 0,-60:DRAWR -6,6
3340 DRAWR 0,48:MOVER 0,-48:DRAWR -104,0:D              [3355]
RAWR -6,-6:DRAWR 164,0
3350 PLOT mx+6,my-6:DRAWR 0,-48:MOVER -6,-              [2607]
6:DRAWR 8,-8:DRAWR 148,0
3360 DRAWR 8,8:MOVER -8,-8:DRAWR 0,-148:DR              [2585]
AWR 8,-8:MOVER -8,8
3370 DRAWR -148,0:DRAWR -8,-8:MOVER 8,8:DR              [2315]
AWR 0,148
3380 ff=1:MOVE mx+116,my:DRAWR 6,-6:DRAWR               [3832]
36,0:DRAWR 6,6:MOVER -6,-6
3390 DRAWR 0,-18:DRAWR 6,-6:MOVER -6,6:DRA              [4246]
WR -36,0:DRAWR -6,-6:MOVER 6,6
3400 DRAWR 0,18:MOVE mx+116,my-30:DRAWR 48              [2206]
,0:GOSUB 3530
3410 PLOT 1000,1000,3:TAG:MOVE mx+118,my-3              [2926]
8:PRINT"QC";
3420 MOVE mx+148,my-38:PRINT"T";                        [1709]
3430 f=0:x=(mx+16)/2:y=(my-76)/2:l=(mx+10)              [3721]
/2:r=(mx+154)/2:o=(my-70)/2
3440 u=(my-214)/2:|GMODUS,0:GOSUB 3600:ff=              [2074]
1
3450 REM Tastaturabfrage                                [1424]
3460 |PFEIL,l,r,o,u,242,243,240,241,0,6,6,              [3762]
@x,@y,@f
3470 t$=UPPER$(CHR$(f)):IF t$="Q" THEN RET              [2813]
URN
3480 IF t$="T" THEN FOR n=8 TO 52 STEP 2:M              [6604]
OVE mx+8,my-n:DRAWR 100,0,0:NEXT:MOVE mx+1
0,my-10:|RFILL,@a$:GOTO 3460
3490 IF t$="C" THEN GOSUB 3550:GOTO 3460                [2412]
3500 IF t$>="0" AND t$<"4" THEN ff=VAL(t$)              [2446]
:GOSUB 3530:GOTO 3460
3510 |WIDTH,5,5:PLOT x*2-4,y*2+4,ff:x1=(x*              [3960]
2-mx)/12:y1=(my-y*2)/12-5
3520 st=x1+y1*12:|ENCODE,ff,@f:MID$(a$,st+              [5287]
12,1)=CHR$(f):|WIDTH,0,0:GOTO 3460
3530 TAG:PLOT 1000,1000,3:MOVE mx+124,my-8              [3567]
:PRINT USING"##";ff;
3540 RETURN                                             [555]
3550 REM Editor loeschen                                [615]
3560 a$=STRING$(12,1)+STRING$(144,0)+STRIN              [4041]
G$(12,1):|WIDTH,72,72 '(!!!!)
3570 PLOT mx+10,my-70,0:|WIDTH,0,0:RETURN               [1779]
3580 PLOT x*2-4,y*2+4,0:NEXT y,x:|WIDTH,0,              [3566]
0:RETURN
3590 REM String a$ auf Bildschirm bringen               [2804]
3600 x5=mx+12:y5=my-72:st=25:|WIDTH,5,5:z=              [2761]
0
3610 WHILE st<169                                       [1026]
3620 a=ASC(MID$(a$,st,1)):ff=0                          [2566]
3630 IF a=0 THEN ff=0 ELSE IF a=255 THEN f              [1717]
f=3
3640 IF a=15 THEN ff=2 ELSE IF a=240 THEN               [2329]
ff=1
3650 PLOT x5,y5,ff:x5=x5+12:st=st+1                     [2394]
3660 z=z+1:IF z=12 THEN z=0:x5=mx+12:y5=y5              [1735]
-12
3670 WEND:|WIDTH,0,0:RETURN                             [1111]
```

```
10 MEMORY &8FFF:LOAD "probe",&9000                      [1802]
20 REM Menue                                            [645]
30 INK 0,1:INK 1,24:INK 2,6:INK 3,26                    [1946]
40 PEN 1:PAPER 0:BORDER 0:MODE 1:PEN#1,1:P              [2241]
APER#1,2
50 p=0:start=36864:ende=41108:le=ende-star              [2990]
t:x=12:y=12
60 WINDOW 1,40,3,25:WINDOW#1,1,40,1,2:KEY               [3164]
139,"mode 1:goto 11040"+CHR$(13)
70 WINDOW#2,3,38,24,24:PEN#2,0:PAPER#2,1                [2500]
80 LOCATE#1,1,1:LOCATE#2,1,1:LOCATE 1,1                 [2495]
90 PRINT#1,USING"Start: & End: & Himem: &"              [4419]
;HEX$(start,4);HEX$(ende,4);"AB7B / A677";
100 IF p=0 THEN p$="off" ELSE p$="on "                  [2512]
110 PRINT#1.USING"          Printerstate  :             [3198]
   &          ";p$;
120 CLS:PEN 2:LOCATE 1,3:PRINT TAB(8);CHR$              [3813]
(150);STRING$(24,154);CHR$(156)
130 GOSUB 320                                           [1012]
140 a$="1 - Gbasic verschieben":GOSUB 330               [2988]
150 GOSUB 320                                           [1012]
160 a$="2 - Merkzettel":GOSUB 330                       [1910]
170 GOSUB 320                                           [1012]
180 a$="3 - Rastergroesse":GOSUB 330:GOSUB              [3312]
 320
190 a$="4 - Gbasic abspeichern":GOSUB 330               [3073]
200 GOSUB 320                                           [1012]
210 a$="C - Cat":GOSUB 330                              [731]
220 GOSUB 320                                           [1012]
230 a$="P - Printer on / off":GOSUB 330                 [2500]
240 GOSUB 320                                           [1012]
250 a$="E - Programmende":GOSUB 330:GOSUB               [2695]
320
260 PRINT TAB(8);CHR$(147);STRING$(24,154)              [2156]
;CHR$(153)
270 PRINT#2," Tippen Sie die entsprechende              [3542]
 Taste ";
280 PLOT 32,14,1:DRAWR 574,0:PLOT 32,32:DR              [2390]
AWR 574,0
290 a$="ECP1234":i$=UPPER$(INKEY$):IF i$="              [2403]
" THEN 290
300 t=INSTR(a$,i$):IF t=0 THEN 290                      [1067]
310 ON t GOSUB 350,370,390,420,690,850,950              [2389]
:GOTO 80
320 a$=""                                               [388]
330 PRINT TAB(8);CHR$(149);:PEN 1:PRINT US              [2075]
ING" \                    \ ";a$;
340 PEN 2:PRINT CHR$(149):RETURN                        [1787]
350 REM Programmende                                    [984]
360 MODE 2:PEN 1:END                                    [1168]
370 REM Catalog anzeigen                                [1213]
380 CLS:PEN 1:CAT:CALL &BB18:RETURN                     [395]
390 REM Drucker umstellen                               [1339]
400 IF p=0 THEN p=8 ELSE p=0                            [773]
410 RETURN                                              [555]
420 REM MC-Programm verschieben                         [2082]
430 CLS:PEN 1:st=start                                  [1077]
440 LOCATE 4,4:PRINT"Eingabe der Anfangsad              [3506]
resse (A)"
450 LOCATE 11,6:PRINT"oder der Endadresse               [1099]
(E) ?"
460 a$=UPPER$(INKEY$):IF a$="" THEN 460                 [1824]
470 IF a$<>"A" AND a$<>"E" THEN 460                     [1431]
480 LOCATE 3,10:IF a$="A" THEN INPUT "Neue              [4270]
 Anfangsadresse ";a
490 IF a$="E" THEN INPUT "Neue Endadresse               [1706]
";a
500 IF a<0 THEN a=a+65536                               [1042]
510 IF a$="E" THEN a=a-4244                             [1012]
520 IF a<10000 THEN PRINT:PRINT:PRINT"DAS               [5586]
GEHT NICHT !!!!!":FOR n=0 TO 3000:NEXT:RET
URN
530 n=MIN(a,start):MEMORY n-2                           [1281]
540 start=a:ende=a+4244:nst=a                           [1476]
550 PRINT:PRINT" Adressen werden angeglich              [4083]
en ":GOSUB 990
560 PRINT:PRINT" Das Programm wird versch               [2179]
oben"
570 IF off2=0 THEN RETURN                               [1837]
580 IF off2>0 THEN GOTO 640                             [1283]
590 REM Per Basic-"LDDR" verschieben                    [2016]
600 de=nst+4244:hl=st+4244:bc=4245                      [2277]
610 WHILE bc>0:bc=bc-1                                  [1074]
620 POKE de,PEEK(hl):de=de-1:hl=hl-1:WEND               [935]
630 MEMORY nst-2:RETURN                                 [1486]
640 REM Per Basic-"LDIR" verschieben                    [1790]
650 de=nst:hl=st:bc=4245                                [1135]
660 WHILE bc>0:bc=bc-1                                  [1074]
670 POKE de,PEEK(hl):hl=hl+1:de=de+1:WEND               [1467]
680 RETURN                                              [555]
690 REM Notizzettel auf Drucker / Bildschi              [3648]
rm
700 of=&90*256-start:b1=38595-of:b2=39235-              [3024]
of:dp=39941-of
710 CLS:PEN 1:LOCATE 1,1:PRINT#p                        [844]
720 PRINT#p," Anfang von GBASIC          ";H            [4980]
EX$(start,4);" /";start:PRINT#p
730 PRINT#p," Ende   von GBASIC          ";H            [3176]
EX$(ende,4);" /";ende:PRINT#p
740 PRINT#p," Bufferadresse bei RFILL    ";H            [5819]
EX$(b1,4);" /";b1:PRINT#p
750 PRINT#p," 1. Buffer bei T/Wdraw      ";H            [4137]
EX$(b1,4);" /";b1:PRINT#p
760 PRINT#p," 2. Buffer bei T/Wdraw      ";H            [4787]
```

```
EX$(b2,4);" /";b2:PRINT#p
770 PRINT#p," Adresse des Pfeiles      ";H [3719]
EX$(dp,4);" /";dp:PRINT#p
780 PRINT#p," Rastergroesse  (X)       ";x: [5154]
PRINT#p
790 PRINT#p," Rastergroesse  (Y)       ";y: [4123]
PRINT#p
800 PRINT#p," Stringformat bei RFILL / PRA [4270]
STER:":PRINT#p
810 PRINT#p,USING" a$=STRING$(##_,1)+";x   [3056]
820 PRINT#p,USING"     ### Zeichen Rasterda [3173]
ten";x*y
830 PRINT#p,USING"     +STRING$(##_,1)";x   [1924]
840 CALL &BB18:RETURN                      [1296]
850 REM Rastergroesse bei RFILL/PRASTER ei [1740]
nstellen
860 CLS:PEN 1:LOCATE 4,6:INPUT "X-Groesse  [2455]
bei PRASTER / RFILL ";xx
870 LOCATE 4,8:INPUT "Y-Groesse bei PRASTE [3681]
R / RFILL ";yy
880 IF xx*yy+xx*2>256 THEN PRINT"   Raster [6291]
 ist zu gross !!":FOR n=0 TO 3000:NEXT:RET
URN
890 x=xx:y=yy:of=&90*256-start:RESTORE 900 [3206]
:xy=x*y
900 DATA 9E48,9E4E,9E7F,9F89,9FD3,9FEB,A00 [1906]
6
910 FOR n=1 TO 7:READ a$:a=VAL("&"+a$)-of: [3204]
POKE a,x:NEXT
920 DATA 9E4D,9FC1,9FC8                    [909]
930 FOR n=1 TO 3:READ a$:a=VAL("&"+a$)-of: [4130]
POKE a,y:NEXT
940 POKE &9FF7-of,xy:POKE &A010-of,xy:RETU [2436]
RN
950 REM MC-programm abspeichern            [2614]
960 CLS:PEN 1:LOCATE 4,6:INPUT "Dateiname  [1389]
";d$
970 SAVE ""+d$,b,start,4245,0              [1896]
980 RETURN                                 [555]
990 REM MC-Programm (akt. Pos.: st neue Po [3462]
s.: nst) verschieben
1000 RESTORE 1070                          [721]
1010 REM                                   [272]
1020 REM Daten der zu veraendernden Adress [1365]
en
1030 REM !XX : laufendes High-Byte veraend [2229]
ern
1040 REM  XX : Adresse, nur Low-Byte davon [1484]
1050 REM ->  Adresse = laufendes High-Byte [2976]
 * 256 + Low-Byte
1060 REM                                   [272]
1070 DATA !90,04,07,0A,10,14,17,1A,1D,20,2 [2311]
3,26,29
1080 DATA 2C,2F,32,35,38,3B,3E,41,44,47,4A [2365]
,4D
1090 DATA 50,53,56,59,5C,5F,62,65,68,6B,6E [2520]
,!91,10
1100 DATA 12,1C,36,3C,40,46,4C,4F,55,5A,5D [2258]
,63,68
1110 DATA 7A,7D,8F,A1,A7,AC,B5,B8,BB,C1,CA [2565]
,CD,D4
1120 DATA D9,E2,E5,F2,F8,!92,02,07,0F,12,1 [2732]
C,20,44
1130 DATA 4E,55,60,66,78,7F,8E,91,A6,AC,C6 [1574]
,C9,CE
1140 DATA D7,DA,EA,FD,FF,!93,03,08,0B,10,1 [892]
9,23,2A
1150 DATA 2E,33,36,39,40,45,50,56,5F,65,68 [849]
,6B,6E
1160 DATA 73,79,7F,83,88,8B,8F,95,A7,AD,B5 [2465]
,BC,C1
1170 DATA C8,CD,D0,D6,DA,E0,!94,08,0B,2A,2 [2855]
E,35,40
1180 DATA 43,55,58,5B,7C,93,9C,A4,B7,C3,CB [2240]
,D4,DC
1190 DATA E7,EB,!95,02,05,0E,14,1A,1F,28,3 [2074]
3,36,3A,3D
1200 DATA 45,49,4C,50,53,56,5E,65,6C,71,79 [1898]
,85,8D
1210 DATA 93,9A,A1,A4,A8,AE,B5,BC,C1,C4,C8 [1487]
,CB,CF
1220 DATA D9,E1,EE,F2,FD,!96,08,1A,2D,38,6 [2504]
1,8A,B0
1230 DATA BB,C0,!9B,C4,C8,D5,!9C,16,2E,36,  [2610]
39,3D,43
1240 DATA 46,4A,51,54,5D,60,64,6D,70,74,77 [1849]
,7A,7D
1250 DATA 8C,90,96,9C,A2,AB,AE,B6,B9,BD,C6 [2402]
,C9,CF
1260 DATA D2,D5,DE,E1,E5,E8,EB,EE,F1,F5,F8 [1942]
,FF,!9D,13
1270 DATA 18,1B,20,23,29,2C,33,41,4E,5A,64 [1838]
,6B,6E
1280 DATA 75,78,7F,82,85,8A,8E,91,96,99,A6 [2413]
,AE,B1
1290 DATA B6,B9,BF,C5,D6,E5,ED,F1,F8,FD,!9 [3328]
E,0A,1D
1300 DATA 22,35,3A,3E,42,45,51,55,5A,60,63 [1349]
,67,6A
1310 DATA 71,74,7C,83,89,96,99,9C,9F,A8,AB [2348]
,B1,BA
1320 DATA CE,D3,E7,EE,!9F,06,0D,15,2B,33,3 [2585]
6,3D,40
1330 DATA 43,54,69,6D,83,86,8E,91,AB,AF,C5 [1634]
,CC,D7
1340 DATA DD,E1,E8,FC,!A0,03,17,22,33,42,5 [2627]
E,67,76
1350 DATA 8A                               [254]
1360 ps=0:bst=&90*256:off1=bst-st:off2=st- [1804]
nst
1370 FOR n=1 TO 353                        [1039]
1380 READ a$:IF LEN(a$)=3 THEN hb=VAL("&"+ [4041]
MID$(a$,2)):na=0:GOTO 1380
1390 adr=VAL("&"+a$)+hb*256:IF na>=adr THE [3468]
N PRINT"Fehler in den Datas !":END
1400 na=adr:b=adr-off1:a=PEEK(b)+PEEK(b+1) [3060]
*256:a=a-off2
1410 h=INT(a/256):POKE b,a-h*256:POKE b+1, [2567]
h:NEXT
1420 RETURN                                [555]
2000 REM ********************************  [1576]
2010 REM *                              *  [403]
2020 REM *           GTUNER             *  [693]
2030 REM *                              *  [403]
2040 REM *    Verschiebt und patcht     *  [2166]
2050 REM *     "PROBE.BIN" (GBASIC)     *  [1786]
2060 REM *                              *  [403]
2070 REM *  COPYRIGHT SEPTEMBER 1986     *  [1753]
2080 REM *                              *  [403]
2090 REM *       BY  DIETER BRAUN       *  [2264]
2100 REM *                              *  [403]
2110 REM ********************************  [1576]
```



# CPC-TERM

## Das Terminalprogramm für die V/24-Schnittstelle der Schneiderware #3

Endlich verfügen die Besitzer einer Schneiderware #3 – V/24 (CPC 8/86) über ein leistungsfähiges Terminalprogramm für die Datenfernübertragung (DFÜ). CPC-TERM ermöglicht die Kommunikation mit Mailboxen oder mit beliebigen anderen Computern über ein Modem, aber auch die direkte Kopplung zweier Computer mit einem sogenannten Nullmodem-Kabel. Hingewiesen sei hier darauf, daß CPC-TERM in der vorliegenden Form nur mit der Schneiderware-V/24 arbeitet, da die RSX-Befehle auf die Adressdekodierung dieser Schnittstelle abgestimmt sind. Kundigen dürfte eine Anpassung jedoch leicht möglich sein. Zunächst einige Hinweise zum Abtippen: Zuerst CPC-TERM (Listing 1) eingeben und ohne zu starten abspeichern. Achtung: Der ON BREAK GOSUB – Sprung in Zeile 360 kann bei Tippfehlern den Rechner ins Nirvana schicken. Danach den Datalader des Verwaltungsprogramms (Listing 2) eingeben. Dieser generiert die Maschinencode-Datei MAIL.OBJ, die von CPC-TERM nachgeladen wird. Kassettenbenutzer müssen die Dateien unbedingt in der angegebenen Reihenfolge abspeichern.

### Die ersten Gehversuche

Für DFÜ-Neulinge zuerst einmal eine kleine Einführung in die Praxis der DFÜ:
Sicher haben Sie schon (bei ausgeschaltetem Computer hoffentlich) die V/24 angeschlossen. Falls Sie nicht auch über das Schneiderware-Netzteil verfügen, müssen Sie noch die Pegelspannungen an die Grundplatine anlegen. Spannungsquellen können zwei 9VBlöcke sein; ein Plus-Pol wird mit dem Minus-Pol des anderen Blocks verbunden und mit dem Pin GND auf der Grundplatine. Die beiden anderen Anschlüsse werden entsprechend ihrer Polarität an die Pins – bzw. +12V angelegt. Verbinden Sie jetzt Ihr Modem mit der V/24. Das Modem auf »originate« schalten und falls einstellbar, die Baudrate auf 300. Legen Sie den Telefonhörer (richtigherum) ins Modem. Und natürlich das Einschalten nicht vergessen! Jetzt brauchen Sie noch die Telefonnummer einer Mailbox, die Sie anrufen wollen. Sicherlich steht in dem Verzeichnis, das Sie möglicherweise haben, hinter der Nr. eine Buchstaben-Zahlen-Kombination. Lautet diese 8N1, so können Sie diese Mailbox sofort anwählen. Ansonsten müssen Sie die Übertragungsparameter ändern: Drücken Sie zweimal ESC und rechts oben erscheint ein MENÜ-Fenster, »1« für Parameter drücken und mit ENTER bestätigen.
Zuerst wird die Baudrate abgefragt. Da diese bereits auf 300 Baud eingestellt ist, brauchen Sie hier nur ENTER zu drücken. Das führt dazu, und das gilt für alle Untermenüs, daß die vorherige Einstellung erhalten bleibt. Für die nächsten Fragen müssen wir zunächst das dreistellige Kürzel entziffern:

8N1
8 N 1 8 Datenbits No parity
1 Stopbit

Unsere Schnittstelle kann auf 7 oder 8 Datenbits eingestellt werden. Das »N« in der Mitte steht für No parity, also keine Parität. Hier könnte auch noch »E« für Even oder »O« für Odd, also gerade bzw. ungerade Parität stehen. (Zur genaueren Erklärung u.a. der Parität sei hier auf den Lehrgang »Mein Computer unterhält sich« ab Heft 2/87 verwiesen.) Das Parametermenü fragt zunächst ob die Parität aus oder ein (0/1) sein soll. Erst wenn sie eingeschaltet wurde, wird nach (E)ven/(O)dd gefragt, wobei Even der voreingestellte Wert ist. Stoppbits kann unsere Schnittstelle 1 oder 2 verarbeiten.
Die nächsten beiden Fragen können in aller Regel mit ENTER beantwortet werden, da hier die für den Mailbox-Verkehr üblichen Parameter voreingestellt sind. Nun sollte einer Verbindung nichts mehr im Wege stehen.
Die Tel.-Nr. der Mailbox wählen, und spätestens, wenn ein Pfeifton hörbar wird, den Hörer ins Modem legen, die Mailbox hat sich gemeldet. Nach einer kurzen Wartezeit sollte auf dem Monitor eine Meldung der Mailbox erscheinen. Ist dies nicht der Fall, so drücken Sie bitte ein paarmal ENTER, manche Mailboxen erwarten dies als Signal zum Senden. Üblicherweise erscheint dann ein Auswahl-Menü, aus dem hervorgeht, was die Mailbox zu bieten hat. Meist kann man sich detaillierte Informationen über die Arbeitsweise und die Bedienung ausgeben lassen. Oft kann man auch ohne Password in ein allgemeines Fach Mitteilungen schreiben, was der Mailbox von ungewogener Seite auch schon die Bezeichnung Müllbox eingetragen hat.

### In Kürze für die Profis!

Die voreingestellten Parameter sind 8N1, Vollduplex, 300 Baud und ENTER ergibt nur carriage return (CR). Zweimal ESC führt zum Menü. Eine falsche Menüwahl wird durch Drücken der richtigen Taste korrigiert. Mit ENTER wird bestätigt. Das Parametermenü initialisiert die Schnittstelle einschließlich des Empfangspuffers und des Spoolers. Verschiedene Baudraten für Senden und Empfangen werden durch einen Punkt getrennt. Baudraten über 9999 werden nicht angenommen.

### Protokoll

Was wäre ein Terminal-Programm ohne die Möglichkeit, die empfangenen Daten später noch einmal ansehen oder bearbeiten zu können. Zwar ist es bei 300 Baud, also grob gesagt 30 Zeichen/sec. noch gut möglich mitzulesen, aber einmal vom Bildschirm gescrollt, wären die Daten ohne Protokollierung auf immer verloren.
CPC-TERM verfügt über mehrere Möglichkeiten, die Daten zu erhalten. Angewählt werden sie über das Untermenü »Protokoll ein/aus«. Die erste Option ist die direkte Ausgabe auf den Drucker. Diese wird von vornherein eingeschaltet, wenn das Programm beim Start einen betriebsbereiten Drucker vorfindet. Ist kein betriebsbereiter Drucker angeschlossen, kann sie auch nachträglich nicht eingeschaltet werden. Eine entprechende Meldung erscheint im Menü-Fenster. Ein-/ausgeschaltet wird mit »j« bzw. »n«, der bestehende Zustand bleibt erhalten bei ENTER.
Die Druckausgabe wird über einen Drucker-Spooler (Zwischenspeicher) organisiert, der im Verwaltungsprogramm implementiert ist. Deshalb ist es sogar möglich mit Einzelblättern zu arbeiten. Der Spooler faßt 4KB. Initialisiert wird der Spooler nur durch die Anwahl des Parametermenüs.
Das RAM-Protokoll ist die zweite Möglichkeit. Hierfür steht ein 14KB großer Bereich zur Zwischenspeicherung zur Verfügung. Die empfangenen Zeichen (und bei Halb-Duplex auch die gesendeten) werden dort der Reihe nach abgelegt. Tritt ein Überlauf auf (mehr als 14K), so erscheint eine Meldung auf dem Schirm und der Puffer wird vom Anfang her überschrieben, so daß immer die letzten 14K erhalten bleiben. Das RAM-Protokoll ist beim Programmstart immer eingeschaltet. Neu Initialisiert wird das RAM-Protokoll durch erneutes Einschalten.
Die dritte Möglichkeit ist das direkte Schreiben aller Daten auf die Diskette. Hierfür muß selbstredend eine beschreibbare Diskette im Laufwerk liegen. Wurde »j« gedrückt, so wird nach

einem File-Namen gefragt. Nur ENTER gibt jetzt den Disc-Catalog aus, mit »a:« oder »b:« kann das gewünschte Laufwerk eingeschaltet werden. Die Namen dürfen keine Extension enthalten, entsprechend der standardmäßigen Kennzeichnung von ASCII-Files.

Bei der Protokollierung auf Diskette treten allerdings regelmäßig Fehler auf und zwar alle 2048 Zeichen. Ist nämlich der Ausgabepuffer voll, so werden 2KB auf die Diskette geschrieben. Dabei werden die Interrupte so lange gesperrt, daß meistens ein paar Zeichen von der V/24 nicht in den Empfangspuffer übernommen werden. Daraus wird auch klar, daß diese Möglichkeit für den Kassettenbetrieb nicht geeignet ist. Besser dran sind hier diejenigen, die über eine RAM-Floppy verfügen, die die Interrupte nicht sperrt (z.B. die »silicon disc« von dk'tronics). Gedacht ist das direkte Disketten- Protokoll für Situationen, in denen mehr als 14KB Daten gesichert werden sollen. Ansonsten empfiehlt sich eine RAM-Zwischenspeicherung und anschließendes Sichern auf Diskette oder Kassette (s.u.).

Bei Halb-Duplex-Betrieb sollten nicht alle drei Protokoll-Funktionen gleichzeitig eingeschaltet sein. Der jeweilige Zustand kann rechts in der Statuszeile abgelesen werden. Die entsprechenden Kürzel sind »Dr:«, »Ra:«, »Fi:« mit jeweils »0« für aus und »1« für ein.

## Protokoll ausgeben

Hier kann das interne RAM-Protokoll nachträglich auf Drucker, Bildschirm oder als ASCII-File auf Diskette ausgegeben werden. So kann beispielsweise eine Disketten- oder Kassetten-Datei als Protokoll erstellt werden, die frei von den oben genannten Fehlern ist, wobei uns die Übertragungsfehler ja trotzdem erhalten bleiben.

Wurde keine der obigen Möglichkeiten ausgewählt, kann ein ASCII- File von der Diskette eingelesen und angesehen werden. Angehalten wird dabei mit ENTER, abgebrochen mit ESC.

## ASCII-File senden

Schreibt man einen längeren Text mit einem Textverarbeitungsprogramm oder auch dem Basic-Editor und speichert diesen als ASCII-File ab, so kann man dieses über die Funktion »ASCII-File send.« mit CPC-TERM an die Gegenstation senden. Der Vorteil liegt darin, daß vorher alle Möglichkeiten der Textverarbeitung genutzt werden können und vor allem der Gebührenzähler noch nicht läuft. Dieses Verfahren funktioniert auch mit Basic-Programmen, die als ASCII-File abgespeichert wurden (SAVE»NAME«,a). Zu beachten ist lediglich, daß der Datei-Name keine Extension beinhalten darf (also max. 8 Buchstaben), notfalls umbenennen oder Zeile 9050 ändern.

Auch hier ist es möglich sich den Disketten-Catalog ausgeben zu lassen, bevor ein Name eingegeben wird. Ist dabei jedoch noch ein Protokoll-File offen, so wird dies automatisch geschlossen, da es andernfalls durch das CAT-Kommando verlorenginge. Wird die anschließende ja/nein Frage mit ENTER, DEL oder n beantwortet, kehrt das Programm in den Terminal-Modus zurück.

## ENDE

Wichtig ist, das Programm nur über die Wahl von ENDE zu verlassen, da ansonsten z.B. ein noch offenes Protokoll-File verlorenginge. Außerdem wird der Rechner in einen definierten Zustand gebracht, durch ein CLEAR wird z.B. das noch aktive Unterbrechungsereignis (ON BREAK GOSUB 1000) beendet. Andernfalls würde das Programm bei einem anschließenden GOTO xxxx nicht mehr zurückfinden. Das Ende muß noch mit »j« bestätigt werden, andere Eingaben (s.o) führen zum Terminal Modus.

## Default Parameter

In den Programmzeilen ab 8150 wird festgelegt, wie die Parameter beim Programmstart eingestellt sein sollen. Hier kann sich jeder seine Standardeinstellung eintragen. Die Variablen haben weitgehend selbsterklärende Namen. Zusätzlich wird aus den Zeilen ab 8000 klar wo sie eingesetzt werden.

Die Variable fi, die das Protokoll File schaltet, darf hier jedoch nicht einfach auf -1 für »an« gestellt werden, da ja noch kein OPENOUT-Kommando erteilt wurde.

Die Funktionstasten 4-9 (mit Shift) wurden am Programmanfang eingebaut (Zeilen 100-150), so daß hier individuelle Einträge möglich sind. Zur Demonstration wurden die Tasten 7 und 8 belegt.

Kopien können nach regulärem Programmstart und ENDE mit RUN 9999 gemacht werden. Das Programm fängt DISC-Fehler nicht ab. Sollte ein Fehler beim Eintippen eines Dateinamens aufgetreten sein, so kann mit

**a$**=»NAME«:cont

fortgefahren werden. Interessierte Leser können sich hier z.B. die Fehlerroutine aus CPC International 1/86 einbauen. Das Programm läßt noch ca. 3KB für Daten und eventuelle Erweiterungen frei.

## Hinweise für Profis

Das Programm verträgt sich nicht ohne weiteres mit allen Schneiderware-Erweiterungen. Deshalb wird hier die Speicherbelegung angegeben, um Anpassungen zu erleichtern. Mit der Hardware-Uhr dürfte es keine Probleme geben.

## SPEICHERBELEGUNG

Angegeben ist jeweils die Obergrenze (incl. letztem Byte)

---- HIMEM (nicht unter A281h) ----------
I/O-Buffer (Disk/Kass)
---- 9280h ----------
Verwaltungsprogramm V/24----8FFFh
frei (evt. Hardware-Uhr)----7FFFh
Drucker-Buffer----6FFFh----------
Empfangs-Buffer----5FFFh----------
Protokoll-Buffer----27FFh (HIMEM nach RUN)----------
Programm- & Variablen-Bereich---170h----------

## XON-XOFF-PROTOKOLL

Das Verwaltungsprogramm blendet bestimmte empfangene Zeichen aus, weil sie ansonsten den Bildschirm durcheinanderbringen würden. Die Tabelle dieser Zeichen steht von 91B4h bis 91CAh. Dort stehen auch die Zeichen 17 (XON) UND 19 (XOFF). Soll also ein XON/XOFF-Protokoll im Programm implementiert werden, so müssen in den Speicherstellen 91BDh und 91BFh Nullen eingetragen werden, damit XON bzw. XOFF nicht mehr ausgeblendet werden. Selbstverständlich muß dann das BASIC-Programm so ergänzt werden, daß die Zeichen entsprechend ausgewertet werden. (B.Noe)

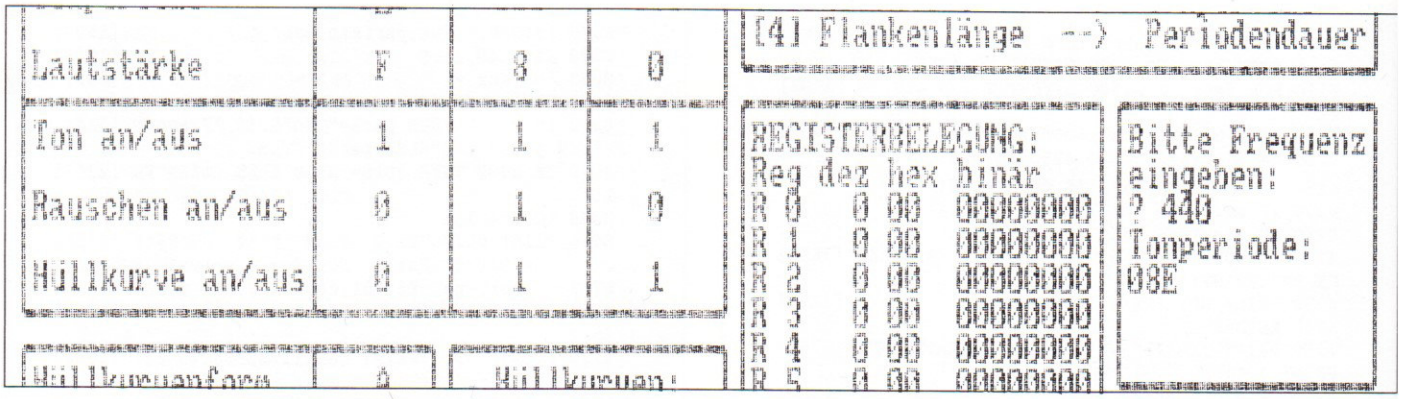
```
10 DATA &01,&0D,&90,&21,&09,&90,&C3,&D1,&B [2714]
C,&09, 945
20 DATA &90,&0D,&90,&21,&90,&C3,&3F,&90,&C [1617]
3,&41, 1140
30 DATA &90,&C3,&7C,&90,&C3,&ED,&90,&C3,&9 [1955]
1,&91, 1668
40 DATA &C3,&4A,&91,&49,&4E,&49,&54,&B4,&4 [2504]
2,&41, 1033
50 DATA &55,&C4,&53,&49,&4F,&49,&4E,&49,&D [2354]
4,&53, 1035
60 DATA &45,&4E,&C4,&52,&45,&43,&D6,&50,&5 [1299]
2,&49, 1010
70 DATA &4E,&D4,&00,&37,&C9,&FE,&02,&C0,&0 [1304]
0,&01, 995
80 DATA &EF,&F9,&FD,&21,&5F,&90,&FD,&7E,&0 [3006]
0,&ED, 1629
90 DATA &79,&FD,&7E,&01,&ED,&79,&FD,&7E,&0 [2577]
2,&ED, 1477
100 DATA &79,&CD,&62,&90,&C9,&36,&76,&B6,& [2825]
00,&01, 1124
110 DATA &EC,&F9,&DD,&7E,&00,&ED,&79,&DD,& [2353]
7E,&01, 1538
120 DATA &ED,&79,&03,&DD,&7E,&02,&ED,&79,& [2149]
DD,&7E, 1415
130 DATA &03,&ED,&79,&C9,&FE,&04,&C0,&01,& [2488]
E1,&F9, 1487
140 DATA &3E,&00,&ED,&79,&ED,&79,&ED,&79,& [3102]
3E,&40, 1262
150 DATA &ED,&79,&21,&EC,&90,&AF,&77,&DD,& [2191]
7E,&00, 1412
160 DATA &FE,&01,&28,&3A,&CB,&AE,&DD,&7E,& [1889]
02,&FE, 1333
170 DATA &01,&28,&35,&CB,&F6,&CB,&FE,&DD,& [2542]
7E,&04, 1351
180 DATA &FE,&01,&28,&30,&CB,&A6,&DD,&7E,& [2586]
06,&FE, 1319
190 DATA &07,&28,&2B,&CB,&86,&CB,&CE,&CB,& [3032]
D6,&CB, 1456
200 DATA &DE,&01,&E1,&F9,&7E,&ED,&79,&3E,& [1762]
37,&ED, 1535
210 DATA &79,&CD,&0B,&91,&00,&CD,&3C,&91,& [2742]
CD,&42, 1163
220 DATA &91,&C9,&CB,&EE,&18,&C4,&CB,&F6,& [2697]
CB,&BE, 1849
230 DATA &18,&C9,&CB,&E6,&18,&CE,&CB,&86,& [2330]
CB,&CE, 1634
240 DATA &CB,&96,&CB,&DE,&18,&D3,&CE,&DD,& [2279]
7E,&00, 1566
250 DATA &F5,&00,&00,&00,&CD,&00,&91,&30,& [1642]
F8,&F1, 1132
260 DATA &01,&E0,&F9,&ED,&79,&C9,&01,&E1,& [3080]
F9,&ED, 1745
270 DATA &78,&E6,&01,&C8,&37,&9F,&C9,&21,& [1540]
00,&60, 1095
280 DATA &22,&1E,&91,&22,&20,&91,&21,&00,& [2560]
70,&22, 599
290 DATA &22,&91,&22,&24,&91,&C9,&01,&60,& [2309]
01,&60, 789
300 DATA &00,&70,&00,&70,&3A,&71,&01,&60,& [1566]
00,&00, 492
310 DATA &00,&00,&00,&81,&01,&92,&00,&2A,& [1038]
91,&2C, 507
320 DATA &91,&00,&80,&49,&91,&00,&21,&33,& [2204]
91,&C3, 915
330 DATA &E3,&BC,&21,&2A,&91,&C3,&E3,&BC,& [1793]
C9,&C9, 1647
340 DATA &E5,&D5,&C5,&F5,&DD,&7E,&00,&CD,& [2719]
59,&91, 1670
350 DATA &F1,&C1,&D1,&E1,&C9,&2A,&24,&91,& [2095]
CD,&79, 1618
360 DATA &91,&ED,&4B,&22,&91,&37,&3F,&ED,& [1487]
42,&CA, 1259
370 DATA &77,&91,&2A,&24,&91,&77,&CD,&79,& [2215]
91,&22, 1111
380 DATA &24,&91,&37,&3F,&C9,&37,&C9,&C5,& [2059]
23,&22, 1022
390 DATA &26,&91,&01,&FF,&7F,&37,&3F,&ED,& [2342]
42,&20, 1019
400 DATA &05,&21,&00,&70,&C1,&C9,&2A,&26,& [1853]
91,&C1, 962
410 DATA &C9,&CD,&9C,&91,&DD,&66,&01,&DD,& [2611]
6E,&00, 1362
420 DATA &77,&C9,&CD,&CC,&91,&DA,&B2,&91,& [2798]
E6,&7F, 1772
430 DATA &21,&B5,&91,&BE,&CA,&B2,&91,&CB,& [1363]
7E,&C0, 1595
440 DATA &23,&C3,&A7,&91,&3E,&00,&C9,&01,& [1889]
02,&03, 811
450 DATA &04,&05,&06,&0E,&0F,&10,&11,&12,& [1291]
13,&14, 134
460 DATA &15,&16,&17,&18,&19,&1A,&1C,&1D,& [2447]
1F,&80, 357
470 DATA &2A,&1E,&91,&ED,&4B,&20,&91,&37,& [2754]
3F,&ED, 1061
480 DATA &42,&CA,&E7,&91,&2A,&1E,&91,&7E,& [3242]
CD,&E9, 1425
490 DATA &91,&22,&1E,&91,&37,&3F,&C9,&37,& [2668]
C9,&C5, 1126
500 DATA &23,&22,&28,&91,&01,&FF,&6F,&37,& [3712]
3F,&ED, 976
510 DATA &42,&20,&05,&21,&00,&60,&C1,&C9,& [2078]
2A,&28, 708
520 DATA &91,&C1,&C9,&F5,&E5,&C5,&D5,&CD,& [2771]
10,&92, 1790
530 DATA &CD,&1C,&92,&D1,&C1,&E1,&F1,&C9,& [2257]
CD,&2E, 1699
540 DATA &BD,&D8,&CD,&2D,&92,&D8,&CD,&31,& [2244]
BD,&C9, 1661
550 DATA &CD,&6A,&92,&E6,&02,&C8,&C5,&01,& [1542]
E0,&F9, 1560
560 DATA &ED,&78,&C1,&CD,&4A,&92,&C9,&2A,& [2644]
22,&91, 1397
570 DATA &ED,&4B,&24,&91,&37,&3F,&ED,&42,& [2270]
CA,&48, 1188
580 DATA &92,&2A,&22,&91,&7E,&CD,&79,&91,& [2514]
22,&22, 1032
590 DATA &91,&37,&3F,&C9,&37,&C9,&2A,&20,& [1876]
91,&CD, 1144
600 DATA &E9,&91,&ED,&4B,&1E,&91,&37,&3F,& [2393]
ED,&42, 1286
610 DATA &CA,&68,&92,&2A,&20,&91,&77,&CD,& [3111]
E9,&91, 1373
620 DATA &22,&20,&91,&37,&3F,&C9,&37,&C9,& [1985]
C5,&01, 984
630 DATA &E1,&F9,&ED,&78,&C1,&C9,&00,&00,& [1859]
00,&00, 1225
640 DATA &00,&00,&00,&00,&00,&00,&00,&00,& [1730]
00,&00, 0
650 DATA &00, 0                                [213]
660 dat=0:sz=0:dz= 10                          [319]
670 FOR adr = &9000 TO  &9280                  [831]
680 READ byte : dat=dat+1                      [956]
690   sz=sz+byte                               [619]
700   POKE adr,byte                            [84]
710   IF dat < 10 AND adr <  &9280 THEN 75     [693]
0
720  READ chksum                               [1222]
730  IF chksum<>sz THEN PRINT "Fehler in z     [2720]
eile :";dz
740   dz=dz + 10 : sz=0:dat=0                  [843]
750 NEXT adr                                   [547]
760 MODE 2:PRINT:PRINT"     MAIL.BIN wird      [2950]
gespeichert"
770 PRINT"     Datentraeger fertigmachen,      [3915]
dann TASTE":CALL &BB06
780 SAVE"mail.bin",b,&9000,&280                [1457]
790 END                                        [110]
```

---

```
10 '********************************          [1285]
20 '***       * CPC-TERM *       ***          [766]
30 '***        1987   B. Noe     ***          [1170]
40 '*** V 24 aus Schneiderware#3 ***          [2468]
50 '***       erforderlich       ***          [1554]
60 '********************************          [1285]
70 '---- Initialisierung ----                 [758]
80 DEFINT a-z                                  [553]
90 a=0:pz=&5FFF                                [1766]
100 KEY 14,"4"                                 [485]
110 KEY 15,"5"                                 [339]
120 KEY 16,"6"                                 [457]
130 KEY 17,"PASSWORD"+CHR$(13)                 [1343]
140 KEY 18,"HELP"+CHR$(13)                     [1133]
150 KEY 19,"9"                                 [592]
```

```
160 KEY DEF 18,0,140,140,140                [1590]
170 KEY DEF 6,0,140,140,140                 [979]
180 KEY DEF 10,0,135,145                    [746]
190 KEY DEF 11,0,136,146                    [1028]
200 KEY DEF 3,0,137,147                     [1081]
210 KEY DEF 20,0,132,142                    [1259]
220 KEY DEF 12,0,133,143                    [1185]
230 KEY DEF 4,0,134,144                     [1195]
240 MODE 2                                  [513]
250 WINDOW#1,1,80,1,23                      [990]
260 WINDOW#2,1,80,25,25                     [1376]
270 WINDOW#3,59,80,1,11                     [1382]
280 WINDOW#4,61,80,1,11:PAPER#4,1:PEN#4,0   [3098]
290 WINDOW#5,1,58,1,23                      [1338]
300 WINDOW#6,59,80,12,23                    [1358]
310 IF HIMEM>&27FF THEN OPENOUT"!d":MEMORY  [7035]
 HIMEM-1:CLOSEOUT:MEMORY &27FF:LOAD"!MAIL.
OBJ":CALL &9000
320 GOSUB 8150                              [842]
330 WINDOW SWAP 1                           [826]
340 PRINT"*  CPC-TERM bereit  *  MENU: [ES  [2800]
C/ESC]  *":PRINT
350 MOVE 0,378:DRAWR 640,0:MOVE 0,20:DRAWR  [2537]
 640,0
360 ON BREAK GOSUB 1000                     [581]
370 KEY DEF 66,0,252                        [877]
400 '--------- Hauptschleifen ---------     [2013]
410 '---- VOLL-Duplex ----                  [1169]
420 a$=INKEY$:IF a$<>""THEN|SEND,ASC(a$)    [2202]
430 |RECV,@a:PRINT CHR$(a);:IF a=0 THEN GO  [1960]
TO 470
440 IF dr THEN |PRINT,a                     [970]
450 IF ra THEN POKE pz,a:pz=pz-1:IF pz<&28  [2441]
00 THEN GOSUB 800
460 IF fi THEN PRINT#9,CHR$(a);             [642]
470 ON hd GOTO 420,510                      [321]
480 '                                       [117]
500 '---- HALB-Duplex ----                  [826]
510 a$=INKEY$:IF a$=""GOTO 550              [1692]
520 |SEND,ASC(a$):PRINT a$;:IF dr THEN |PR  [2709]
INT,ASC(a$)
530 IF ra THEN POKE pz,ASC(a$):pz=pz-1:IF   [3011]
pz<&2800 THEN GOSUB 800
540 IF fi THEN PRINT#9,a$;                  [929]
550 |RECV,@a:PRINT CHR$(a);:IF a=0 THEN GO  [1769]
TO 590
560 IF dr THEN|PRINT,a                      [970]
570 IF ra THEN POKE pz,a:pz=pz-1:IF pz<&28  [2441]
00 THEN GOSUB 800
580 IF fi THEN PRINT#9,CHR$(a);             [642]
590 ON hd GOTO 420,510                      [321]
800 '---- Protokollbuffer voll ----         [1292]
810 PRINT:PRINT"** PROTOKOLLBUFFER UEBERLA  [2862]
UF **"
820 pkb.uebrl=-1                            [1007]
830 pz=&5FFF                                [185]
840 RETURN                                  [555]
1000 '---- H A U P T - M E N U ----         [836]
1010 KEY DEF 66,0,0:',0 --x-- ergaenzen!!!  [2148]
!!
1020 LOCATE#5,1,VPOS(#0)                    [897]
1030 PAPER#3,1:CLS#3:PAPER#3,0              [2724]
1040 WINDOW SWAP 4                          [823]
1050 CLS                                    [91]
1060 KEY DEF 18,0,13,13,13                  [756]
1070 KEY DEF 6,0,13,13,13                   [1229]
1080 PRINT:PRINT"1  Parameter               [1529]
1090 PRINT:PRINT"2  Protok. ein/aus         [2188]
1100 PRINT:PRINT"3  Protok. ausgeben        [2134]
1110 PRINT:PRINT"4  ASCII-File send.        [2513]
1120 PRINT:PRINT"5  zurueck  6 Ende";       [1416]
1130 wahl=VAL("0"+INKEY$)                   [1651]
1140 IF wahl=0 OR wahl>6 THEN 1130          [2207]
1150 IF wahl=5 THEN 1200                    [982]
1160 GOSUB 1300                             [950]
1170 a$=INKEY$:IF a$=""OR VAL("0"+a$)=wahl  [2251]
 GOTO 1170
1180 IF VAL("0"+a$)<>wahl AND a$<>CHR$(13)  [4958]
 THEN GOSUB 1300:wahl=VAL("0"+a$):GOTO 114
0
1190 ON wahl GOSUB 2000,3000,4000,5000,125  [1943]
0,6000
1200 IF PEEK(&B289)=60 THEN CLS#3:WINDOW S  [2828]
WAP 4
1210 KEY DEF 18,0,140,140,140               [1590]
1220 KEY DEF 6,0,140,140,140                [979]
1230 KEY DEF 66,0,252,252,252               [1063]
1240 WHILE INKEY$<>"":WEND:a$=""            [2110]
1250 RETURN                                 [555]
1300 FOR i=1-11*(wahl=6)TO 19:LOCATE i,wah  [4696]
1*2+2*(wahl=6):CALL &BB8A:NEXT:RETURN
2000 '---- Parameter-Menu ----              [1450]
2010 CLS                                    [91]
2020 ON ERROR GOTO 2240                     [1467]
2030 PRINT:INPUT"BAUD S(.E): ",in$          [474]
2040 IF VAL(in$)>1 THEN bd1=INT(VAL(in$))   [2656]
2050 IF LEN(in$)>=LEN(STR$(bd1))THEN bd2=V  [4914]
AL(MID$(in$,LEN(STR$(bd1))+1))ELSE bd2=bd1
2060 IF bd2=0 THEN bd2=bd1                  [1023]
2070 IF bd1>9999 OR bd2>9999 THEN 2010      [2490]
2080 PRINT:INPUT"Datenbits:    ",in         [1884]
2090 IF in>0 THEN IF in<7 OR in>8 THEN 208  [1633]
0 ELSE dat=in
2100 PRINT:INPUT;"Parit. (0/1): ",in$       [883]
2110 IF in$<>""THEN in=VAL(in$):IF in<0 OR  [3821]
 in>1 THEN 2100 ELSE par=-in
2120 IF NOT par THEN PRINT:GOTO 2150        [1255]
2130 PRINT CHR$(17)CHR$(13);:INPUT"(E)ven/  [3572]
(O)dd: ",in$
2140 IF in$<>""THEN in$=UPPER$(in$):IF in$  [5597]
="E"THEN evn=1 ELSE IF in$="O"THEN evn=0 E
LSE GOTO 2130
2150 PRINT:INPUT"Stoppbits:    ",in         [1805]
2160 IF in>0 THEN IF in<1 OR in>2 THEN 215  [3032]
0 ELSE stp=in
2170 PRINT:INPUT"Duplex (V/H): ",in$        [1332]
2180 IF in$<>""THEN in$=UPPER$(in$):IF in$  [5272]
="V"THEN hd=1 ELSE IF in$="H"THEN hd=2 ELS
E GOTO 2170
2190 PRINT:INPUT"ENTER C+L,C,L ",in$        [2122]
2200 IF in$<>""THEN in$=UPPER$(in$):IF in$  [6912]
="C+L"THEN etr$=CHR$(13)+CHR$(10)ELSE IF i
n$="C"THEN etr$=CHR$(13)ELSE IF in$="L"THE
N etr$=CHR$(10)ELSE 2190
2210 ON ERROR GOTO 0                        [1381]
2220 GOSUB 8000                             [824]
2230 RETURN                                 [555]
2240 RESUME NEXT                            [703]
3000 '---- Protokolle ein/aus ----          [946]
3010 CLS                                    [91]
3020 PRINT:PRINT"PROTOKOLLE ein/aus         [2967]
3030 PRINT:PRINT"Drucker   ";               [2323]
3040 GOSUB 9130                             [1046]
3050 GOSUB 9190:IF fehl GOTO 3030 ELSE IF   [5672]
NOT abr THEN dr=ja ELSE ja=dr:GOSUB 9190:I
F fehl GOTO 3030
3060 PRINT:PRINT"RAM-Prokl.";:GOSUB 9130    [1796]
3070 IF NOT abr THEN ra=ja:pz=&5FFF:pkb.ue  [2534]
brl=0
3080 PRINT:PRINT"ASCII-File";:GOSUB 9130    [2387]
3090 IF abr THEN 3110 ELSE IF fi THEN GOSU  [2373]
B 9090
3100 IF ja THEN GOSUB 9000:IF NOT fehl THE  [3706]
N fi=-1:OPENOUT a$ ELSE 3100
3110 GOSUB 8040                             [1040]
3120 RETURN                                 [555]
4000 '---- Protokolle ausgeben ----         [1607]
4010 CLS                                    [91]
4020 PRINT:PRINT"PROTOKOLLE ausgeben        [2693]
4030 IF pz=&5FFF AND NOT pkb.uebrl THEN PR  [6466]
INT:PRINT"RAM leer..":PRINT:PRINT:GOTO 430
0
4040 PRINT:PRINT"Ra drucken";               [1119]
4050 GOSUB 9130                             [1046]
4060 GOSUB 9190:IF fehl GOTO 4040 ELSE dru  [3356]
=ja
4070 PRINT:PRINT"Ra ansehen";               [1503]
4080 GOSUB 9130:ans=ja                      [1342]
4090 PRINT:PRINT"Ra sichern";               [1738]
4100 GOSUB 9130:sic=ja:PRINT                [1831]
4110 IF sic THEN GOSUB 9090:GOSUB 9000:IF   [3870]
NOT fehl THEN OPENOUT a$ ELSE 4110
4120 IF NOT(dru OR ans OR sic)THEN 4280     [2776]
4130 CLS#3                                  [371]
4140 WINDOW SWAP 4                          [823]
4150 PRINT TAB(1),:PRINT:PRINT"  ** PROTOK  [4568]
OLL ** Anhalten: [ENTER] ** Abr.: [ESC] **
```

```
4160 IF NOT pkb.uebrl THEN FOR i=&5FFF TO  [4355]
pz+1 STEP-1:GOSUB 4190:NEXT:GOTO 4250
4170 FOR i=pz TO &2800 STEP-1:GOSUB 4190:N [3308]
EXT
4180 IF i>0 THEN FOR i=&5FFF TO pz+1 STEP- [4436]
1:GOSUB 4190:NEXT:GOTO 4250 ELSE 4250
4190 IF dru THEN PRINT#8,CHR$(PEEK(i));    [2100]
4200 IF ans THEN PRINT CHR$(PEEK(i));      [2345]
4210 IF sic THEN PRINT#9,CHR$(PEEK(i));    [1889]
4220 IF NOT INKEY(66)THEN PRINT"  ** ABBRU [3626]
CH **":PRINT:i=0
4230 WHILE NOT INKEY(18):WEND              [2421]
4240 RETURN                                [555]
4250 PRINT TAB(1)"  ** PROTOKOLL ENDE **": [2761]
PRINT
4260 IF sic THEN CLOSEOUT                  [1783]
4270 RETURN                                [555]
4280 '-- ASCII-File ansehen --             [2442]
4290 LOCATE 1,4:PRINT CHR$(20);            [2361]
4300 PRINT"ASCII-File ansehen":PRINT TAB(1 [2204]
1)
4310 GOSUB 9130:IF NOT ja THEN GOTO 4430   [3062]
4320 PRINT                                 [361]
4330 GOSUB 9000:IF NOT fehl THEN OPENIN a$ [3351]
 ELSE 4330
4340 CLS#3                                 [371]
4350 WINDOW SWAP 4                         [823]
4360 PRINT TAB(1),:PRINT:PRINT"  ** ASCII- [5711]
File >"a$"< ansehen ** Anhalten: [ENTER] *
* Abr.: [ESC] **
4370 WHILE NOT EOF                         [1840]
4380 LINE INPUT#9,a$:PRINT a$              [1776]
4390 IF NOT INKEY(66)THEN PRINT"  ** ABBRU [4322]
CH **":PRINT:GOTO 4420
4400 WHILE NOT INKEY(18):WEND              [2421]
4410 WEND                                  [390]
4420 PRINT TAB(1)"  ** ASCII-File ansehen  [2433]
ENDE **":PRINT
4430 CLOSEIN                               [752]
4440 RETURN                                [555]
5000 '---- ASCII-File senden ----          [1868]
5010 CLS                                   [91]
5020 PRINT:PRINT"ASCII-File senden"        [2146]
5030 PRINT:PRINT"CAT > ENTER":PRINT        [2003]
5040 GOSUB 9000                            [1040]
5050 IF fehl GOTO 5040                     [919]
5060 KEY DEF 66,0,252                      [877]
5070 OPENIN a$                             [715]
5080 PRINT:PRINT"... senden";:GOSUB 9130:I [3805]
F NOT ja THEN CLOSEIN:CLS#3:GOTO 5230
5090 CLS#3                                 [371]
5100 WINDOW SWAP 4                         [823]
5110 PRINT:PRINT"  ** ASCII-File >"a$"< se [1383]
nden ** Abr.:[ESC] **
5120 WHILE NOT EOF                         [1840]
5130 LINE INPUT#9,a$                       [1529]
5140 FOR i=1 TO LEN(a$)                    [791]
5150 |SEND,ASC(MID$(a$,i))                 [2121]
5160 IF INKEY$=CHR$(252)THEN GOSUB 5240:PR [4264]
INT"  ** ABBRUCH **":GOTO 5210
5170 NEXT                                  [350]
5180 GOSUB 5240                            [899]
5190 |RECV,@a:IF a THEN PRINT CHR$(a);:GOT [2459]
O 5190
5200 WEND                                  [390]
5210 CLOSEIN                               [752]
5220 PRINT:PRINT"  ** ASCII-File senden EN [3079]
DE **":PRINT
5230 RETURN                                [555]
5240 FOR i=1 TO LEN(etr$):|SEND,ASC(MID$(e [3179]
tr$,i)):NEXT
5250 FOR i=1 TO 35:NEXT:' Pause nach 'CR'  [2310]
 ca.50ms
5260 RETURN                                [555]
6000 '---- E N D E ----                    [794]
6010 PRINT"...wirklich";:GOSUB 9130        [2020]
6020 IF NOT ja THEN RETURN                 [1554]
6030 GOSUB 9090:WINDOW SWAP 3:PAPER 0:PEN  [4859]
1:KEY DEF 66,0,252:CLEAR
6040 END                                   [110]
8000 '---- V/24 einstellen und Anzeigen -- [2426]
--
8010 |BAUD,125000/bd1,125000/bd2           [2071]
8020 |SIOINIT,dat,par,stp,evn              [1809]
8030 KEY 140,etr$                          [1249]
8040 '-- anz -                             [360]
8050 WINDOW SWAP 2                         [833]
8060 IF par=0 THEN par$=" No"ELSE IF evn=0 [6134]
 THEN par$="ODD"ELSE par$="EVEN"
8070 IF hd=2 THEN dpl$="HALB"ELSE dpl$="VO [2263]
LL"
8080 LOCATE 1,1                            [611]
8090 PRINT USING"Dr:#  Ra:#  Fi:#  Bd:#### [8238]
     = Dat:#  Par:&  Stp:# =   &-Dupl  E
NTER: ";-dr,-ra,-fi,bd1,dat,par$,stp,dpl$
8100 IF bd1<>bd2 THEN LOCATE 26,1:PRINT bd [3629]
2;
8110 LOCATE 79,1:PRINT CHR$(1)LEFT$((etr$) [1598]
,1);
8120 IF LEN(etr$)>1 THEN PRINT CHR$(1)MID$ [4328]
(etr$,2);ELSE PRINT" ";
8130 WINDOW SWAP 2                         [833]
8140 RETURN                                [555]
8150 '---- Default-Parameter ----          [1883]
8160 bd1=300:bd2=300                       [937]
8170 dat=8:par=0:stp=1:evn=1               [898]
8180 hd=1:' 1=Vd 2=Hd                      [723]
8190 etr$=CHR$(13)                         [1118]
8200 IF(INP(&F500)AND 64)=0 THEN dr=-1     [2143]
8210 ra=-1                                 [381]
8220 GOTO 8000                             [312]
9000 '---- Disc-Name & a:/b:-Umschaltung - [1957]
---
9010 INPUT"FILE-Name: ",a$                 [961]
9020 fehl=-1                               [112]
9030 IF INSTR(a$,":")=2 AND LEN(a$)=2 THEN [6236]
 IF UPPER$(LEFT$(a$,1))="B" THEN |B:GOTO 9
080 ELSE |A:GOTO 9080
9040 IF a$="" AND fi THEN GOSUB 9090:GOSUB [2599]
 8040
9050 IF a$=""THEN PRINT CHR$(11);:WINDOW S [7704]
WAP 5:CAT:WINDOW SWAP 5 ELSE IF LEN(a$)<9+
INSTR(a$,":")THEN fehl=0 ELSE PRINT"max. 8
 Buchstaben
9060 IF a$=""AND VPOS(#5)=23 THEN CLS#6    [2457]
9070 LOCATE#4,1,VPOS(#5)-(VPOS(#5)=23)     [1209]
9080 RETURN                                [555]
9090 '---- Fi-File schliessen ----          [1764]
9100 IF fi THEN PRINT#5:PRINT#6:PRINT#5,"  [8626]
 ** Protokoll-File geschlossen **":PRINT#6
:PRINT#5:PRINT#6:CLOSEOUT:fi=0
9110 LOCATE#4,1,VPOS(#5)-(VPOS(#5)=23)     [1209]
9120 RETURN                                [555]
9130 '---- ja/nein (/abr) ----             [956]
9140 PRINT" ? (j/n)";                      [963]
9150 abr=0:j$=LOWER$(INKEY$):IF j$=""THEN  [2278]
9150
9160 IF j$="j"THEN ja=-1 ELSE ja=0:IF j$=" [3892]
n"THEN ELSE IF j$=CHR$(127)OR j$=CHR$(13)T
HEN abr=-1 ELSE PRINT CHR$(7);:GOTO 9150
9170 PRINT STRING$(9,8);:IF abr THEN PRINT [4348]
"       "ELSE IF ja THEN PRINT"   ja  "ELS
E PRINT"  nein "
9180 RETURN                                [555]
9190 '---- Drucker bereit? ----            [472]
9200 IF ja AND(INP(&F500)AND 64)THEN PRINT [4861]
 CHR$(13)CHR$(11)CHR$(24)"Drucker einschal
ten"CHR$(24):fehl=-1 ELSE fehl=0
9210 RETURN                                [555]
9999 SAVE"cpc-term":SAVE"!MAIL.OBJ",b,&900 [3589]
0,&280:STOP
```




# Sound relativ und absolut

**Keine Angst, es handelt sich hier nicht um die x-te »Einführung in die Soundprogrammierung für Anfänger«. Wir gehen davon aus, daß Sie sich schon mit der Programmierung von Klängen beschäftigt haben; insbesondere sollte Ihnen der SOUND-Befehl des Schneider-Basic vertraut sein. Im Folgenden wollen wir uns hauptsächlich mit jenen Details der Soundprogrammierung befassen, die im CPC-Handbuch nur versteckt angedeutet werden; teilweise sind sie dort auch schlicht falsch dargestellt oder werden gar nicht erst erwähnt. Das betrifft vor allem die Hüllkurven-Kommandos ENV und ENT.**

**ENT**

Als erstes wollen wir ENT, den Befehl zur Definition einer Tonhüllkurve, unter die Lupe nehmen. Das Format ist bekannt:

ENT <Nummer>,<maximal 5 Veränderungsfolgen>

Der Parameter <Nummer> kann zwischen 1 und 15 liegen, es sind also bis zu 15 verschiedene Tonhüllkurven programmierbar. Wenn Sie keine< Veränderungsfolge> angeben (z.B.: ENT 3), wird die betreffende Tonhüllkurve gelöscht. Das Standardformat einer <Veränderungsfolge> sieht so aus:

<Schrittzahl>,<Schrittgröße>,
<Pausenzeit>
Dabei gilt:
<Schrittzahl>: Wertebereich 0...239
<Schrittgröße>: Wertebereich−128...+127
<Pausenzeit>: Wertebereich 0...255
Die Abarbeitung einer Tonhüllkurve geschieht wie folgt: Zunächst wird die aktuelle Tonperiode um den Wert <Schrittgröße> vergrößert (bzw.vermindert, wenn <Schrittgröße> negativ ist). Der so entstandene Ton wird so viele Hundertstelsekundenlang gehalten, wie durch <Pausenzeit> angegeben (eine <Pausenzeit> von 0 wird als 256 interpretiert). Dieser Vorgang wird so oft wiederholt, wie durch <Schrittzahl> definiert. Anschließend wird, falls vorhanden, die nächste <Veränderungsfolge> abgearbeitet usw. Um uns diesen Vorgang ganz klarzumachen, wollen wir ihn einmal durch ein Basic-Programm simulieren.

Allgemein sieht der Aufruf einer Tonhüllkurve so aus:

ENT nr, sz(1), sg(1), pz(1),..., sz(n), sg(n), pz(n)
SOUND kanal, grundperiode, dauer, lautstärke, 0, nr

(n bezeichnet hier die Anzahl der<Veränderungsfolgen> und liegt zwischen 1 und 5.) Das gleiche wird durch das folgende Programm bewirkt:

```
10 periode = grundperiode
20 FOR folge = 1 TO n
30 FOR schritt = 1 TO sz(folge)
40 periode = periode + sg(folge)
50 SOUND kanal, periode,pz(folge),
lautstärke
60 NEXT schritt
70 NEXT folge
```

Beachten Sie dabei, daß die erste Veränderung der Tonperiode schon zu Beginn des erzeugten Tons stattfindet; insofern ist die Grafik im Sound-Kapitel des Handbuchs nicht ganz korrekt. Dazu ein Beispiel:

```
ENT 1, 4,-10,25
SOUND 1,150,100,15,0,1
```

bewirkt das gleiche wie die Befehlsfolge:

```
SOUND 1, 140, 25, 15
SOUND 1, 130, 25, 15
SOUND 1, 120, 25, 15
SOUND 1, 110, 25, 15
```

Sie hören vier aufsteigende Töne von je einer Viertelsekunde Länge. Die Dauer der Tonhüllkurve hat übrigens keinen Einfluß auf die Gesamtdauer des Tons. Geben Sie z.B.ein:

```
SOUND 1, 150, 50, 15, 0, 1
```

so hören Sie nur die ersten zwei Töne.

```
SOUND 1, 150, 200, 15, 0, 1
```

bewirkt wieder die Ausgabe von vier Tönen, wobei der letzte Ton diesmal jedoch eine Sekunde länger gehalten wird. Wenn Sie beim ENT-Kommando eine negative <Nummer> angeben, wird die Tonhüllkurve wiederholt – natürlich nur so lange, wie durch den dritten Parameter des SOUND-Befehls definiert. Beispiel:

```
ENT -1, 4,-10, 25
SOUND 1, 150, 300, 15, 0, 1
```

erzeugt zwölf verschiedene Töne. Die Wiederholungsfunktion eignet sich vorzüglich zur Programmierung von Soundeffekten:

```
ENT -1, 50,-1, 2, 50, 1, 2
SOUND 1, 200, 2000, 15, 0, 1
```

ergibt einen sirenenartigen, an-und abschwellenden Klang. Ein anderes Anwendungsgebiet ist die Erzeugung eines Vibratos, das den doch recht monotonen Rechteckklang des Soundchips etwas interessanter macht:

```
ENT -1, 1, 1, 5, 1,-1, 5
```

Falls Sie mit tiefen Tönen arbeiten, sollten Sie den Betrag von <Schrittgröße> entsprechend größer wählen.

Bisher haben wir uns mit der relativen Änderung der Tonperiode beschäftigt. Jedoch gibt es, wie anscheinend kaum jemand weiß, auch die Möglichkeit, den Ton absolut zu ändern. Hier sieht das Format der<Veränderungsfolge> so aus:

=<Tonperiode>,<Pausenzeit> mit:
<Tonperiode>: Wertebereich 0...4095
<Pausenzeit>: Wertebereich 0...255

In diesem Format hat eine <Veränderungsfolge> also nur zwei Parameter; zur Unterscheidung von der Standardform muß vor dem ersten Parameter ein Gleichheitszeichen stehen. <Pausenzeit> hat dabei die gleiche Funktion wie im Normalformat, bezeichnet also die Pausenlänge in Hundertstelsekunden. <Tonperiode> steht für die absolute Tonhöhe, wie Sie es ja vom SOUND-Kommando her kennen. Beispiel:

```
ENT -1, =239,20, =213,20, =190,20,
=179,20
SOUND 1, 0, 30000, 15, 0, 1
```

spielt Ihnen fünf Minuten lang die ersten vier Noten der C-Dur-Tonleiter vor. (Wenn Ihnen das Gedudel auf die Nerven zu gehen beginnt, drücken Sie einmal auf die DEL-oder CLR-Taste; das erzeugt den bekannten CHR$(7)-Piepser, und der bewirkt nichts anderes als der Befehl SOUND 135, 90, 20, 11.)
Selbstverständlich lassen sich beide Formate – relative und absolute Tonhöhenänderung – beliebig mischen, wobei die Gesamtzahl der < Veränderungsfolgen > natürlich Fünf nicht überschreiten darf.

**ENV**

Wenden wir uns nun der Lautstärkenhüllkurve zu. Der Befehl ENV ist formal mit dem ENT-Kommando fast identisch:

ENV < Nummer >, < maximal 5 Veränderungsfolgen >

Auch hier können bis zu 15 verschiedene Hüllkurven programmiert werden. Das Standardformat einer < Veränderungsfolge > sieht ebenfalls genauso aus:

< Schrittzahl >, < Schrittgröße >, < Pausenzeit >
mit:
< Schrittzahl >: Wertebereich 0...127
< Schrittgröße >: Wertebereich -128... +127
< Pausenzeit >: Wertebereich 0...255
Der einzige Unterschied zur Tonhüllkurve besteht darin, daß < Schrittzahl > den Wert 127 nicht überschreiten darf. < Schrittgröße > bezeichnet hier natürlich nicht die Änderung der Tonhöhe, sondern den Betrag, um den sich die Lautstärke verändert. Es verblüfft vielleicht zunächst, daß < Schrittgröße > Werte zwischen -128 und +127 annehmen darf, da doch die Lautstärkeskala nur von 0 bis 15 reicht. Die Erklärung ist einfach: Wenn die aktuelle Lautstärke aus dem zulässigen Bereich herauswandert, wird sie mit 15 verUNDet und so wieder auf einen korrekten Wert gebracht. Wir wollen auch die Lautstärkenhüllkurve durch ein Basic-Programm simulieren. Das allgemeine Format für ihren Aufruf:

ENV nr, sz(1), sg(1), pz(1),..., sz(n), sg(n), pz(n)
SOUND kanal, periode, dauer, grundlautstärke, nr
Und als Basic-Programm:

```
10 lautstärke = grundlautstärke
20 FOR folge = 1 TO n
30 FOR schritt = 1 TO sz(folge)
40 lautstärke = (lautstärke + sg(folge))
AND 15
50 SOUND kanal, periode, pz(folge),
lautstärke
60 NEXT schritt
70 NEXT folge
```

Betrachten wir jetzt ein einfaches Beispiel:

```
ENV 1, 15, 1, 10
SOUND 1, 250, 150, 0, 1
```

Die Lautstärke wird, wie zu erwarten war, in 15 Schritten von 0 auf den Maximalwert hochgeregelt. Was passiert nun aber, wenn wir im SOUND-Kommando eine andere Grundlautstärke definieren? Geben Sie ein:

```
SOUND 1, 250, 150, 8, 1
```

Ergebnis: Die Lautstärke steigt zunächst von 9 auf 15, wird dann aber abrupt auf 0 gesetzt (denn 16 AND 15 ergibt 0) und klettert anschließend weiter bis zum Endwert 7. Um Verwirrungen zu vermeiden, empfiehlt es sich daher zumeist, die Grundlautstärke im SOUND-Kommando prinzipiell auf 0 zu setzen und jegliche Lautstärke nur über die Hüllkurve zuzugeben.

Auch die Lautstärkenhüllkurve hat im Normalfall keinen Einfluß auf die Dauer des Tons; diese wird durch den dritten Parameter des SOUND-Kommandos bestimmt. Falls die Tondauer kürzer ist als die Länge der Hüllkurve, wird die Hüllkurve nur zum Teil abgearbeitet:

```
SOUND 1, 250, 80, 0, 1
```

regelt die Lautstärke bis auf 8 hinauf und bricht dann ab. Ist die Tondauer dagegen länger als die Hüllkurve, so wird der Ton mit dem zuletzt erreichten Lautstärkepegel fortgesetzt:

```
SOUND 1, 250, 400, 0, 1
```

Soweit also alles wie beim ENT-Kommando. Jedoch gibt es die Möglichkeit, den dritten Parameter des SOUND-Befehls auf 0 zu setzen:

```
SOUND 1, 250, 0, 0, 1
```

Jetzt wird die Tondauer allein durch die Länge der Hüllkurve bestimmt. Übrigens ist auch eine gelöschte Lautstärkenhüllkurve noch definiert; das Betriebssystem rechnet dann mit folgenden Default-Werten: < Schrittzahl > = 1, < Schrittgröße > = 0, < Pausenzeit > = 200. Das können Sie ganz einfach überprüfen; geben Sie einmal ein:

```
ENV 1
SOUND 1, 250, 0, 15, 1
```

Der Ton erklingt genau zwei Sekunden lang. Auch für die Lautstärkenhüllkurve existiert eine Wiederholfunktion.

**Hardwarehüllkurven des PSG AY-3-8912**

| Nr | Kurvenform |
|---|---|
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |

*Abb. 2*

Dazu müssen wir im SOUND-Befehl für die Dauer einen negativen Wert einsetzen:

```
ENV 1, 16,-1,2
SOUND 1, 250,-10, 0, 1
```

Die Hüllkurve wiederholt sich genau zehnmal. Diese Funktion ermöglicht es uns, den CPC auf einfache Weise in einen Drum-Computer zu verwandeln:

```
ENV 1, 16,-1, 2, 8, -4, 4, 16, -1, 2, 4, -4, 8
SOUND 1, 0, -30000, 0, 1, 0, 20
```

ergibt den bekannten DUM-DA-DA-DUM-DAH-Rhythmus. Zur Krönung des Ganzen können wir noch eine Tonhüllkurve definieren:

```
ENT -1, =1000, 32, =500, 32
```

Drücken Sie jetzt DEL oder CLR, um den Ton abzubrechen, und geben Sie dann ein:

```
SOUND 1, 0, -30000, 0, 1, 1, 20
```

Der Takt wird jetzt 30000mal wiederholt, das heißt, das Schlagzeug läuft genau 10 Stunden und 40 Minuten ohne Pause. Das dürfte wohl ausreichen, um auch Ihre längsten Kompositionen mit einer Rhythmusbegleitung zu versehen. Soviel zur relativen Lautstärkenänderung. Natürlich gibt es auch beim ENV-Befehl die Möglichkeit, die Lautstärke absolut zu definieren. Dazu müssen wir < Schrittzahl > auf den (an sich ja sinnlosen) Wert 0 setzen; < Schrittgröße > steht dann für den absoluten Lautstärkewert:

```
ENV 1, 0, 8, 50, 0, 3, 50, 0,15,50
SOUND 1, 250, 150, 0, 1
```

Der Ton erklingt zunächst mit mittlerer Lautstärke (8), wird dann auf 3 gesetzt und schließlich auf den Maximalpegel hochgeschaltet. Selbstverständlich können Sie auch beim ENV-Befehl relative und absolute < Veränderungsfolgen > beliebig mischen. Damit hätten wir alle Aspekte des Standardformats der < Veränderungsfolge > abgehandelt. Zusätzlich gibt es noch ein Format mit nur zwei Parametern; wie beim ENT-Befehl muß

| | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|
| R0 | Tonperiode Kan. A | | | Lo-Byte | | | | |
| R1 | | | | | Hi-Nibble | | | |
| R2 | Tonperiode Kan. B | | | Lo-Byte | | | | |
| R3 | | | | | Hi-Nibble | | | |
| R4 | Tonperiode Kan. C | | | Lo-Byte | | | | |
| R5 | | | | | Hi-Nibble | | | |
| R6 | Rauschperiode | | | | 5 Bit | | | |
| R7 | Freischaltreg. | | Rauschen 0=an | | Ton 0=an | | | |
| | | | C | B | A | C | B | A |
| R8 | Lautstärke Kan. A | | | Hüllk | | Lautst. 4 Bit | | |
| R9 | Lautstärke Kan. B | | | Hüllk | | Lautst. 4 Bit | | |
| R10 | Lautstärke Kan. C | | | Hüllk | | Lautst. 4 Bit | | |
| R11 | Hüllkurvenperiode | | | Lo-Byte | | | | |
| R12 | | | | Hi-Byte | | | | |
| R13 | Hüllkurvenform | | | | 4 Bit | | | |

*Abb.2*

vor dem ersten Parameter zur Unterscheidung ein Gleichheitszeichen gesetzt werden:

= <Hüllkurvenform>, <Veränderungsperiode>
Dabei gilt:
<Hüllkurvenform>: Wertebereich 0...15
<Veränderungsperiode>: Wertebereich 0...65535

Zur Erklärung dieses Formats müssen wir etwas weiter ausholen. Bisher haben wir uns ausschließlich mit softwaremäßig erzeugten Hüllkurven befaßt. Das Betriebssystem des CPC arbeitet die Hüllkurven per Interrupt ab und erzeugt Lautstärkeveränderungen durch direkte Manipulation der Lautstärkeregister des Soundchips. Daneben verfügt der Soundchip aber noch über einen eigenen Hüllkurvengenerator, der normalerweise nicht benutzt wird. Die Zwei-Parameter-Form der <Verände- rungsfolge> erlaubt uns nun den direkten Zugriff auf die Hardwarehüllkurven des PSG (PSG = Programmable Sound Generator, also der Soundchip)

Der Parameter <Hüllkurvenform> bestimmt die gewünschte Hardwarehüllkurve. Diese sind von 0 bis 15 durchnumeriert; allerdings sind die Hüllkurven 0 bis 3 mit Nummer 9 sowie 4 bis 7 mit Nummer 15 identisch, so daß uns de facto nur acht verschiedene Hüllkurven zur Verfügung stehen. Mit Worten lassen sich diese nur unzulänglich beschreiben, wir haben sie daher in einer Tabelle abgebildet (Abb. 1).

Wie Sie sehen, setzen sich die Hardwarehüllkurven des PSG aus verschiedenen Kombinationen von steigenden und fallenden Sägezahnflanken zusammen. Hüllkurve 8 beispielsweise erzeugt einen Ton, der von der Maximallautstärke auf 0 absinkt, dann abrupt wieder auf den Maximalpegel gesetzt wird, wieder leiser wird usw. Hüllkurve 13 dagegen steigt von 0 auf den Maximal- wert und bleibt für den Rest des Tons auf der höchsten Lautstärke. Beachten Sie, daß Lautstärkeveränderungen per Hardwarehüllkurve immer zwischen den beiden Extremwerten der Lautstärkeskala ablaufen (also von 0 nach 15 oder von 15 nach 0); auch die im SOUND-Befehl angegebene Grundlautstärke hat hier keinerlei Einfluß auf den Tonverlauf.

Denken Sie außerdem daran, daß der PSG zwar drei Tongeneratoren besitzt, aber nur einen Hüllkurvengenerator. Wenn Sie also beispielsweise auf Kanal A einen Ton mit der Hardwarehüllkurve 8 ausgeben, können Sie nicht gleichzeitig auf Kanal B einen Ton mit Hüllkurve 14 spielen.

Wir kommen jetzt zum zweiten Parameter. Die <Periodendauer> bestimmt die Flankenlänge der Hardwarehüllkurve. Leider hält sich der PSG nicht an den üblichen Hundertstelsekunden-Takt der CPC-Soundbefehle; eine Einheit steht hier für 256 Mikrosekunden (1 Mikrosekunde = 1 Millionstelsekunde). Angenommen, wir wollen die Hüllkurve 10 mit einer Flankenlänge von einer Sekunde aufrufen. 1 dividiert durch 0,000256 ergibt rund 3906. Geben wir also ein:

```
ENV 1, =10,3906
SOUND 1, 250, 400, 0, 1
```

Selbstverständlich sind auch hier bis zu fünf <Veränderungsfolgen> möglich. Ein Problem besteht allerdings darin, daß wir im Zwei-Parameter-Format keine Pausenzeit angeben können; diese müssen wir daher durch eine weitere <Veränderungsfolge> im Standardformat definieren, wobei die <Schrittgröße> hier natürlich 0 betragen muß (andernfalls würde das Betriebssystem die Hardwarehüllkurve sofort abschalten). Beispiel:

```
ENV 1, =8,1953, 1,0,150, =12,1953
SOUND 1, 250, 0, 0, 1
```

Der Ton erklingt eineinhalb Sekunden lang mit Hüllkurve 8, anschließend hören Sie für 256 Hundertstelsekunden die Hüllkurve 12 (wenn eine ENV-Anweisung mit einer <Verän- derungsfolge> im Zwei-Parameter-Format abschließt, nimmt das Betriebssystem eine Default-Pausenzeit von 256 an).

**PSG**

In der Beschreibung des zweiten Formats der ENV-<Veränderungsfolge> sind wir bereits Ansatzweise auf die direkte Programmierung der Hardware eingegangen. Wir wollen das Thema jetzt vertiefen. Der PSG AY-3-8912 verfügt über fünfzehn dem Benutzer zugängliche Datenregister, die wir als R0 bis R14 bezeichnen wollen. R14 fungiert allerdings als Datenspeicher für den In/Out-Port des Soundchips und hat keinen Einfluß auf die Klangerzeugung (im CPC wird der I/O-Port für die Abfrage von Tastatur und Joystick eingesetzt). Hier die Funktion der restlichen vierzehn Register:

R0/R1: Der Inhalt dieser beiden Register bestimmt die Periode des Tons von Kanal A. Die Tonperiode ist ein 12-Bit-Wert zwischen 0 und 4095 und entspricht dem zweiten Parameter des SOUND-Kommandos; dabei steht in R0 das Low-Byte, in R1 dagegen die obersten vier Bit des Wertes. Es gilt also:

R0 = periode AND 255
R1 = INT (periode / 256)

Es sei noch darauf hingewiesen, daß die Notentabelle im Anhang des CPC-Handbuchs falsch ist. Anscheinend war ursprünglich geplant, den Soundchip mit einer Taktfrequenz von 2 MHz zu versehen; tatsächlich ist der PSG aber nur mit 1 MHz getaktet. Die Formel zur Berechnung der Tonperiode muß daher lauten: periode = ROUND (62500 / frequenz)

R2/R3: Funktion wie R0/R1, aber für Kanal B.

R4/R5: Funktion wie R0/R1, aber für Kanal C.

R6: Dieses Register legt die Rauschperiode fest. Die Rauschperiode kann Werte zwischen 0 und 31 annehmen und entspricht dem siebten Parameter des SOUND-Kommandos. Dabei gilt: je höher die Rauschperiode, desto tiefer die

Durchschnittsfrequenz des erzeugten Rauschens.

R7: Dies ist das Freischaltregister. Die einzelnen Bit kontrollienen:

Bit 0: Ton Kanal A; 1=aus, 0=an
Bit 1: Ton Kanal B; 1=aus, 0=an
Bit 2: Ton Kanal C; 1=aus, 0=an
Bit 3: Rauschen Kanal A; 1=aus, 0=an
Bit 4: Rauschen Kanal B; 1=aus, 0=an
Bit 5: Rauschen Kanal C; 1=aus, 0=an
Bit 6: Datenrichtung des I/O-Ports: 1=out, 0=in

Um also einen Ton auszugeben, müssen Sie zuerst das entsprechende Bit auf 0 setzen. Achten Sie darauf, daß Bit 6 immer 0 ist; andernfalls würde der CPC auf keinen Tastendruck mehr reagieren, und Ihnen bliebe nichts anderes übrig, als den Computer auszuschalten.

R8: Die unteren vier Bit dieses Registers bestimmen die Lautstärke von Kanal A (Wertebereich 0 bis 15, entspricht dem vierten Parameter des SOUND-Befehls). Das fünfte Bit hat eine Sonderfunktion: Falls es gesetzt ist, werden die unteren vier Bit ignoriert und jegliche Lautstärke nur über die Hüllkurve geregelt (siehe R11 bis R13).

R9: Funktion wie R8, aber für Kanal B.

R10: Funktion wie R8, aber für Kanal C.

R11/R12: Durch diese beiden Register wird die Periodendauer der Hüllkurve festgelegt; dabei steht in R11 das Low-Byte und in R12 das High-Byte:

R11 = UNT (periodendauer) AND 255

R12 = INT (periodendauer / 256)
Die Periodendauer läßt sich nach folgender Formel berechnen:
periodendauer = ROUND (sekunden * 3906.25)

R13: Dieses Register bestimmt die Hüllkurvenform; dabei sind Werte zwischen 0 und 15 möglich (siehe Abb. 1).

Zur schnellen Referenz haben wir die Registerbelegung noch einmal in einer Tabelle zusammengefaßt (Abb. 2).

Jetzt stellt sich natürlich die Frage: Wie bekommen wir unsere Daten in die PSG-Register hinein? Von Basic aus geht es schon mal nicht; wir müssen uns also mit Maschinensprache behelfen. Leider ist der PSG nicht direkt an den Prozessor angeschlossen, so daß wir ihn auf dem Umweg über die 8255-PIO programmieren müssen, was allerdings ein ziemlich kompliziertes Datengeschiebe notwendig macht. Aber zum Glück gibt es im ROM die Routine MC SOUND REGISTER, die über den Vektor &BD34 aufgerufen wird; wir müssen ihr lediglich im Akku die Registernummer und in C die zu schreibenden Daten übergeben. Dazu genügt das folgende kleine Assembler-Programm:

```
LD A,(IX+2) ; Registernummer in A
LD C,E ; Datenbyte in C
CALL &BD34 ; MC SOUND REGISTER
RET ; fertig
```

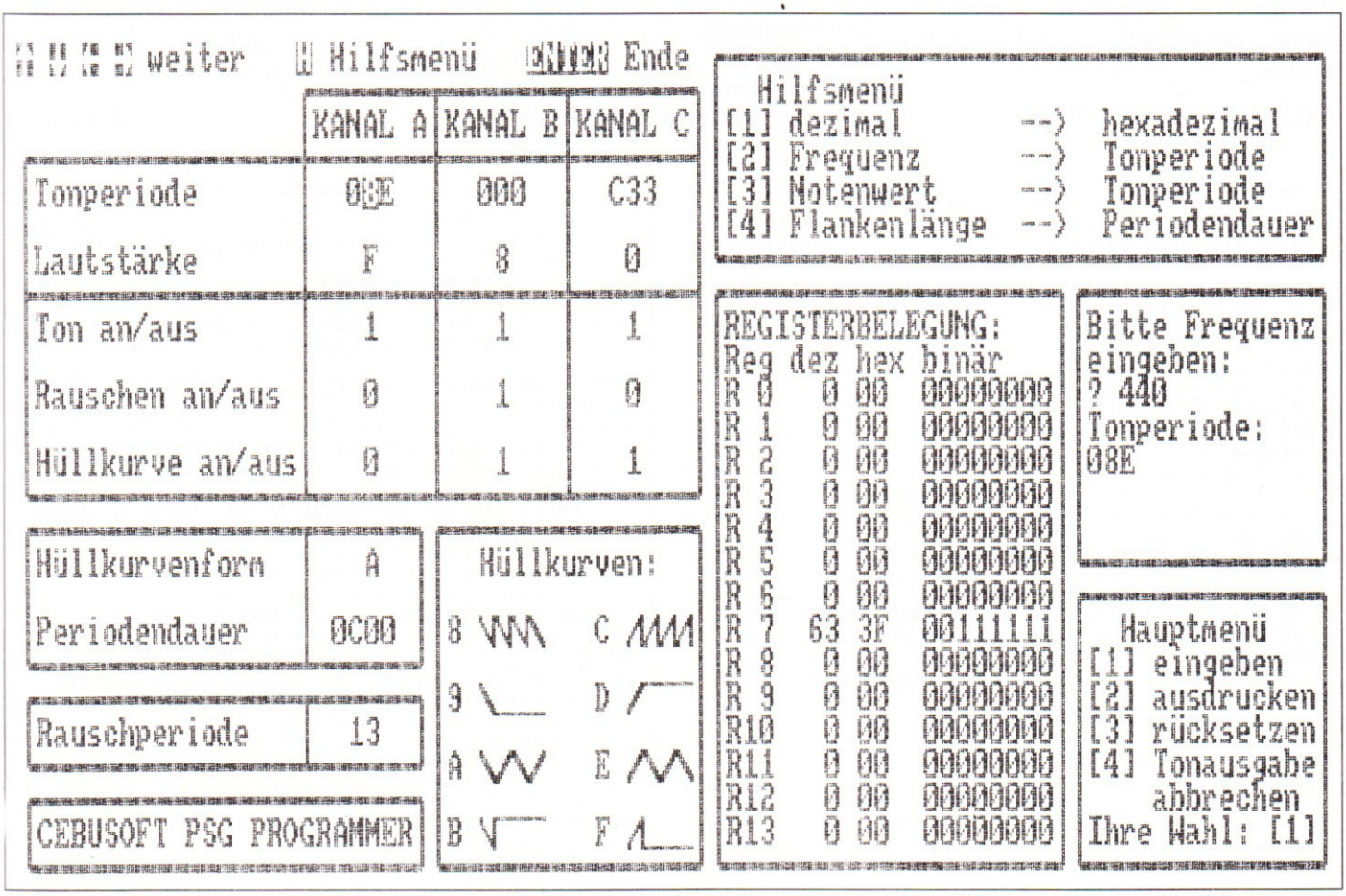
Und hier das gleiche als Basiclader:

```
10 DATA DD, 7E, 02, 4B, CD, 34, BD, C9
20 INPUT »Startadresse«; adresse
30 IF adresse < HIMEM THEN MEMORY
H IMEM + 1
40 FOR i = adresse TO adresse + 7
50 READ byte$
60 POKE i,VAL(»&« + byte$)
70 NEXT i
```

Der Aufruf erfolgt mit:

CALL <Adresse>,<Registernummer>, <Datenwert>

(Falls es Sie interessiert: Unser CALL entspricht genau dem SOUND-Befehl des MSX-Basic.) Zur Verdeutlichung ein kleines Beispiel. Wir wollen auf Kanal B den Kammerton a' mit der Hüllkurve Nummer 14 ausgeben; die Flankenlänge der Hüllkurve soll eine Viertelsekunde betragen. Die international ge-



normte Frequenz des Kammertons beträgt 440 Hz; nach unserer Formel periode = ROUND (62500 / frequenz) ergibt sich eine Tonperiode von 142. Wir müssen also R2 mit 142 und R3 mit 0 laden:

```
CALL adresse,2,142
CALL adresse,3,0
```

Jetzt ist R13 mit der Hüllkurvennummer zu laden:

```
CALL adresse,13,14
```

Die Flankenlänge soll 0,25 Sekunden betragen; die Formel zur Berechnung der Periodendauer lautet:

periodendauer = ROUND(sekunden * 3906.25)

Es ergibt sich eine Periodendauer von 977 oder hexadezimal &3D1. Das Low-Byte müssen wir in R11, das High-Byte in R12 eintragen:

```
CALL adresse,11,&D1
CALL adresse,12,&3
```

Nun müssen wir dem PSG mitteilen, daß die Lautstärke von Kanal B durch die Hüllkurve gesteuert wird; dazu ist Bit 4 von R9 zu setzen:

```
CALL adresse,9,&X10000
```

Um den Ton erklingen zu lassen, muß nur noch das zuständige Bit im Freischaltregister auf 0 gesetzt werden:

```
CALL adresse,7,&X111101
```

Das war's. Um den PSG wieder auszuschalten, geben Sie ein:

```
CALL adresse,7,&X111111
```

## DER PSG-PROGRAMMER

Wie Sie sehen, gestaltet sich die Direktprogrammierung des Soundchips doch recht umständlich. Um Ihnen ein komfortables Tool zur Klangentwicklung in die Hand zu geben, haben wir daher den PSG-PROGRAMMER geschrieben (siehe nachfolgendes Listing). Nach dem Starten des Programms befinden Sie sich im Hauptmenü (rechtes unteres Bildschirmfenster). Vier Menüpunkte stehen zur Auswahl:

<1>eingeben: Soundparameter eingeben.

<2>ausdrucken: Die aktuelle Re- gisterbelegung des PSG wird auf Ihren Drucker ausgegeben.

<3>rücksetzen: Der Psg wird vollständig auf seine Standardwerte zurückgesetzt, also quasi abgeschaltet. Sämtliche bisher eingegebenen Tonwerte gehen verloren.

<4>Tonausgabe abbrechen: Der Ton wird abgeschaltet. Im Gegensatz zu <3> bleiben aber die eingegebenen Werte zur weiteren Bearbeitung erhalten.

Zum Menüpunkt <1> sind noch nähere Erläuterungen erforderlich. Wenn Sie diese Funktion gewählt haben, erscheint ein blinkender Cursor in der linken Bildschirmhälfte. Steuern Sie den Cursor mit den Pfeiltasten zum gewünschten Parameter und geben Sie dann den Wert ein. Alle Eingaben müssen hexadezimal erfolgen; für das Ein-/Ausschaltfeld gilt grundsätzlich: 0=aus, 1=an.
Durch Drücken von H gelangen Sie ins Hilfsmenü, wo Sie zwischen vier Optionen wählen können:

<1> Umrechnung einer Dezimalzahl in den entsprechenden Hexadezimalwert.

<2> Umrechnung Frequenz in Tonperiode.

<3> Umrechnung eines Notenwerts in die entsprechende Tonperiode. Dabei ist der der Note entsprechende Buchstabe einzugeben (C, D, E, F, G, A, B oder H); Halbtöne sind durch ein nachgestelltes »#« oder »b« zu kennzeichnen. Beispiele: A, C#, Eb. Außerdem ist die Oktave anzugeben; dabei steht 0 für die mittlere Oktave (sog. eingestrichene Oktave).

<4> Umrechnung der Flankenlänge der Hüllkurve in die entsprechende Periodendauer. Die Flankenlänge ist in Sekunden einzugeben; dabei sind natürlich auch gebrochene Werte wie 0.75 erlaubt. Durch Drücken der ENTER-Taste beenden Sie die Eingabe. Der PSG-PROGRAMMER errechnet aus den eingegebenen Werten die Daten für den Soundchip und gibt den Ton aus. Anschließend gelangen Sie ins Hauptmenü zurück.

(G.Cebulla)

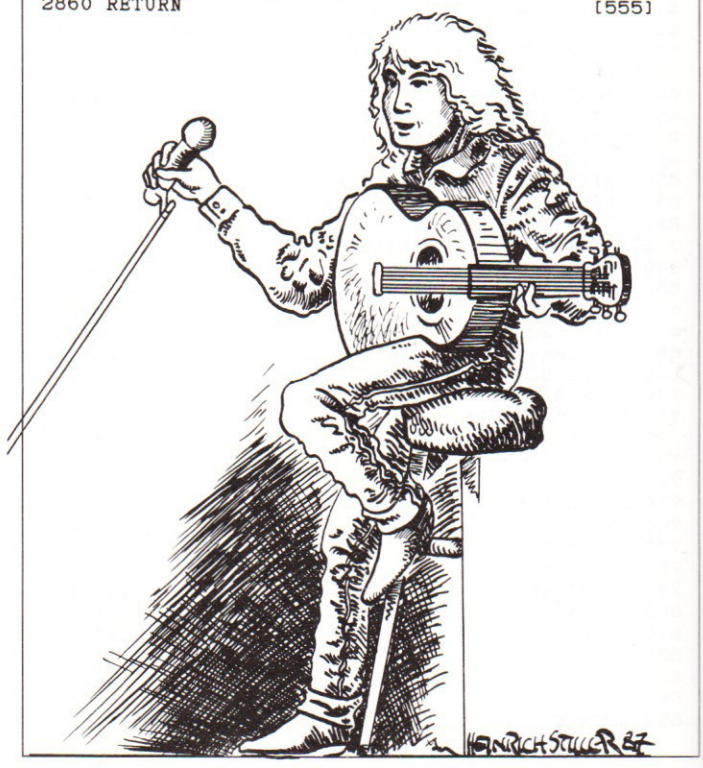
```
10 '******************                          [659]
20 '* PSG-PROGRAMMER *                          [1248]
30 '* (c) G. Cebulla *                          [593]
40 '******************                          [659]
50 '--- Initialisierung                         [2303]
60 DEFINT a-z:WIDTH 20                          [569]
70 DIM x[18],y[18],art[18],laenge[18],wert      [3287]
$[18],reg[13]
80 cur.on=&BB81:cur.off=&BB84                   [1594]
90 okt$="C D EF G A BH"                         [1276]
100 SYMBOL AFTER 255                            [1409]
110 psg=UNT(HIMEM)+1                            [912]
120 SYMBOL AFTER 250                            [1398]
130 '--- Maschinenkode                         [798]
140 DATA DD,7E,02,4B,CD,34,BD,C9                [912]
150 FOR adr=psg TO psg+7                        [1083]
160 READ a$:POKE adr,VAL("&"+a$)                [1712]
170 NEXT adr                                    [547]
180 '--- Zeichendefinition                      [1266]
190 SYMBOL 250,&CC,0,&78,&C,&7C,&CC,&76         [2701]
200 SYMBOL 251,&66,0,&66,&66,&66,&66,&3E        [2367]
210 SYMBOL 252,3,7,&F,&1B,&33,&63,&C3,&C3       [2574]
220 SYMBOL 253,&C3,&C3,&63,&33,&1B,&F,7,3       [2365]
230 SYMBOL 254,&FF                              [880]
240 '--- Daten f. Soundparameter                [1383]
250 DATA 21,5,1,3,29,5,1,3,37,5,1,3             [1885]
260 DATA 22,7,1,1,30,7,1,1,38,7,1,1             [960]
270 DATA 22,9,2,1,30,9,2,1,38,9,2,1             [1501]
280 DATA 22,11,2,1,30,11,2,1,38,11,2,1          [1293]
290 DATA 22,13,2,1,30,13,2,1,38,13,2,1          [1812]
300 DATA 22,16,1,1,20,18,1,4                    [1039]
310 DATA 21,21,2,1,22,21,1,1                    [1183]
```

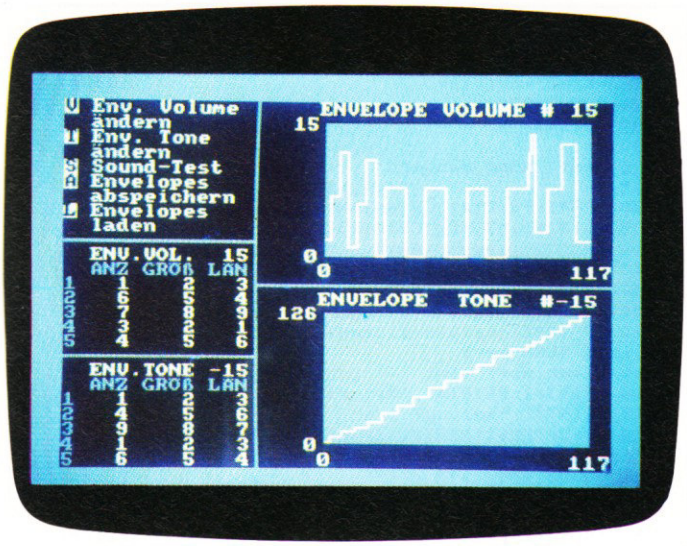
```
320 FOR i=0 TO 18                               [529]
330 READ x[i],y[i],art[i],laenge[i]             [4077]
340 NEXT i                                      [375]
350 '--- Bildschirmaufbau                       [3113]
360 PAPER 0:PEN 1                               [965]
370 INK 0,0:INK 1,0:BORDER 13:MODE 2            [1989]
380 ae$=CHR$(250):ue$=CHR$(251):s$=CHR$(14      [2374]
9)
390 PRINT TAB(43) CHR$(150) STRING$(36,154      [3626]
) CHR$(156);
400 PRINT TAB(18) CHR$(150) STRING$(7,154)      [7723]
 CHR$(158) STRING$(7,154) CHR$(158) STRING
$(7,154) CHR$(156) s$ "  Hilfsmen" ue$ TAB
(80) s$;
410 PRINT TAB(18) s$ "KANAL A" s$ "KANAL B      [4406]
" s$ "KANAL C" s$ s$ "[1] dezimal" TAB(62)
 "--> hexadezimal " s$;
420 PRINT CHR$(150) STRING$(16,154) CHR$(1      [9291]
59) STRING$(7,154) CHR$(159) STRING$(7,154
) CHR$(159) STRING$(7,154) CHR$(157) s$ "[
2] Frequenz" TAB(62) "--> Tonperiode "
s$;
430 PRINT s$ "Tonperiode" TAB(18) s$ TAB(2      [6206]
6) s$ TAB(34) s$ TAB(42) s$ s$ "[3] Notenw
ert" TAB(62) "--> Tonperiode " s$;
440 PRINT s$ TAB(18) s$ TAB(26) s$ TAB(34)      [6051]
 s$ TAB(42) s$ s$ "[4] Flankenl" ae$ "nge
--> Periodendauer" s$;
450 PRINT s$ "Lautst" ae$ "rke" TAB(18) s$      [5794]
 TAB(26) s$ TAB(34) s$ TAB(42) s$ CHR$(147
) STRING$(36,154) CHR$(153);
460 PRINT CHR$(151) STRING$(16,154) CHR$(1      [8028]
```

```
59) STRING$(7,154) CHR$(159) STRING$(7,154
) CHR$(159) STRING$(7,154) CHR$(157) CHR$(
150) STRING$(20,154) CHR$(156) CHR$(150) S
TRING$(14,154) CHR$(156);
470 PRINT s$ "Ton an/aus" TAB(18) s$ TAB(2 [5325]
6) s$ TAB(34) s$ TAB(42) s$ s$ TAB(64) s$
s$ TAB(80) s$;
480 PRINT s$ TAB(18) s$ TAB(26) s$ TAB(34) [3098]
 s$ TAB(42) s$ s$ TAB(64) s$ s$ TAB(80) s$
;
490 PRINT s$ "Rauschen an/aus " s$ TAB(26) [6325]
 s$ TAB(34) s$ TAB(42) s$ s$ TAB(64) s$ s$
 TAB(80) s$;
500 PRINT s$ TAB(18) s$ TAB(26) s$ TAB(34) [3098]
 s$ TAB(42) s$ s$ TAB(64) s$ s$ TAB(80) s$
;
510 PRINT s$ "H" ue$ "llkurve an/aus" s$ T [4967]
AB(26) s$ TAB(34) s$ TAB(42) s$ s$ TAB(64)
 s$ s$ TAB(80) s$;
520 PRINT CHR$(147) STRING$(16,154) CHR$(1 [7993]
55) STRING$(7,154) CHR$(155) STRING$(7,154
) CHR$(155) STRING$(7,154) CHR$(153) s$ TA
B(64) s$ s$ TAB(80) s$:
530 PRINT CHR$(150) STRING$(16,154) CHR$(1 [7173]
58) STRING$(6,154) CHR$(156) CHR$(150) STR
ING$(15,154) CHR$(156) s$ TAB(64) s$ s$ TA
B(80) s$;
540 PRINT s$ "H" ue$ "llkurvenform  " s$ T [8891]
AB(25) s$ s$ "  H" ue$ "llkurven:  " s$ s$
 TAB(64) s$ CHR$(147) STRING$(14,154) CHR$
(153);
550 PRINT s$ TAB(18) s$ TAB(25) s$ s$ TAB( [4487]
42) s$ s$ TAB(64) s$ CHR$(150) STRING$(14,
154) CHR$(156);
560 PRINT s$ "Periodendauer   " s$ TAB(25) [6754]
 s$ s$ "8 " STRING$(3,253) CHR$(205) TAB(3
6) "C " STRING$(4,252) s$ s$ TAB(64) s$ s$
 "  Hauptmen" ue$ TAB(80) s$;
570 PRINT CHR$(147) STRING$(16,154) CHR$(1 [4681]
55) STRING$(6,154) CHR$(153) s$ TAB(42) s$
 s$ TAB(64) s$ s$ "[1] eingeben  " s$;
580 PRINT CHR$(150) STRING$(16,154) CHR$(1 [9465]
58) STRING$(6,154) CHR$(156) s$ "9 " CHR$(
205) STRING$(3,95) TAB(36) "D " CHR$(204)
STRING$(3,254) s$ s$ TAB(64) s$ s$ "[2] au
sdrucken" s$;
590 PRINT s$ "Rauschperiode   " s$ TAB(25) [5182]
 s$ s$ TAB(42) s$ s$ TAB(64) s$ s$ "[3] r"
 ue$ "cksetzen" s$;
600 PRINT CHR$(147) STRING$(16,154) CHR$(1 [9517]
55) STRING$(6,154) CHR$(153) s$ "A " CHR$(
205) CHR$(204) CHR$(205) CHR$(204) TAB(36)
 "E " CHR$(204) CHR$(205) CHR$(204) CHR$(2
05) s$ s$ TAB(64) s$ s$ "[4] Tonausgabe" s
$;
610 PRINT CHR$(150) STRING$(23,154) CHR$(1 [4710]
56) s$ TAB(42) s$ s$ TAB(64) s$ s$ TAB(70)
 "abbrechen " s$;
620 PRINT s$ "CEBUSOFT PSG PROGRAMMER" s$  [10397]
s$ "B " CHR$(253) STRING$(3,254) TAB(36) "
F " CHR$(252) STRING$(3,95) s$ s$ TAB(64)
s$ s$ "Ihre Wahl: [ ]" s$;
630 PRINT CHR$(147) STRING$(23,154) CHR$(1 [8537]
53) CHR$(147) STRING$(15,154) CHR$(153) CH
R$(147) STRING$(20,154) CHR$(153) CHR$(147
) STRING$(14,154) CHR$(153);
640 WINDOW#1,66,79,9,15                    [1063]
650 WINDOW#2,44,63,9,24                    [1119]
660 GOSUB 2400                             [859]
670 INK 1,26                               [56]
680 '--- Hauptmenue (Endlosschleife)       [1874]
690 WHILE NOT feierabend                   [1700]
700 LOCATE 78,24                           [479]
710 PRINT" " CHR$(8);                      [855]
720 GOSUB 2780                             [831]
730 a$=""                                  [388]
740 WHILE a$<"1" OR a$>"4"                 [1593]
750 a$=INKEY$                              [278]
760 WEND                                   [390]
770 i=REMAIN(0)                            [622]
780 CALL cur.off                           [350]
790 PRINT a$;                              [420]
800 ON VAL(a$) GOSUB 850,2340,2400,2550    [2468]
810 WEND                                   [390]
820 :                                      [174]
830 '--- Eingeben                          [1617]
840 :                                      [174]
850 LOCATE 1,1                             [611]
```

```
860 PRINT CHR$(24) CHR$(240) CHR$(24) " "  [11905]
CHR$(24) CHR$(241) CHR$(24) " " CHR$(24) C
HR$(242) CHR$(24) " " CHR$(24) CHR$(243) C
HR$(24) " weiter" SPC(3) CHR$(24) "H" CHR$
(24) " Hilfsmen" ue$ SPC(3) CHR$(24) "ENTE
R" CHR$(24) " Ende"
870 cp=0:nr=0                              [966]
880 LOCATE x[0],y[0]                       [1085]
890 GOSUB 2780                             [831]
900 WHILE a$<>CHR$(13)                     [1953]
910 LOCATE x[nr]+cp,y[nr]                  [1753]
920 a$=""                                  [388]
930 WHILE a$=""                            [933]
940 a$=UPPER$(INKEY$)                      [789]
950 WEND                                   [390]
960 IF a$>=CHR$(240) THEN ON ASC(a$)-239 G [9315]
OSUB 1240,1300,1360,1420 ELSE IF a$>="0" A
ND a$<="9" OR a$>="A" AND a$<="F" THEN GOS
UB 1480 ELSE IF a$="H" THEN GOSUB 1560
970 WEND                                   [390]
980 '--- Eingabe beenden                   [2026]
990 FOR i=0 TO 2                           [471]
1000 reg[2*i]=VAL("&"+RIGHT$(wert$[i],2))  [1666]
1010 reg[2*i+1]=VAL("&"+LEFT$(wert$[i],1)) [2748]
1020 NEXT i                                [375]
1030 reg[6]=VAL("&"+wert$[18]) + 16*VAL(we [1596]
rt$[17])
1040 zw$="&X"                              [996]
1050 FOR i=11 TO 6 STEP-1                  [1133]
1060 zw$=zw$+wert$[i]                      [1106]
1070 NEXT i                                [375]
1080 reg[7]=VAL(zw$) XOR 63                [718]
1090 FOR i=8 TO 10                         [556]
1100 reg[i]=VAL("&"+wert$[i-5]) + 16*VAL(w [3695]
ert$[i+4])
1110 NEXT i                                [375]
1120 reg[11]=VAL("&"+RIGHT$(wert$[16],2))  [4571]
1130 reg[12]=VAL("&"+LEFT$(wert$[16],2))   [2412]
1140 reg[13]=VAL("&"+wert$[15])            [2377]
1150 GOSUB 2600                            [851]
1160 stream=2:GOSUB 2670                   [902]
1170 LOCATE 41,1:PRINT CHR$(17);           [1910]
1180 RETURN                                [555]
1190 :                                     [174]
1200 '--- Subs Eingabe                     [981]
1210 :                                     [174]
1220 '--- Cursor hoch                      [1146]
1230 :                                     [174]
1240 IF nr<3 THEN nr=17 ELSE IF nr<15 THEN [4591]
 nr=nr-3 ELSE nr=nr-1+(nr=18)
1250 cp=0                                  [69]
1260 RETURN                                [555]
1270 :                                     [174]
1280 '--- Cursor runter                    [1323]
1290 :                                     [174]
1300 IF nr<13 THEN nr=nr+3 ELSE IF nr<17 T [2784]
HEN nr=nr+1-(nr=13) ELSE nr=0
1310 cp=0                                  [69]
1320 RETURN                                [555]
1330 :                                     [174]
1340 '--- Cursor links                     [1663]
1350 :                                     [174]
1360 cp=cp-1                               [686]
1370 IF cp<0 THEN nr=nr-1-19*(nr=0):cp=lae [2553]
nge[nr]-1
1380 RETURN                                [555]
1390 :                                     [174]
1400 '--- Cursor rechts                    [1400]
1410 :                                     [174]
1420 cp=cp+1                               [642]
1430 IF cp=laenge[nr] THEN nr=nr+1+19*(nr= [2389]
18):cp=0
1440 RETURN                                [555]
1450 :                                     [174]
1460 '--- Eingabe                          [1440]
1470 :                                     [174]
1480 IF a$>"1" AND art[nr]=2 THEN RETURN   [1469]
1490 MID$(wert$[nr],cp+1)=a$               [1937]
1500 PRINT a$;                             [420]
1510 GOSUB 1420                            [911]
1520 RETURN                                [555]
1530 :                                     [174]
1540 '--- Hilfsmenue                       [1471]
1550 :                                     [174]
1560 LOCATE 56,2                           [663]
1570 PRINT"aktiv. Bitte w" ae$ "hlen: [ ]" [2879]
1580 LOCATE 78,2                           [481]
1590 a$=""                                 [388]
```

```
1600 WHILE a$<"1" OR a$>"4"                          [1593]
1610 a$=INKEY$                                       [278]
1620 WEND                                            [390]
1630 i=REMAIN(0)                                     [622]
1640 CALL cur.off                                    [350]
1650 PRINT a$                                        [399]
1660 ON VAL(a$) GOSUB 1760,1860,1970,2220            [2577]
1670 LOCATE 56,2                                     [663]
1680 PRINT STRING$(24,32)                            [1453]
1690 GOSUB 2780                                      [831]
1700 RETURN                                          [555]
1710 :                                               [174]
1720 '--- Subs Hilfsmenue                            [440]
1730 :                                               [174]
1740 '--- dez -> hex                                 [1252]
1750 :                                               [174]
1760 CLS#1                                           [373]
1770 PRINT#1,"Bitte Dezimal-wert eingeben:           [3323]
";
1780 INPUT#1,dez!                                    [406]
1790 IF dez!>65535 OR dez!<-32768 THEN PRI           [3226]
NT#1,"Unzul" ae$ "ssige" "Eingabe!":GOTO 1
770
1800 PRINT#1,"Hexadezimal:"                          [1640]
1810 PRINT#1,HEX$(dez!,4)                            [1268]
1820 RETURN                                          [555]
1830 :                                               [174]
1840 '--- Frequenz -> Tonperiode                     [1734]
1850 :                                               [174]
1860 CLS#1                                           [373]
1870 PRINT#1,"Bitte Frequenz" "eingeben:"            [2862]
1880 INPUT#1,freq!                                   [626]
1890 per=ROUND(62500/freq!)                          [1341]
1900 IF per<0 OR per>4095 THEN PRINT#1,"Un           [6901]
zul" ae$ "ssige" "Eingabe!":GOTO 1870
1910 PRINT#1,"Tonperiode:"                           [1952]
1920 PRINT#1,HEX$(per,3)                             [991]
1930 RETURN                                          [555]
1940 :                                               [174]
1950 '--- Notenwert -> Tonperiode                    [2127]
1960 :                                               [174]
1970 CLS#1                                           [373]
1980 PRINT#1,"Bitte Noten-" "wert eingeben           [3240]
!";
1990 PRINT#1,"Beispiele:"                            [1099]
2000 PRINT#1,"D  f" ue$ "D"                          [863]
2010 PRINT#1,"D# f" ue$ "Dis"                        [1360]
2020 PRINT#1,"Db f" ue$ "Des"                        [873]
2030 INPUT#1,note$:note$=UPPER$(note$)               [1197]
2040 note=INSTR(okt$,LEFT$(note$,1))                 [2759]
2050 IF note=0 THEN PRINT#1,"Unzul" ae$ "s           [2636]
sige" "Eingabe!":GOTO 1980
2060 IF note=13 THEN note=12                         [882]
2070 IF LEN(note$)=1 THEN 2100                       [1751]
2080 IF RIGHT$(note$,1)"#" THEN note=note+           [1943]
1
2090 IF RIGHT$(note$,1)"B" THEN note=note-           [2029]
1
2100 PRINT#1,"Bitte Oktave" "eingeben!"              [2787]
2110 PRINT#1,"0 = mittlere"                          [1468]
2120 PRINT#1,TAB(5) "Oktave"                         [553]
2130 INPUT#1,oktave                                  [1083]
2140 IF ABS(oktave)>4 THEN PRINT#1,"Unzul"           [2027]
 ae$ "ssige" "Eingabe!":GOTO 2100
2150 per=ROUND(62500/(440*(2^(INT(oktave)+           [3701]
(note-10)/12))))
2160 PRINT#1,"Tonperiode:"                           [1952]
2170 PRINT#1,HEX$(per,3)                             [991]
2180 RETURN                                          [555]
2190 :                                               [174]
2200 '--- Flankenlaenge -> Periodendauer             [2711]
2210 :                                               [174]
2220 CLS#1                                           [373]
2230 PRINT#1,"Bitte Flanken-l" ae$ "nge in           [2741]
"
2240 PRINT#1,"Sekunden" "eingeben:"                  [2211]
2250 INPUT#1,sek!                                    [290]
2260 pdauer!=ROUND(sek!*3096.25)                     [2013]
2270 IF pdauer!<1 OR pdauer!>65535 THEN PR           [6255]
INT#1,"Unzul" ae$ "ssige" "Eingabe!":GOTO
2230
2280 PRINT#1,"Periodendauer:";                       [1347]
2290 PRINT#1,HEX$(pdauer!,4)                         [1470]
2300 RETURN                                          [555]
2310 :                                               [174]
2320 '--- Registerinhalte ausdrucken                 [1636]
2330 :                                               [174]
2340 stream=8                                        [421]
2350 GOSUB 2670                                      [829]
2360 RETURN                                          [555]
2370 :                                               [174]
2380 '--- Register ruecksetzen                       [1003]
2390 :                                               [174]
2400 FOR i=0 TO 13                                   [514]
2410 reg[i]=0                                        [719]
2420 NEXT i                                          [375]
2430 reg[7]=63                                       [328]
2440 GOSUB 2600                                      [851]
2450 stream=2:GOSUB 2670                             [902]
2460 FOR i=0 TO 18                                   [529]
2470 LOCATE x[i],y[i]                                [1347]
2480 wert$[i]=STRING$(laenge[i],"0")                 [2525]
2490 PRINT wert$[i]                                  [729]
2500 NEXT i                                          [375]
2510 RETURN                                          [555]
2520 :                                               [174]
2530 '--- Ton abschalten                             [1506]
2540 :                                               [174]
2550 SOUND 135,0,1,0                                 [1372]
2560 RETURN                                          [555]
2570 :                                               [174]
2580 '--- Register belegen                           [1397]
2590 :                                               [174]
2600 FOR i=0 TO 13                                   [514]
2610 CALL psg,i,reg[i]                               [785]
2620 NEXT i                                          [375]
2630 RETURN                                          [555]
2640 :                                               [174]
2650 '--- Registerinhalte ausgeben                   [1382]
2660 :                                               [174]
2670 LOCATE#2,1,1                                    [652]
2680 PRINT#stream,"REGISTERBELEGUNG:"                [2187]
2690 PRINT#stream,"Reg dez hex bin" ae$ "r           [3113]
"
2700 FOR i=0 TO 13                                   [514]
2710 PRINT#stream,USING"R## ### ";i;reg[i]           [3230]
;
2720 PRINT#stream,HEX$(reg[i],2) TAB(13) B           [3155]
IN$(reg[i],8);
2730 NEXT i                                          [375]
2740 RETURN                                          [555]
2750 :                                               [174]
2760 '--- Cursor anschalten                          [966]
2770 :                                               [174]
2780 CALL cur.on                                     [354]
2790 AFTER 10 GOSUB 2840                             [945]
2800 RETURN                                          [555]
2810 :                                               [174]
2820 '--- Cursor ausschalten                         [1381]
2830 :                                               [174]
2840 CALL cur.off                                    [350]
2850 AFTER 10 GOSUB 2780                             [1005]
2860 RETURN                                          [555]
```






# Envpro

**Gehören Sie auch zu den Usern, die gerne (Computer)-Musik hören? Sollten Sie sich angesprochen fühlen, dann wollen Sie sicher auch mal was eigenes programmieren. Sie werden sich sagen »hab ich doch schon probiert, aber es kam nicht's gescheites heraus«. Nun, mit diesem Programm, erfinden auch Laien die schönsten Musikstücke, die auch noch auf wunderbare Weise in Ihre eignen Basicprogramme eingebunden werden können. Lassen Sie sich in die wundervolle Welt der Computermusik entführen, nachdem Sie das Programmm abgetippt haben.**

Der im CPC eingebaute AY-3-8912-Soundchip wird durch das Basic hervorragend unterstützt. Vor allem wären da die Kommandos ENV und ENT zu nennen, die aus ihm mittels Interruptsteuerung ungeahnte Fähigkeiten herauskitzeln. Leider gehören diese Befehle mit ihren bis zu 16 Parametern auch zu den unanschaulichsten des Schneider-Basic. Hier soll ENVPRO Abhilfe schaffen, indem es die Hüllkurvenprogrammierung sowohl akustisch als auch optisch unterstützt. Nach dem Starten des Programmes sehen Sie einen in drei Text/Daten-Windows und zwei Grafik-Windows unterteilten Bildschirm vor sich. Im Steuerfenster (links oben) erscheint das Hauptmenue:

V – Envelope Volume ändern
T – Envelope Tone ändern
S – Sound-Test
A – Abspeichern
L – Laden

1) Envelope Volume ändern
Durch Drücken von V gelangen Sie in das Unterprogramm zur Bearbeitung einer Lautstärkenhüllkurve. Nach Eingabe der Nummer (1 – 15) werden im ENV – Datenwindow die aktuellen Parameter aufgeführt; falls die Hüllkurve bereits definiert war, wird sie außerdem grafisch dargestellt. Mit den Cursortasten können Sie jetzt die zu ändernden Parameter ansteuern. Da gilt:
–ANZ: Schrittzahl.

–GRÖß: Die Größe der Lautstärkeänderung pro Schritt. Hier sind sowohl positive als auch negative Werte zulässig. Beachten Sie: Wenn die Lautstärke kleiner als Null wird, dann wird der Ton mit der Maximallautstärke 15 fortgesetzt; ebenso wird die Lautstärke bei Überschreiten der 15 auf Null gesetzt. Dadurch lassen sich interessante Effekte erzielen; probieren Sie z.B. einmal aus: ANZ 127, GRÖß –1, LÄN 2.

LÄN: Schrittlänge, bezeichnet die Pausenzeit bis zur nächsten Lautstärkenänderung in Hundertstelsekunden. Nachdem Sie alle Eingaben getätigt haben, gelangen Sie durch Drücken der ENTER-Taste wieder ins Hauptmenü; außerdem wird die Hüllkurve grafisch ausgegeben.

2) Envelope Tone ändern
Wenn Sie eine Tonhüllkurve bearbeiten wollen, müssen Sie T drücken. Die Eingabe erfolgt im wesentlichen wie bei der Lautstärkenhüllkurve. Zusätzlich werden Sie hier noch gefragt, ob die Hüllkurve sich wiederholen soll; das ist z.B. dann nützlich, wenn Sie Sirenengeheul erzeugen wollen. Die Schrittgröße bezeichnet hier die Änderung der Tonperiode relativ zum Ausgangswert. Dabei gilt: Positive Werte machen den Ton tiefer, negative dagegen höher!

3) Sound – Test
Nachdem Sie eine Lautstärken- oder Tonhüllkurve definiert haben, drücken Sie S, um sie auszuprobieren. Die Klaviatur liegt auf den unteren beiden Tastenreihen und umfaßt eineinhalb Oktaven (18 Halbtöne von C bis F); die Tastenbelegung wird im Steuerfenster angezeigt. ENVPRO benutzt alle drei Kanäle des Soundchips, Sie können also bis zu drei Töne gleichzeitig spielen. Weitere Steuermöglichkeiten:

– Cursor hoch/runter: Auf eine höhere bzw. tiefere Oktave umschalten. Dabei stehen Ihnen neun Oktaven von –4 bis +4 zur Verfügung,

– Cursor links/rechts: Rauschperiode verringern bzw. vergrößern. Wertebereich: 0 (kein Rauschen) bis 31 (tiefes Rauschen). Mit ENTER gelangen Sie wieder ins Hauptmenue.

4.) Abspeichern
Wenn Sie Ihre Hüllkurve abspeichern wollen, drücken Sie A. Nach der Eingabe des Dateinamens werden Sie gefragt, ob eine Speicherung im Binär- oder im Basicformat gewünscht ist:
– Binär:

Die Parameter aller Hüllkurven werden vom Interpreter in einem 480 Byte großen Datenblock im System-Ram abgelegt. (Beim CPC 464 ab Adresse & B61A, beim CPC 664=6128 dagegen ab Adresse & B2B6; ENVPRO erkennt übrigens selbständig den verwendeten CPC-Typ und stellt sich darauf ein). Bei der Abspeicherung im Binärformat wird dieser Datenblock als Binärfile aus Diskette/Kassette geschrieben; er läßt sich ohne weitere Maßnahmen mit LOAD wieder laden,

– Basic:
Bei diesem Format wird ein Basic-Programm erzeugt, das Sie mit MERGE in eigene Programme einbauen können. Dabei können Sie den Zeilennummernbereich selbst bestimmen.

5) Laden
Wenn Sie Hüllkurven von Diskette/Kassette laden wollen, drücken Sie L. Voraussetzung ist allerdings, daß das File im Binärformat abgespeichert wurde. Noch ein Hinweis zum Schluß: ENVPRO löscht in Zeile 50 sämtliche Hüllkurven. Falls bereits irgendwelche Hüllkurven im Speicher stehen, die Sie erhalten wollen, dann starten Sie das Programm mit RUN 60.

(G. Cebulla/CD)

```
10 '************************                [577]
20 '**    E n v P r o    **                [409]
30 '** (c)1986 G.Cebulla **                [896]
40 '************************                [577]
50 FOR i=1 TO 15:ENV i:ENT i:NEXT           [826]
60 SYMBOL 251,90,60,102,102,126,102,102     [2221]
70 SYMBOL 252,72,0,120,12,124,204,118       [1859]
80 SYMBOL 253,186,108,198,198,198,108,56    [2170]
90 SYMBOL 254,0,66,60,102,102,102,60        [1799]
100 SYMBOL 255,56,108,104,108,102,102,108   [2159]
110 'initialisierung                        [1300]
120 DEFINT a-z                              [553]
130 IF PEEK(&BCD5)=&B2 THEN ea=&B60A ELSE   [2917]
ea=&B2A6
140 DEF FNf$=STRING$(3-(j=2),35)            [1491]
150 DIM e(3,5),l(3)                         [1158]
160 l(1)=1:l(2)=5:l(3)=10                   [713]
170 t$="ZSXDCVGBHNJM,L.:/\"                 [2288]
180 FOR i=240 TO 243:t$=t$+CHR$(i):NEXT     [1901]
190 t$=t$+CHR$(13)                          [962]
200 'bildschirmaufbau                       [1166]
210 INK 0,0:INK 1,26:INK 2,14:INK 3,26,14   [729]
220 MODE 1:BORDER 14:PAPER 0:PEN 2:CLS      [2244]
230 FOR i=1 TO 25                           [312]
240 LOCATE 15,i                             [720]
250 IF i=10 OR i=18 THEN PRINT CHR$(157) E  [4823]
LSE IF i=13 THEN PRINT CHR$(151) ELSE PRIN
T CHR$(149)
260 NEXT                                    [350]
270 LOCATE 1,10:PRINT STRING$(14,154)       [2139]
280 LOCATE 1,18:PRINT STRING$(14,154)       [1979]
290 LOCATE 16,13:PRINT STRING$(25,154)      [972]
300 LOCATE 3,12:PRINT"ANZ GR"CHR$(253)CHR$  [3758]
(255)" L"CHR$(251)"N"
310 LOCATE 3,20:PRINT"ANZ GR"CHR$(253)CHR$  [3443]
(255)" L"CHR$(251)"N"
320 FOR i=1 TO 5                            [449]
330 LOCATE 1,i+12:PRINT CHR$(i+48)          [1546]
340 LOCATE 1,i+20:PRINT CHR$(i+48)          [1450]
350 NEXT:PEN 1                              [901]
360 LOCATE 20,1:PRINT"ENVELOPE VOLUME #"    [2415]
370 LOCATE 20,14:PRINT"ENVELOPE  TONE  #"   [2503]
380 LOCATE 3,11:PRINT"ENV.VOL."             [2591]
390 LOCATE 3,19:PRINT"ENV.TONE"             [1927]
400 WINDOW#1,1,14,1,9                       [1018]
410 WINDOW#2,12,14,11,11                    [1428]
420 WINDOW#3,37,39,1,1                      [1178]
430 WINDOW#4,3,14,13,17                     [1581]
440 WINDOW#5,12,14,19,19                    [1461]
450 WINDOW#6,37,39,14,14                    [1407]
460 WINDOW#7,3,14,21,25                     [1458]
470 FOR i=1 TO 7:PEN#i,1:PAPER#i,0:NEXT     [1668]
480 ORIGIN 310,232,312,614,376,232:CLG 2    [1417]
490 ORIGIN 310,24,312,614,168,24:CLG 2      [1603]
500 'hauptmenue - endlosschleife            [2236]
510 WHILE 1                                 [921]
520 PRINT#1,CHR$(24)"V"CHR$(24)" Env. Volu  [2418]
me"
530 PRINT#1,TAB(3)CHR$(252)"ndern"          [1907]
540 PRINT#1,CHR$(24)"T"CHR$(24)" Env. Tone  [2848]
"
550 PRINT#1,TAB(3)CHR$(252)"ndern"          [1907]
560 PRINT#1,CHR$(24)"S"CHR$(24)" Sound-Tes  [2327]
t"
570 PRINT#1,CHR$(24)"A"CHR$(24)" Envelopes  [1653]
"
580 PRINT#1,TAB(3)"abspeichern"             [2338]
590 PRINT#1,CHR$(24)"L"CHR$(24)" Envelopes  [2303]
"
600 PRINT#1,TAB(3)"laden"                   [1136]
610 a$=UPPER$(INKEY$)                       [789]
620 IF a$="V" THEN GOSUB 700 ELSE IF a$="T  [10153]
" THEN GOSUB 820 ELSE IF a$="S" THEN GOSUB
 1960 ELSE IF a$="A" THEN GOSUB 2320 ELSE
IF a$="L" THEN GOSUB 2850 ELSE 610
630 c=REMAIN(0):CLS#1                       [1718]
640 WEND                                    [390]
650 '                                       [117]
660 '*** unterprogramme ***                 [1727]
670 '                                       [117]
680 'env. volume aendern                    [1584]
690 '                                       [117]
700 CLS#1:PRINT#1,CHR$(24)"ENVELOPE"        [3080]
710 PRINT#1,"VOLUME "CHR$(252)"ndern"CHR$(  [2606]
24)
720 PRINT#1:INPUT#1,"ENV-Nr. (1-15)";en     [2572]
730 IF en<1 OR en>15 THEN 700               [1580]
740 IF en=vn THEN n=0 ELSE n=1              [1689]
750 IF PEEK(ea+&10*en)=0 THEN ne=0 ELSE ne  [2442]
=1
760 vn=en:wn=4:tf=0                         [949]
770 GOSUB 970                               [842]
780 RETURN                                  [555]
790 '                                       [117]
800 'env. tone aendern                      [1277]
810 '                                       [117]
820 CLS#1:PRINT#1,CHR$(24)"ENVELOPE"        [3080]
830 PRINT#1,"TONE "CHR$(252)"ndern"CHR$(24  [2280]
)
840 PRINT#1:INPUT#1,"ENT-Nr. (1-15)";en     [2556]
850 IF en<1 OR en>15 THEN 820               [1636]
860 IF en=tn THEN n=0 ELSE n=1              [1448]
870 IF (PEEK(ea+&F0+&10*en) AND 127)=0 THE  [1739]
N ne=0 ELSE ne=1
880 tn=en:wn=7:tf=1                         [530]
890 PRINT#1:PRINT#1,"Mit Wieder-"           [1315]
900 PRINT#1,"holung? j/n"                   [2031]
910 a$=UPPER$(INKEY$):IF a$="J" THEN en=-e  [4306]
n ELSE IF a$<>"N" THEN 910
920 GOSUB 970                               [842]
930 RETURN                                  [555]
940 '                                       [117]
950 'huellkurvenparameter eingeben          [1444]
960 '                                       [117]
970 FOR i=1 TO 2                            [436]
980 PRINT#wn-1,USING"###";en;               [1045]
990 NEXT                                    [350]
1000 a=ea+&F0*tf+&10*ABS(en)                [2239]
1010 z=PEEK(a) AND 127:b=a                  [1848]
1020 FOR i=1 TO 5                           [449]
1030 FOR j=1 TO 3                           [618]
1040 b=b+1                                  [569]
1050 IF i>z THEN e(j,i)=0 ELSE e=PEEK(b):e  [4963]
(j,i)=e+256*(e>127 AND j=2)
1060 LOCATE#wn,l(j),i:PRINT#wn,USING FNf$;  [3135]
e(j,i);
1070 NEXT j,i                               [396]
1080 ON n GOSUB 1610                        [643]
1090 GOSUB 1380                             [935]
1100 i=1:j=1:c=0:in=0                       [1132]
1110 WHILE c<>13                            [1300]
1120 ae=0:c=0:a$=""                         [585]
1130 LOCATE#wn,l(j),i:PRINT#wn,CHR$(24)USI  [4047]
NG FNf$;e(j,i);
1140 WHILE c<240 AND c<>13                  [1977]
1150 WHILE c$="":c$=INKEY$:WEND             [3068]
1160 c=ASC(c$)                              [1214]
1170 IF (c=45 AND a$="" AND j=2) OR (c>47   [9624]
AND c<58 AND LEN(a$)<LEN(FNf$)) THEN ae=1:
in=1:a$=a$+CHR$(c) ELSE IF c=127 AND a$<>"
" THEN a$=LEFT$(a$,LEN(a$)-1)
1180 LOCATE#wn,l(j),i:PRINT#wn,SPC(LEN(FNf  [3753]
$)-LEN(a$))a$;
1190 c$=""                                  [165]
1200 WEND                                   [390]
1210 v=VAL(a$):PRINT#wn,CHR$(24);           [2836]
1220 IF (j=1 AND v>127+112*tf) OR (j=2 AND  [4850]
 (v<-128 OR v>127)) OR (j=3 AND v>255) THE
N c=0:GOSUB 1480:GOTO 1250 ELSE IF ae THEN
 e(j,i)=v
1230 LOCATE#wn,l(j),i:PRINT#wn,USING FNf$;  [3135]
e(j,i);
1240 IF c=240 AND i>1 THEN i=i-1 ELSE IF c  [6309]
=241 AND i<5 THEN i=i+1 ELSE IF c=242 AND
j>1 THEN j=j-1 ELSE IF c=243 AND j<3 THEN
j=j+1
1250 WEND                                   [390]
1260 b=a                                    [555]
1270 FOR i=1 TO 5                           [449]
1280 FOR j=1 TO 3                           [618]
1290 b=b+1                                  [569]
1300 POKE b,e(j,i)-256*(e(j,i)<0 AND j=2)   [1953]
1310 IF j=1 AND e(j,i)=0 THEN z=i-1:POKE a  [3410]
,z+(en AND 128) ELSE NEXT j,i:z=5:POKE a,z
+(en AND 128)
1320 ne=1                                   [179]
1330 ON in GOSUB 1610                       [711]
1340 RETURN                                 [555]
1350 '                                      [117]
1360 'hilfsmenue huellkurveneingabe         [1901]
1370 '                                      [117]
1380 CLS#1:PRINT#1,CHR$(24)"CURSOR"CHR$(24  [2938]
)";"
1390 PRINT#1:PRINT#1,"Parameter"            [1515]
1400 PRINT#1,"ansteuern"                    [1771]
1410 PRINT#1:PRINT#1,CHR$(24)"ENTER"CHR$(2  [2213]
4)";"
1420 PRINT#1:PRINT#1,"Eingabe"              [2202]
1430 PRINT#1,"beenden"                      [1876]
1440 RETURN                                 [555]
```

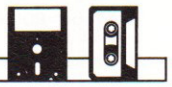
```
1450 '                                                  [117]
1460 'eingabefehler                                     [1188]
1470 '                                                  [117]
1480 CLS#1:PEN 3                                        [512]
1490 PRINT#1,"EINGABEFEHLER!";                          [1976]
1500 PEN 1:PRINT#1,"Schrittanzahl:";                    [2552]
1510 PRINT#1,"0..."USING"###";127+112*tf                [2187]
1520 PRINT#1:PRINT#1,"Schrittgr"CHR$(254)C              [2169]
HR$(255)"e:"
1530 PRINT#1,"-128...127"                               [902]
1540 PRINT#1:PRINT#1,"Schrittl"CHR$(252)"n              [2259]
ge:"
1550 PRINT#1,"0...255"                                  [755]
1560 AFTER 500 GOSUB 1380                               [568]
1570 RETURN                                             [555]
1580 '                                                  [117]
1590 'huellkurve zeichnen                               [717]
1600 '                                                  [117]
1610 ORIGIN 310,232-208*tf,312,614,376-208              [2880]
*tf,232-208*tf:CLG 2
1620 IF ne=0 THEN RETURN                                [785]
1630 mi=0:ma=0:tg=0:v=0                                 [1530]
1640 FOR i=1 TO z                                       [428]
1650 tg=tg+e(1,i)*e(3,i)                                [1167]
1660 v=v+e(1,i)*e(2,i)                                  [1239]
1670 ma=MAX(ma,v)                                       [1215]
1680 mi=MIN(mi,v)                                       [683]
1690 NEXT                                               [350]
1700 IF tf=0 THEN ma=15:mi=0                            [703]
1710 LOCATE 16,2+13*tf:PRINT USING"####";m              [1762]
a
1720 LOCATE 16,11+13*tf:PRINT USING"####";              [2100]
mi
1730 LOCATE 19,12+13*tf:PRINT 0                         [1600]
1740 LOCATE 36+(tg<100),12+13*tf:PRINT USI              [1858]
NG"#####";tg;
1750 IF tg<100 THEN PRINT" ";                           [660]
1760 IF tg=0 THEN tg=1                                  [1558]
1770 t!=304/tg                                          [478]
1780 m=ma-mi:IF m=0 THEN m=1                            [1853]
1790 v!=144/m                                           [774]
1800 v0=-mi*v!                                          [1017]
1810 IF tf THEN MOVE 0,v0:DRAWR 340,0,0                 [1653]
1820 x=0:y=v0:MOVE x,y                                  [1590]
1830 v=0:t=0                                            [943]
1840 FOR i=1 TO z                                       [428]
1850 FOR j=1 TO e(1,i)                                  [1270]
1860 v=v+e(2,i)                                         [1013]
1870 WHILE v>ma:v=mi+v-ma-1:WEND                        [2245]
1880 WHILE v<mi:v=ma+v-mi+1:WEND                        [1763]
1890 y=v0+v*v!:DRAW x,y,1                               [1161]
1900 t=t+e(3,i):x=t*t!:DRAW x,y                         [704]
1910 NEXT j,i                                           [396]
1920 RETURN                                             [555]
1930 '                                                  [117]
1940 'sound test                                        [1043]
1950 '                                                  [117]
1960 CLS#1:PEN#1,0:PAPER#1,2                            [1099]
1970 PRINT#1," KEYBOARD: "                              [1359]
1980 ORIGIN 0,382,0,174,382,368:CLG 2                   [1387]
1990 TAG:PLOT 8,0,0                                     [1023]
2000 PRINT"SD GHJ L*";                                  [1975]
2010 TAGOFF                                             [1066]
2020 PRINT#1:PRINT#1,"ZXCVBNM<>?\"                      [1442]
2030 PEN#1,1:PAPER#1,0                                  [830]
2040 PRINT#1,CHR$(24)CHR$(240)CHR$(24)" "C              [4134]
HR$(24)CHR$(241)CHR$(24)"  Oktave"
2050 PRINT#1,TAB(6)CHR$(252)"ndern"                     [1215]
2060 PRINT#1,CHR$(24)CHR$(242)CHR$(24)" "C              [4865]
HR$(24)CHR$(243)CHR$(24)"  Noise"
2070 PRINT#1,TAB(6)CHR$(252)"ndern"                     [1215]
2080 PRINT#1,CHR$(24)"ENTER"CHR$(24)"  End              [1829]
e"
2090 PRINT#1,TAB(3)"NOISE:  0"                          [1120]
2100 ok=0:ka=1:noise=0                                  [1291]
2110 FOR i=-4 TO 4                                      [668]
2120 LOCATE#1,13,5-i                                    [597]
2130 IF i=0 THEN PRINT#1,CHR$(24);                      [1632]
2140 PRINT#1,USING"+#";i;                               [1073]
2150 IF i=0 THEN PRINT#1,CHR$(24);                      [1632]
2160 NEXT                                               [350]
2170 a$="":WHILE a$="":a$=UPPER$(INKEY$):W              [2173]
END
2180 i=INSTR(t$,a$)                                     [736]
2190 IF i=0 THEN 2170                                   [1190]
2200 WHILE i<>23                                        [968]
2210 IF i=19 OR i=20 THEN LOCATE#1,13,5-ok              [8759]
:PRINT#1,USING"+#";ok;:ok=ok-(i=19 AND ok<
4)+(i=20 AND ok>-4):LOCATE#1,13,5-ok:PRINT
#1,USING CHR$(24)+"+#"+CHR$(24);ok;:GOTO 2
170
2220 IF i=21 OR i=22 THEN noise=noise+(i=2               [5753]
1 AND noise>0)-(i=22 AND noise<31):LOCATE#
1,10,9:PRINT#1,USING"##";noise:GOTO 2170
2230 pe=ROUND(62500/(440*(2^(ok+(i-10)/12)              [2787]
)))
2240 ka=ka*2:IF ka>4 THEN ka=1                          [1937]
2250 SOUND ka+128,pe,0,0,vn,tn,noise                    [2363]
2260 GOTO 2170                                          [361]
2270 WEND                                               [390]
2280 RETURN                                             [555]
2290 '                                                  [117]
2300 'huellkurven abspeichern                           [1494]
2310 '                                                  [117]
2320 CLS#1:PRINT#1,CHR$(24)"Envelopes"                  [2873]
2330 PRINT#1,"abspeichern"CHR$(24)                      [2734]
2340 PRINT#1:INPUT#1,"Name";n$                          [1876]
2350 IF LEN(n$)<1 THEN RETURN                           [1811]
2360 IF LEN(n$)>8 THEN 2340                             [1092]
2370 PRINT#1:PRINT#1,"Bitte Format"                     [2151]
2380 PRINT#1,"w"CHR$(252)"hlen:"                        [2612]
2390 PRINT#1,CHR$(24)"1"CHR$(24)" Bin"CHR$              [2655]
(252)"r"
2400 PRINT#1,CHR$(24)"2"CHR$(24)" Basic"                [2582]
2410 WHILE a$<>"1" AND a$<>"2":a$=INKEY$:W              [1605]
END
2420 WINDOW SWAP 1,0:CLS                                [1434]
2430 ON VAL(a$) GOSUB 2490,2540                         [1680]
2440 WINDOW SWAP 0,1                                    [1031]
2450 RETURN                                             [555]
2460 '                                                  [117]
2470 'binaer abspeichern                                [1945]
2480 '                                                  [117]
2490 SAVE n$+".env",b,ea+&10,&1E0                       [1739]
2500 RETURN                                             [555]
2510 '                                                  [117]
2520 'basic-file generieren                             [1509]
2530 '                                                  [117]
2540 INPUT"1.Zeilennr.:  ",zn                           [1793]
2550 IF zn=0 THEN zn=10                                 [1204]
2560 INPUT"Zeilenabstand:",za                           [1913]
2570 IF za=0 THEN za=10                                 [1316]
2580 OPENOUT n$+".bas"                                  [1272]
2590 FOR ev=1 TO 15                                     [360]
2600 ad=ea+&10*ev                                       [994]
2610 z=PEEK(ad)                                         [1290]
2620 IF z THEN z$=STR$(zn)+"ENV"+STR$(ev):              [3278]
GOSUB 2740:zn=zn+za
2630 NEXT                                               [350]
2640 FOR et=1 TO 15                                     [578]
2650 ad=ea+&F0+&10*et                                   [622]
2660 z=PEEK(ad)                                         [1290]
2670 IF z AND 127 THEN z$=STR$(zn)+"ENT"+S              [5169]
TR$(et+2*et*(z>128)):GOSUB 2740:zn=zn+za
2680 NEXT                                               [350]
2690 CLOSEOUT                                           [902]
2700 RETURN                                             [555]
2710 '                                                  [117]
2720 'basic-zeile erzeugen                              [1326]
2730 '                                                  [117]
2740 FOR i=1 TO z AND 127                               [577]
2750 FOR j=1 TO 3                                       [618]
2760 ad=ad+1                                            [315]
2770 w=PEEK(ad)                                         [949]
2780 z$=z$+","+STR$(w+256*(w>127 AND j=2))              [2092]
2790 NEXT j,i                                           [396]
2800 PRINT#9,z$                                         [1016]
2810 RETURN                                             [555]
2820 '                                                  [117]
2830 'huellkurven laden                                 [619]
2840 '                                                  [117]
2850 CLS#1:PRINT#1,CHR$(24)"Envelopes"                  [2873]
2860 PRINT#1,"laden"CHR$(24)                            [2083]
2870 PRINT#1:INPUT#1,"Name";n$                          [1876]
2880 IF LEN(n$)<1 THEN RETURN                           [1811]
2890 IF LEN(n$)>8 THEN 2870                             [1330]
2900 WINDOW SWAP 1,0:CLS                                [1434]
2910 LOAD n$+".env",ea+&10                              [1275]
2920 WINDOW SWAP 0,1                                    [1031]
2930 RETURN                                             [555]
```

# SOUND-DEMO

Diese beiden Sound-Demo Programme zeigen in eindrucksvoller Weise, was man bei etwas Geschick aus den drei Sound-Kanälen des CPC 464 herausholen kann.

(Jens Köplinger)

**für 464**



```
10 '====================================  [2282]
20 '  ****                          ****   [278]
30 '   **        SOUND - DEMO        **   [1414]
40 '   **                            **   [104]
50 '   ** written and composed 1987 **    [1702]
60 '   **     by JENS KOEPLINGER     **   [665]
70 '   **                            **   [104]
80 '  ****                          ****   [278]
90 '====================================  [2282]
100 :                                      [174]
110 '                                      [117]
120 ' ENVs und ENTs festlegen              [1638]
130 '                                      [117]
140 ENV 1,1,15,1,1,0,10,2,-1,5,10,-1,20    [1459]
150 ENT 1: REM ENT-1 in Normalzustand brin [3137]
gen
160 ENV 2,1,15,1,1,0,10,4,-1,5,10,-1,30    [1317]
170 ENT -2,2,1,1,2,-1,1,2,-1,1,2,1,1       [1162]
180 ENV 3,1,15,1,1,0,5,14,-1,4             [1178]
190 ENV 4,1,13,1,1,0,10,2,-1,5,10,-1,30    [2083]
200 ENV 5,1,15,1,1,0,10,1,-2,5,7,-1,10     [2207]
210 '                                      [117]
220 ' ** DATAs fuer SOUND                  [986]
230 '                                      [117]
240 ' Negative Zahl = Aenderung der Klangf [3031]
arbe
250 '                                      [117]
260 ' ==== THEMA 1 normal                  [1402]
270 DATA -1,1,106,89,71,119,95,71,127,106, [3834]
80,134,106,80,142,213,179,159,142,119,106,
71,89,106,-2,125
280 ' ==== THEMA 1 variiert                [1380]
290 DATA -1,2,106,89,71,119,95,71,127,106, [4523]
80,134,106,80,142,213,179,159,142,119,106,
142,119,106,-2,22
300 ' ==== Ueberleitung in rasanten Teil   [1975]
310 DATA -1,3,851,2,0,1,1,851,1,716,1,0,1, [4011]
2,851,2,956,2,1136,2,1136,2,0,2,3
320 DATA 851,2,0,1,1,851,1,716,1,0,1,2,851 [3627]
,1,638,1,568,3,478,1,568,2,0,2,3
330 ' ==== Ueberleitung variiert           [2588]
340 DATA 851,2,0,1,1,851,1,716,1,0,1,2,851 [3722]
,2,956,2,1136,2,1136,2,0,2,3
350 DATA 851,2,0,1,1,851,1,716,1,0,1,2,851 [3886]
,1,638,1,568,3,478,1,426,2,0,2,3
360 ' ==== CHINA - THEMA alleine           [1786]
370 DATA -1,4,53,1,71,53,1,71,53,1,71,53,1 [3348]
,71,60,2,80,60,2,80,71,2,95,71,2,95,60,2,8
0,0,2,3
380 DATA 53,1,71,53,1,71,53,1,71,53,1,71,6 [4364]
0,2,80,60,2,80,53,1,71,45,1,60,53,1,71,60,
1,80,53,1,71,0,1,2,0,2,1
390 ' ==== CHINA - THEMA - Begleitung alle [3059]
ine
400 DATA -1,3,851,1,851,1,638,1,716,1,851, [5462]
1,956,1,1136,1,1136,1,851,1,851,1,638,1,71
6,1,851,1,956,1,1136,1,0,1,2
410 DATA 851,1,851,1,638,1,716,1,851,1,956 [4839]
,1,1136,1,1136,1,851,1,851,1,638,1,716,1,1
136,1,956,1,851,1,0,1,3
420 ' ==== und nun beides zusammen         [1630]
430 DATA -1,5,851,53,71,851,53,71,638,53,7 [7520]
1,716,53,71,851,60,80,956,0,0,1136,60,80,1
136,0,0,851,71,95,851,0,0,638,71,95,716,60
,80,851,0,0,956,0,0,1136,0,0,0,0,0,2
440 DATA 851,53,71,851,53,71,638,53,71,716 [7930]
,53,71,851,60,80,956,0,0,1136,60,80,1136,0
,0,851,53,71,851,45,60,638,53,71,716,60,80
,1136,53,71,956,0,0,851,0,0,0,0,0,3
450 ' ==== CHINA - THEMA mit Begleitung wi [4056]
ederholt
460 DATA 851,53,71,851,53,71,638,53,71,716 [8042]
,53,71,851,60,80,956,0,0,1136,60,80,1136,0
,0,851,71,95,851,0,0,638,71,95,716,60,80,8
51,0,0,956,0,0,1136,0,0,0,0,0,2
470 DATA 851,53,71,851,53,71,638,53,71,716 [7930]
,53,71,851,60,80,956,0,0,1136,60,80,1136,0
,0,851,53,71,851,45,60,638,53,71,716,60,80
,1136,53,71,956,0,0,851,0,0,0,0,0,3
480 ' ==== Ueberleitung zur Wiederholung   [2201]
490 DATA -1,3,1136,4,568,2,0,1,2,568,1,758 [4474]
,1,716,2,758,1,1136,1,0,1,2,0,2,3
500 DATA -1,4,142,4,71,71,2,36,0,1,2,71,2, [4067]
36,95,1,47,89,1,45,95,1,47,36,1,71,0,1,2,0
,2,3
510 ' ==== und noch mal CHINA              [1209]
520 DATA -1,5,851,53,71,851,53,71,638,53,7 [7318]
1,716,53,71,851,60,80,956,0,0,1136,60,80,1
136,0,0,851,71,95,851,0,0,638,71,95,716,0,
0,851,60,80,956,0,0,1136,0,0,0,0,0,2
530 DATA 851,53,71,851,53,71,638,53,71,716 [8332]
,53,71,851,60,80,956,0,0,1136,60,80,1136,0
,0,851,53,71,851,45,60,638,53,71,716,60,80
,1136,53,71,956,0,0,851,0,0,0,0,0,3,-2,125
540 ' ==== Anfangsthema                    [2252]
550 DATA -1,1,106,89,71,119,95,71,127,106, [3834]
80,134,106,80,142,213,179,159,142,119,106,
71,89,106,-2,125
560 DATA -1,2,106,89,71,119,95,71,127,106, [7859]
80,134,-2,2,106,-2,2,80,-2,2,142,-2,2,213,
-2,2,179,-2,4,159,-2,7,142,-2,10,119,-2,14
,106,-2,22,142,-2,44,201,-2,50,213,-2,100
570 DATA -1,3,0,10,3                       [672]
990 DATA -3, *** ENDE ***                  [1000]
1000 '                                     [117]
1010 ' DATAZEILEN einlesen                 [963]
1020 '                                     [117]
1030 RESTORE                               [621]
1040 zeit=TIME:' Zeitpunkt festhalten      [2763]
1050 READ a                                [428]
1060 IF a>=0 THEN 1150                     [993]
1070 IF a=-3 THEN 1710:' ENDE ? Wenn ja, n [2263]
ach 1260
1080 IF a=-1 THEN READ art:GOTO 1050: 'Neu [4415]
e Klangfarbe ?
1090 ' PAUSE machen                        [990]
1100 READ dauer                            [921]
1110 GOSUB 1630                            [861]
1120 GOTO 1050                             [305]
1130 '                                     [117]
1140 ' Klangfarben spielen                 [1091]
1150 ON art GOTO 1170,1230,1290,1410,1510  [1671]
1160 ' Normal (1)                          [1069]
1170 dauer=8                               [371]
1180 k=(k+1) MOD 3                         [96]
1190 SOUND 128+2^k,a,-1,0,1,1              [1904]
1200 GOSUB 1630                            [861]
1210 GOTO 1050                             [305]
1220 ' Schwingend (2)                      [1442]
1230 dauer=8                               [371]
1240 k=(k+1) MOD 3                         [96]
```

```
1250 SOUND 128+2^k,a,-1,0,1,2                          [1899]
1260 GOSUB 1630                                        [861]
1270 GOTO 1050                                         [305]
1280 ' einstimmig mit Rhythmus (3)                     [3197]
1290 READ d                                            [421]
1300 dauer=d*6                                         [667]
1310 IF a<>0 THEN SOUND 129,a,1000,0,5,1:G             [3853]
OSUB 1630:GOTO 1050
1320 ' Routine fuer Rhythmus                           [1973]
1330 READ k                                            [458]
1340 ON k GOTO 1350,1360,1370                          [2195]
1350 SOUND 132,a,-1,15,2,1,31:GOTO 1380                [2001]
1360 SOUND 132,a,-1,15,3,1,1:GOTO 1380                 [2015]
1370 SOUND 132,2000,-1,15,1,1,31                       [2223]
1380 GOSUB 1630                                        [861]
1390 GOTO 1050                                         [305]
1400 ' zweistimmig mit Rhythmus (4)                    [2875]
1410 READ d                                            [421]
1420 dauer=d*6                                         [667]
1430 IF a=0 THEN 1490                                  [530]
1440 READ b                                            [431]
1450 SOUND 129,a,1000,0,4,2                            [1569]
1460 SOUND 130,b,1000,0,4,2                            [1561]
1470 GOSUB 1630                                        [861]
1480 GOTO 1050                                         [305]
1490 GOTO 1330                                         [448]
1500 ' dreistimmig mit Rhythmus (5)                    [1931]
1510 dauer=6                                           [349]
1520 READ b,c                                          [547]
1530 IF a=0 THEN 1330                                  [538]
1540 SOUND 132,a,1000,0,2,1                            [1532]
1550 IF b=0 THEN 1580                                  [969]
1560 SOUND 129,b,1000,0,4,2                            [1943]
1570 SOUND 130,c,1000,0,4,2                            [1543]
1580 GOSUB 1630                                        [861]
1590 GOTO 1050                                         [305]
1600 '                                                 [117]
1610 ' SUBRoutine fuer Warten                          [2056]
1620 '                                                 [117]
1630 flag=0                                            [263]
1640 AFTER dauer GOSUB 1680                            [919]
1650 WHILE flag=0                                      [837]
1660 WEND                                              [390]
1670 RETURN                                            [555]
1680 flag=1                                            [262]
1690 RETURN                                            [555]
1700 ' **** ENDE; Zeit errechnen und ausge             [3563]
ben
1710 s=(TIME-zeit)/300                                 [807]
1720 std=s\3600:s=s-std*3600                           [2064]
1730 MI=s\60:s=s-MI*60                                 [1361]
1740 PRINT "Dauer:"                                    [1351]
1750 PRINT USING"## Minute(n) ## Sekunden"             [3672]
;MI;ROUND(s)

20000 '================================== [2282]
==
20010 '  ****                          **** [278]

20020 '   **       SOUND - DEMO         **  [1414]
20030 '   **                            **  [104]
20040 '   ** written and composed 1987 **  [1702]
20050 '   **      JENS KOEPLINGER       **  [2175]
20060 '   **                            **  [104]
20070 '  ****                          **** [278]
20080 '================================== [2282]
==
20090 ENV 1,3,4,1,1,0,5,12,-1,10:ENV 2,1,1 [1983]
2,1,1,0,10,12,-1,30
20100 '                                                [117]
20110 ' Einsetzen der Begleitmelodie                   [2700]
20120 AFTER 86 GOSUB 20380                             [351]
20130 '                                                [117]
20140 ' DATAs 1. Stimme                                [1174]
```

```
20150 DATA 213,3,190,1,179,3,213,1,179,3,1 [7189]
59,1,142,3,213,1,127,1,159,1,213,6,127,1,1
59,1,213,2,106,1,127,1,159,2,142,1,179,1,2
13,6,142,1,179,1,213,2,179,1,213,1,284,2,1
59,1,213,1,253,6
20160 DATA 159,1,213,1,253,2,213,1,253,1,3 [11200]
19,2,179,1,239,1,284,6,179,1,239,1,284,2,2
39,1,284,1,358,2,190,1,239,1,284,6,190,1,2
39,1,284,2,190,1,239,1,284,2,284,1,190,1,1
90,1,142,1,142,1,95,1,95,1,71,1,71,1,47,1,
47,1,36,1,36,4
20170 '                                                [117]
20180 ' DATAs 2. Stimme                                [831]
20190 DATA 568,851,426,1136,568,851,426,11 [10405]
36,568,851,426,1136,568,851,426,1136, 638,
851,426,1276,638,851,426,1276,638,851,426,
1276,638,851,426,1276, 568,851,426,1136,56
8,851,426,1136,568,851,426,1136,568,851,42
6,1136
20200 DATA 638,851,426,1276,638,851,426,12 [10905]
76,638,851,426,1276,638,851,426,1276, 716,
956,478,1432,716,956,478,1432,716,956,478,
1432,716,956,478,1432, 568,758,379,1136,56
8,758,379,1136,568,758,379,1136,568,758,37
9,1136
20210 DATA 568,638,379,1136,568,638,379,11 [3297]
36,568,638,379,1136,568,638,379,1136
20220 '                                                [117]
20230 ' DATAs 3.Stimme                                 [718]
20240 DATA 851,851,851,851,851,851,851,851 [10864]
,1276,1276,1276,1276,1276,1276,1276,1276,8
51,851,851,851,851,851,851,851,1276,1276,1
276,1276,1276,1276,1276,1276,1432,1432,143
2,1432,1432,1432,1432,1432,1136,1136,1136,
1136,1136,1136,1136,1136
20250 DATA 1136,1136,1136,1136,1136,1136,1 [1866]
136,1136
20260 DATA -1,-1                                       [418]
20270 '                                                [117]
20280 ' Ablaufprogramm                                 [1082]
20290 '                                                [117]
20300 DIM t1(67),l1(67),t2(112),t3(56)                 [2435]
20310 RESTORE 20150:FOR g=1 TO 66:READ a,1 [5745]
1(g):t1(g)=a:NEXT:FOR g=1 TO 112:READ a:t2
(g)=a/2:NEXT:ton2=1:FOR g=1 TO 56:READ t3(
g):NEXT
20320 GOSUB 20390                                      [992]
20330 RETURN:' Den Rest machen die INTERUP [1740]
TS
20340 d=(d+16) MOD 32:OUT &F60E,d:SOUND 4, [3225]
t3((ton2+1)/2),40,2,2,1:RETURN
20350 ton2=ton2+1:IF ton2>112 THEN ton2=1  [2957]
20360 SOUND 2,t2(ton2),20,1,2,1                        [1834]
20370 RETURN                                           [555]
20380 EVERY 20 GOSUB 20340:EVERY 10,2 GOSU [3684]
B 20350:RETURN
20390 ton=ton+1:IF ton=67 THEN ton=1                   [1520]
20400 SOUND 1,t1(ton),l1(ton)*20,3,1,1                 [2843]
20410 p=p/1/2/3:AFTER l1(ton)*9,1 GOSUB 20 [2858]
390:RETURN
```





# COLFUNKT

**Colfunk ist ein Programm, mit welchem man sehr schöne Grafiken erstellen kann, und zwar aus simplen algebraischen Gleichungen.**

Nachdem Sie das Programm eingetippt, abgespeichert und gestartet haben, werden Sie aufgefordert, die Funktion, welche darzustellen ist, einzugeben. Hier einige Beispiele:

**z= sin(x)*y+cos(y)*x-sqr(abs(x*y))**
**z= sqr(abs(x*y))**
**z= tan(x)*y**
**z= x-sin(y)*cos(x)+y**

Danach werden Sie noch nach den XMin, XMax bzw. YMin und YMax Werten gefragt (sinnvollerweise von −1000 bis 1000) und Sie müssen zwischen verschiedenen Darstellungsarten und Bildschirmmodi wählen.

Dann legt das Programm mit der Darstellung der Funktion los (kann etwas länger dauern).

Wenn es fertig ist, haben Sie noch die Möglichkeit, den Screen abzuspeichern, wonach einer weiteren Bearbeitung nichts mehr im Wege steht.

(Michael Wessel)

**für 464**

```
100 '                                          [117]
110 '(C) 1987 by Michael Wessel . MW-Soft      [2163]
29.5.1987
120 '                                          [117]
130 '                                          [117]
140 'Integer-Variablen                         [253]
150 '                                          [117]
160 DEFINT a-w                                 [556]
170 '                                          [117]
180 'Maschinenprogramm                         [1465]
190 '                                          [117]
200 MEMORY &4FFF                               [174]
210 FOR adr=&5000 TO &5078:READ a$:POKE ad     [4376]
r,VAL("&"+a$):NEXT
220 DATA 01,09,50,21,24,50,C3,D1               [1331]
230 DATA BC,14,50,C3,28,50,C3,3A               [1070]
240 DATA 50,C3,4C,50,53,43,52,D3               [1020]
250 DATA 53,43,52,CC,50,55,54,4C               [1040]
260 DATA 49,4E,C5,00,FC,A6,09,50               [714]
270 DATA FE,01,C0,DD,56,01,DD,5E               [204]
280 DATA 00,21,00,C0,01,FF,3F,ED               [1152]
290 DATA B0,C9,FE,01,C0,DD,66,01               [1518]
300 DATA DD,6E,00,11,00,C0,01,FF               [1062]
310 DATA 3F,ED,B0,C9,FE,01,C0,D5               [1917]
320 DATA 21,A4,AC,06,FF,36,00,23               [1420]
330 DATA 10,FB,DD,E1,DD,4E,00,DD               [1230]
340 DATA 66,02,DD,6E,01,11,A4,AC               [1843]
350 DATA 06,00,ED,B0,97,12,21,A4               [1143]
360 DATA AC,CD,00,B9,CD,BC,E6,C9               [1730]
370 DATA 00,00,00,00,00,00,00,00               [1005]
380 '                                          [117]
390 'Einbinden                                 [1017]
400 '                                          [117]
410 svscr=adr+1                                [1302]
420 CALL &5000                                 [668]
430 '                                          [117]
440 'Farben und Toene                          [2247]
450 '                                          [117]
460 RESTORE 500                                [775]
470 DIM t(15)                                  [178]
480 FOR i=15 TO 1 STEP -1:READ a:INK i,a:N     [2565]
EXT
490 FOR i=0 TO 15:READ t(i):NEXT               [1196]
500 DATA 3,6,15,24,25,22,21,12,13,1,2,5,4,     [1260]
14,26
510 DATA 478,451,426,402,379,358,338,319,3     [3632]
01,284,268,253,239,225,213,201,190
520 INK 0,0:BORDER 0                           [620]
530 ENV 1,15,-1,4:ENV 2,15,-1,10,7,1,5         [1882]
540 '                                          [117]
550 'Funktion definieren                       [1408]
560 '                                          [117]
```

```
570 MODE 2                                          [513]
580 WINDOW #1,1,80,5,13                             [948]
590 WINDOW #2,1,80,14,25                            [1077]
600 LOCATE 36,1:PRINT "COLFUNKT"                    [2139]
610 LOCATE 20,2:PRINT "Programm zur Darste          [5492]
llung von Funktionen ."
620 LOCATE 22,3:PRINT CHR$(164)+" 1987 by           [3994]
Michael Wessel . MW-Soft ."
630 WINDOW SWAP 0,1                                 [1031]
640 PRINT "Bitte definieren Sie Ihre Funkt          [4017]
ion :"
650 PRINT                                           [361]
660 b$="720 Def Fn"                                 [601]
670 PRINT "z(x,y)=";:INPUT "",c$                    [1524]
680 c$=" z(x,y)="+c$                                [617]
690 a$=b$+c$                                        [471]
700 |PUTLINE,@a$                                    [299]
710 PRINT #2,"Funktion :"+c$                        [1685]
720 '**                                             [175]
730 '                                               [117]
740 'Parameter eingeben                             [1031]
750 '                                               [117]
760 CLS                                             [91]
770 PRINT "Definition :"+c$                         [2030]
780 PRINT                                           [361]
790 INPUT "Geben Sie X-Minimum und X-Maxim          [5411]
um an (Xmin,Xmax , wobei Xmin < Xmax) :",x
mi,xma
800 IF xmi>=xma THEN 790                            [900]
810 PRINT                                           [361]
820 INPUT "Geben Sie Y-Minimum und Y-Maxim          [5702]
um an (Ymin,Ymax , wobei Ymin < Ymax) :",y
mi,yma
830 IF ymi>=yma THEN 820                            [1557]
840 PRINT #2                                        [317]
850 PRINT #2,"X-Bereich von "xmi"bis "xma           [3129]
860 PRINT #2,"Y-Bereich von "ymi"bis "yma           [2356]
870 CLS                                             [91]
880 PRINT "Darstellungsart der Funktion"+c          [4408]
$+" :"
890 PRINT                                           [361]
900 PRINT "1 = ...Ganzer Schirm ausgefuell          [4031]
t (graphisch)"
910 PRINT "2 = ....Einmal an Y=0 gespiegel          [3849]
t (graphisch)"
920 PRINT "3 = An X=0 und an Y=0 gespiegel          [3610]
t (graphisch)"
930 PRINT "4 = .....Als Computer-Soun               [3965]
d (akustisch)"
940 PRINT                                           [361]
950 INPUT "Waehlen Sie bitte :",n:IF n<1 O          [3349]
R n>4 THEN 880
960 PRINT #2                                        [317]
970 PRINT #2,"Darstellungsart Nr"n                  [2364]
980 IF n=4 THEN n=1:so=1:mo=-1:GOTO 1130            [2504]
990 CLS                                             [91]
1000 PRINT "Screen-Modi :"                          [1787]
1010 PRINT                                          [361]
1020 PRINT "0 = 16 Farben , 32000 Pixels"           [1760]
1030 PRINT "1 = 4  Farben , 64000 Pixels"           [2228]
1040 PRINT "2 = 2 Farben , 128000 Pixels"           [2761]
1050 PRINT                                          [361]
1060 INPUT "Setzen Sie den Modus :",mo:IF           [3743]
mo>2 OR mo<0 THEN 1000
1070 PRINT #2                                       [317]
1080 PRINT #2,"Screen-Modus"mo                      [1402]
1090 CLS                                            [91]
1100 INPUT "Geschwindikeitssteigerung um 1          [9180]
0% (kein Abruch durch ESC moeglich !) <J>
<N> ";a$:a$=UPPER$(a$)
1110 PRINT #2                                       [317]
1120 IF a$="J" THEN PRINT #2,"Schneller Ze          [8310]
ichenmodus" ELSE PRINT #2,"Langsamer Zeich
enmodus"
1130 PRINT #2                                       [317]
1140 INPUT #2,"Alles korekt <J> <N> ";k$:I          [3292]
F UPPER$(k$)<>"J" THEN 570
1150 '                                              [117]
1160 'Zeichen-Modus setzen                          [1601]
1170 '                                              [117]
1180 IF mo>-1 THEN MODE mo                          [1053]
1190 ORIGIN 320,200                                 [215]
1200 '                                              [117]
1210 'Ausschnitt und Vergroesserungsfaktor          [2939]
 berechnen
1220 '                                              [117]
1230 st=4-(2*mo)+INT(mo/2)                          [1140]
1240 xrn=640/(ROUND(1/(4-n))+1)                     [2212]
1250 xn=ABS(xma-xmi)/(xrn/st)                       [2189]
1260 yrn=200/(ROUND(n/2))                           [1971]
1270 yn=ABS(yma-ymi)/yrn                            [1089]
1280 na=(1-(1 MOD n))*319-((1 MOD n)*st)            [966]
1290 nb=(1 MOD (4-n))*199-(1-(1 MOD (4-n))          [1702]
)
1300 x=xmi:y=ymi                                    [950]
1310 '                                              [117]
1320 'Funktion darstellen                           [1394]
1330 '                                              [117]
1340 IF a$="J" THEN POKE &B939,&C9                  [1693]
1350 FOR a=-320 TO na STEP st                       [2122]
1360 FOR b=-200 TO nb STEP 2                        [1233]
1370 c=ABS(FN z(x,y) MOD 16)                        [1074]
1380 IF so THEN IF c<>cc THEN cc=c:SOUND 1          [6032]
29,t(c),32767,0,2:SOUND 132,4,32767,15,1,,
c:GOTO 1450 ELSE 1450
1390 PLOT a,b,c                                     [746]
1400 IF n=1 THEN 1450                               [872]
1410 PLOT -a-1,b                                    [304]
1420 IF n<>3 THEN 1450                              [586]
1430 PLOT a,-b-1                                    [686]
1440 PLOT -a-1,-b-1                                 [164]
1450 y=y+yn                                         [749]
1460 NEXT:x=x+xn:y=ymi:NEXT                         [1942]
1470 IF so THEN CLS:PRINT "Das war's . END          [4520]
E !":PRINT "<Taste druecken>":CALL &BB06:R
UN
1480 '                                              [117]
1490 'Abspeichern                                   [1169]
1500 '                                              [117]
1510 POKE &B939,&F3                                 [487]
1520 |SCRS,svscr                                    [1397]
1530 MODE 2                                         [513]
1540 PRINT "Ihre Funktion"+c$                       [1937]
1550 PRINT "ist nun vollstaendig im Bereic          [4311]
h fuer"
1560 PRINT xmi"<= X <= "xma"und "ymi"<= Y           [4237]
<= "yma"gezeichnet ."
1570 PRINT                                          [361]
1580 INPUT "Wollen Sie das erzeugte Bild a          [5470]
bspeichern <J> <N> ";a$
1590 a$=UPPER$(a$)                                  [533]
1600 IF a$<>"J" THEN MODE mo:|SCRL,svscr:C          [2486]
ALL &BB06:RUN
1610 PRINT                                          [361]
1620 INPUT "Geben Sie dann bitte den Namen          [4469]
 des Bildes ein : ",name$
1630 MODE mo:|SCRL,svscr                            [1045]
1640 IF LEN(name$)=0 OR LEN(name$)>8 THEN           [1908]
1520
1650 SAVE name$+".pic",b,&C000,&4000                [2495]
1660 RUN                                            [243]
```

# Morse

## *Ein (CW)-Code Lernprogramm...*

**...nicht nur für angegehende Funkamateure der A bzw. B-Lizenz sondern auch für alle die Ihre Kenntnisse auffrischen wollen oder die ganz einfach Spaß an diesen Tönen haben.**

**AMATEURFUNK DIE BRUECKE ZUR WELT**
so meldet sich der CPC nach Drücken der Funktionstaste 'f6 und zwar sprichwörtlich in »Wort und Ton«.
Die Morse-Telegrafie (CW), im Jahre 1837 von Samuel Morse entwickelt, ist selbst heute noch im internationalen Funkverkehr nicht wegzudenken. Das Morsealphabet besteht aus unterschiedlichen Kombinationen von kurzen und langen Impulsen mit dazwischenliegenden Pausen. Jede dieser Kombinationen entspricht einem bestimmten Buchstaben, einer bestimmten Ziffer oder einem bestimmten Zeichen. Die Striche und Pausen haben eine definierte Länge wobei ein Strich die 3fache Länge eines Punktes, die Pause zwischen den Zeichen die Länge eines Punktes, die Pause zwischen Buchstaben die Länge eines Striches und der Abstand zwischen zwei Worten 5 Punkte haben soll.

### Das Programm

Nachdem Sie es gestartet haben, werden Sie nach dem Datum und der Uhrzeit gefragt. Dann können Sie auch schon mit dem Morsen loslegen. Außerdem sind die Funktionstasten f0 bis f6 mit veschiedenen Morsezeichen belegt. Die restlichen Funktionstasten sind für den Umgang mit z.B. dem Diskettenlaufwerk bestimmt. Der Anschluß einer Morsetaste ist vorgesehen. Dazu wird eine Junkers an den Stift 6 und 8 der Joystick-Buchse angeschlossen. Alle eingegebenen Zeichen werden in Dezimalzeichen umgewandelt und entsprechend auf die Diskette gelegt. So werden alle Zeichen oder nur vorgewählte 6 Zeichen mit einem Zufallsgenerator sortiert und anschließend ausgegeben. Um die gleichen Zeichen immer wieder zu lernen, benutzt man die Taste 'f7 um diese auf die Diskette zu sichern. (Nicht im Einzelzeichenmodus vorgesehen, es wäre ja unsinnig. Dieser Menuepunkt dient zur Einprägung von Einzelzeichen wobei immer wieder die gleiche Taste gedrückt wird.) Nach einer gewissen Lernzeit kennt man die Zeichen auswendig. Um einen Selbstbetrug zu vermeiden, läßt man sich neue Zeichenfolgen sortieren und lernt diese auswendig.

Die Festtexte sollten genau abgetippt werden. Natürlich können Sie sich eigene Texte aus Büchern entnehmen, abtippen und auf die Diskette legen. Unnötige Leerzeichen sollten vermieden werden. Beim 'saven werden die Texte am Ende automatisch mit einem Stern gekennzeichnet.

### Das Abtippen

Zuerst tippen Sie Listing 1 ab und legen es auf die Diskette mit »SAVE"MORSE« ab.(Anmerkung: Wird für den Programmablauf nicht unbedingt benötigt. Wer Arbeit sparen will verzichtet einfach darauf.)

Anschließend tippen Sie Listing 2 ab und legen es auf die Diskette mit »SAVE"CWLERN« ab.

Viel Spaß beim Morsen

(Hans J. Ziegler)

Festtext a: ZUM ERLERNEN DER MORSEZEICHEN UND HALTUNG DER MORSETASTE GIBT ES VER SCHIEDENE PUBLIKATIONEN WIE DIES RICHTIG GEHANDHABT WIRD. AUCH VERSCHIEDENE ME THODEN SOLLEN DIE LERNFAEHIGKEIT ERHOE HEN.ICH BIN ALL DIESEN METHODEN AUS DEM WEG GEGANGEN UND HABE MITTELS EINES ZU FALLGENERATORS DIE ZEICHEN ERMITTELT.EINS SOLLTEN SIE AUF JEDEN FALL BEDENKEN.DAS RICHTIGE ERLERNEN MIT DEM GROESSTMOEGLI CHEN RESULTAT KANN NUR IN EINER GRUPPE ERFOLGEN.DAS SELBSTERLERNEN WIRD ZWAR DURCH DEN COMPUTER WESENTLICH ERLEI CHERT.ZUSAETZLICH SOLLTEN SIE WENN SIE UN TERWEGS SIND VERSUCHEN DIE VERSCHIEDE NEN BUCHSTABEN DURCH PHONETISCHE UEBUNGEN ZU ERLERNEN,DAZU EIGNEN SICH ALLE TEXTE AUCH ORTSSCHILDER KOENNEN DAZU DEGRADIERT WERDEN Z.B.: HUNDSANGEN » DI DI DI DIT DI DI DAH DAH DIT DAH DI DIT D I DI DIT DI DAH DAH DIT DAH DAH DIT DIT DAH DIT «

Festtext b: DIE FRAGEN UND ANTWORTEN ZUR FACHLICHEN PRUEFUNG FUER FUNKAMATEU RE,ZUSAMMENGESTELLT UND HERAUSGEGE BEN VOM FERNMELDETECHNISCHEN ZENTRA LAMT DER DEUTSCHEN BUNDESPOST KANN MAN AN JEDER POSTDIENSTSTELLE ERHALTEN FUER NUR EIN PAAR MARK.DIESE ZUSAMMEN STELLUNG VON FRAGEN UND ANTWORTEN WEI CHER ALS DIENSTBEHELF WIE AUCH ALS HILFS MITTEL ZUR VORBEREITUNG AUF DIE AMATEUR FUNKPRUEFUNG DIENT,IST AUCH GRUNDLAGE DER PRUEFUNGSFRAGEBOGEN UND SOLL LEDIG LICH AUF DAS LERNZIEL HINWEISEN UND KANN SOMIT INHALTLICH DIE VIELSEITIGKEIT DER HAN DELSUEBLICHEN LEHRBUECHER NICHT ERSET ZEN. DER DEUTSCHE AMATEUR-RADIO-CLUB E.V. HAT BEI DER GESTALTUNG DES FACHLICHEN BE REICHES MITGEWIRKT. PROGRAMM UND VORLA GEN VON DK9PM IM DARC ORTSVERBAND WALL MEROD DOK K37 UND EISENBAHN-FUNKAMA TEUR-GRUPPE (EFA) LIMBURG.INFORMATIONEN VON JEDEM FUNKAMATEUR ODER VOM DARC.

```
10 MODE 1                                          [506]
20 LOCATE 5,4:PRINT CHR$(24)STRING$(8,144) [3639]
CHR$(24)
30 LOCATE 3,5:PRINT CHR$(24)STRING$(4,144) [4238]
CHR$(24)STRING$(4,32)CHR$(24)STRING$(4,144
)CHR$(24)
40 LOCATE 3,6:PRINT CHR$(24)STRING$(4,144) [8130]
CHR$(24)STRING$(4,32)CHR$(24)STRING$(4,144
)CHR$(24)
50 LOCATE 1,7:PRINT CHR$(24)STRING$(4,144) [2426]
CHR$(24)
60 LOCATE 1,8:PRINT CHR$(24)STRING$(4,144) [3479]
CHR$(24)
70 LOCATE 1,9:PRINT CHR$(24)STRING$(4,144) [2740]
CHR$(24)
80 LOCATE 1,10:PRINT CHR$(24)STRING$(4,144 [2260]
)CHR$(24)
90 LOCATE 1,11:PRINT CHR$(24)STRING$(4,144 [2927]
)CHR$(24)
100 LOCATE 1,12:PRINT CHR$(24)STRING$(4,14 [2377]
4)CHR$(24)
110 LOCATE 3,13:PRINT CHR$(24)STRING$(4,14 [7631]
4)CHR$(24)STRING$(4,32)CHR$(24)STRING$(4,1
44)CHR$(24)
120 LOCATE 3,14:PRINT CHR$(24)STRING$(4,14 [3642]
4)CHR$(24)STRING$(4,32)CHR$(24)STRING$(4,1
44)CHR$(24)
130 LOCATE 5,15:PRINT CHR$(24)STRING$(8,14 [3237]
4)CHR$(24)
140 LOCATE 5,16:PRINT CHR$(24)STRING$(8,14 [2985]
4)CHR$(24)
150 LOCATE 20,7:PRINT CHR$(24)STRING$(4,14 [4332]
4)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,1
44)CHR$(24)
160 LOCATE 20,8:PRINT CHR$(24)STRING$(4,14 [4969]
4)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,1
44)CHR$(24)
170 LOCATE 20,9:PRINT CHR$(24)STRING$(4,14 [6800]
4)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,1
44)CHR$(24)
180 LOCATE 20,10:PRINT CHR$(24)STRING$(4,1 [4355]
44)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,
144)CHR$(24)
190 LOCATE 21,11:PRINT CHR$(24)STRING$(3,1 [4907]
44)CHR$(24)STRING$(6,32)CHR$(24)STRING$(3,
144)CHR$(24)
200 LOCATE 21,12:PRINT CHR$(24)STRING$(3,1 [7711]
44)CHR$(24)"  "CHR$(24)STRING$(2,144)CHR$(
24)"  "CHR$(24)STRING$(3,144)CHR$(24)
210 LOCATE 21,13:PRINT CHR$(24)STRING$(3,1 [5965]
44)CHR$(24)"  "CHR$(24)STRING$(2,144)CHR$(
24)"  "CHR$(24)STRING$(3,144)CHR$(24)
220 LOCATE 21,14:PRINT CHR$(24)STRING$(12, [3219]
144)CHR$(24)
230 LOCATE 21,15:PRINT CHR$(24)STRING$(12, [2809]
144)CHR$(24)
240 LOCATE 20,16:PRINT CHR$(24)STRING$(6,1 [5655]
44)CHR$(24)"  "CHR$(24)STRING$(6,144)CHR$(
24)
250 LOCATE 20,17:PRINT CHR$(24)STRING$(6,1 [4139]
44)CHR$(24)"  "CHR$(24)STRING$(6,144)CHR$(
24)
260 LOCATE 20,18:PRINT CHR$(24)STRING$(4,1 [3911]
44)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,
144)CHR$(24)
270 LOCATE 20,19:PRINT CHR$(24)STRING$(4,1 [5391]
44)CHR$(24)STRING$(6,32)CHR$(24)STRING$(4,
144)CHR$(24)
280 LOCATE 1,21:PRINT STRING$(40,144)       [2287]
290 LOCATE 1,23:PRINT CHR$(24)"ooo"CHR$(24 [10737]
)    " "   CHR$(24)"o"CHR$(24)    " "    C
HR$(24)"ooo"CHR$(24)    " "    CHR$(24)"o"
CHR$(24)"       "CHR$(24)"o"CHR$(24)   "
"    CHR$(24)"ooo"CHR$(24)    " "CHR$(24)"
ooo"CHR$(24)
300 LOCATE 1,25:PRINT STRING$(40,144);      [1972]
310 LOCATE 1,1:PRINT "MORSE - LERNPROGRAMM [5562]
"STRING$(12,32)CHR$(24)CHR$(164)"dk9pm"CHR
$(24)
320 RUN"cwlern.bas"                                [813]
```

---

```
330 'cqcqcqcqcqcqcqcqcqcqcqcqcqcq          [2324]
340 'cq   Ziegler Hans Joachim   cq         [964]
350 'cq   DK 9 PM  <>  DARC-K37  cq         [1065]
360 'cqcqcqcqcqcqcqcqcqcqcqcqcqcq          [2324]
370 '                                      [117]
380 hz=800:ton=78:wpm=75:dauer=6.3:aut1=0  [2021]
390 KEY DEF 66,0,0,0,0                     [1054]
400 KEY DEF 19,0,0,0,0                     [839]
410 KEY DEF 17,0,0,0,0                     [866]
420 KEY DEF 22,0,0,0,0                     [1088]
430 KEY DEF 26,0,0,0,0                     [599]
440 KEY DEF 24,0,0,0,0                     [1113]
450 KEY 128,"\i"+CHR$(13)                  [1417]
460 KEY 129,"\s"+CHR$(13)                  [1289]
470 KEY 130,"\a"+CHR$(13)                  [751]
480 KEY 131,"\k"+CHR$(13)                  [749]
490 KEY 132,"\e"+CHR$(13)                  [1750]
500 KEY 133,"\v"+CHR$(13)                  [1090]
510 KEY 134," AMATEURFUNK DIE BRUECKE ZUR  [3661]
WELT "
520 KEY 135,"*7"+CHR$(13)                  [1397]
530 KEY 136,"*8"+CHR$(13)                  [700]
540 KEY 137,"*9"+CHR$(13)                  [1094]
550 DIM m(96,12)                           [174]
560 FOR z=38 TO 96:FOR z1=1 TO 9 :READ m(z [3669]
,z1):NEXT z1,z
570 DATA 5,0,0,0,0,0,0,0,0                 [743]
580 DATA 5,0,0,0,0,0,0,0,0                 [743]
590 DATA 3,1,3,3,1,5,0,0,0                 [577]
600 DATA 3,1,3,3,1,3,5,0,0                 [844]
610 DATA 5,0,0,0,0,0,0,0,0                 [743]
620 DATA 1,3,1,3,1,5,0,0,0                 [886]
630 DATA 3,3,1,1,3,3,5,0,0                 [848]
640 DATA 3,1,1,1,3,5,0,0,0                 [763]
650 DATA 1,3,1,3,1,3,5,0,0                 [775]
660 DATA 3,1,1,3,1,5,0,0,0                 [718]
670 DATA 3,3,3,3,3,5,0,0,0                 [693]
680 DATA 1,3,3,3,3,5,0,0,0                 [1276]
690 DATA 1,1,3,3,3,5,0,0,0                 [791]
700 DATA 1,1,1,3,3,5,0,0,0                 [1211]
710 DATA 1,1,1,1,3,5,0,0,0                 [780]
720 DATA 1,1,1,1,1,5,0,0,0                 [623]
730 DATA 3,1,1,1,1,5,0,0,0                 [757]
740 DATA 3,3,1,1,1,5,0,0,0                 [279]
750 DATA 3,3,3,1,1,5,0,0,0                 [859]
760 DATA 3,3,3,3,1,5,0,0,0                 [867]
770 DATA 3,3,3,1,1,1,5,0,0                 [939]
780 DATA 3,1,3,1,3,1,5,0,0                 [555]
790 DATA 5,0,0,0,0,0,0,0,0                 [743]
800 DATA 3,1,1,1,3,5,0,0,0                 [763]
810 DATA 5,0,0,0,0,0,0,0,0                 [743]
820 DATA 1,1,3,3,1,1,5,0,0                 [1121]
830 DATA 5,0,0,0,0,0,0,0,0                 [743]
840 DATA 1,3,5,0,0,0,0,0,0                 [890]
850 DATA 3,1,1,1,5,0,0,0,0                 [566]
860 DATA 3,1,3,1,5,0,0,0,0                 [784]
870 DATA 3,1,1,5,0,0,0,0,0                 [801]
880 DATA 1,5,0,0,0,0,0,0,0                 [928]
```

```
890 DATA 1,1,3,1,5,0,0,0,0                    [967]
900 DATA 3,3,1,5,0,0,0,0,0                    [922]
910 DATA 1,1,1,1,5,0,0,0,0                    [895]
920 DATA 1,1,5,0,0,0,0,0,0                    [1013]
930 DATA 1,3,3,3,5,0,0,0,0                    [1095]
940 DATA 3,1,3,5,0,0,0,0,0                    [762]
950 DATA 1,3,1,1,5,0,0,0,0                    [995]
960 DATA 3,3,5,0,0,0,0,0,0                    [585]
970 DATA 3,1,5,0,0,0,0,0,0                    [972]
980 DATA 3,3,3,5,0,0,0,0,0                    [645]
990 DATA 1,3,3,1,5,0,0,0,0                    [1038]
1000 DATA 3,3,1,3,5,0,0,0,0                   [733]
1010 DATA 1,3,1,5,0,0,0,0,0                   [1059]
1020 DATA 1,1,1,5,0,0,0,0,0                   [1032]
1030 DATA 3,5,0,0,0,0,0,0,0                   [674]
1040 DATA 1,1,3,5,0,0,0,0,0                   [1216]
1050 DATA 1,1,1,3,5,0,0,0,0                   [1274]
1060 DATA 1,3,3,5,0,0,0,0,0                   [1344]
1070 DATA 3,1,1,3,5,0,0,0,0                   [857]
1080 DATA 3,1,3,3,5,0,0,0,0                   [697]
1090 DATA 3,3,1,1,5,0,0,0,0                   [433]
1100 DATA 1,1,1,3,1,3,5,0,0                   [1231]
1110 DATA 1,3,1,3,1,5,0,0,0                   [886]
1120 DATA 3,1,3,1,3,5,0,0,0                   [478]
1130 DATA 1,3,1,1,1,5,0,0,0                   [1376]
1140 DATA 1,1,1,3,1,5,0,0,0                   [1221]
1150 DATA 1,1,1,1,1,1,1,1,5                   [914]
1160 MODE 2                                   [513]
1170 WINDOW #7,60,80,1,25                     [1763]
1180 WINDOW #0,1,56,5,25                      [1169]
1190 WINDOW#1,1,56,1,1                        [1248]
1200 WINDOW #2,1,56,3,4                       [1390]
1210 WINDOW#4,58,58,1,25:PRINT#4,STRING$(2    [2705]
5,127);
1220 WINDOW#5,78,78,1,25:PRINT#5,STRING$(2    [3101]
5,127);
1230 PRINT#7,CHR$(24)"  (c) DK9PM HJZ  "CH    [2349]
R$(24)
1240 LOCATE#7,1,3:PRINT#7,"f0 = Irrung"       [1462]
1250 LOCATE#7,1,5:PRINT#7,"f1 = Verkehrsen    [2294]
de"
1260 LOCATE#7,1,7:PRINT#7,"f2 = Spruchende    [2643]
"
1270 LOCATE#7,1,9:PRINT#7,"f3 = Spruchanfa    [2791]
ng"
1280 LOCATE#7,1,11:PRINT#7,"f4 = Warten"      [1741]
1290 LOCATE#7,1,13:PRINT#7,"f5 = Verstande    [3590]
n"
1300 LOCATE#7,1,15:PRINT#7,CHR$(24)" MORSE    [4361]
  <>  (CW) "CHR$(24)
1310 LOCATE#7,1,16:PRINT#7,CHR$(24)" Lern     [2578]
- Programm "CHR$(24)
1320 LOCATE#7,1,17:PRINT#7,CHR$(24)" BPM:     [8824]
ca.        ";:LOCATE#7,10,17:PRINT#7,wpm:L
OCATE#7,1,18:PRINT#7," TON: Hz         ";:
LOCATE#7,10,18:PRINT#7,hz;CHR$(24)
1330 LOCATE#7,1,20:PRINT#7,"f7 = DISKETTE"    [2843]
1340 LOCATE#7,1,21:PRINT#7,"f8 = DRUCKEN"     [1788]
1350 LOCATE#7,1,22:PRINT#7,"f9 = MENUE"       [2906]
1360 LOCATE#7,1,23:PRINT#7,"DLE= Unterbrec    [3575]
hen"
1370 LOCATE#7,1,25:dat1$="10.10.86":PRINT#    [3070]
7,CHR$(24)dat1$+"   00:00 "CHR$(24);
1380 SPEED KEY 30,2                           [1207]
1390 ON ERROR GOTO 5810                       [1465]
1400 GOSUB 5230                               [909]
1410 EVERY 3000 GOSUB 5300                    [1306]
1420 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq        [2488]
cqcq
1430 'cqcqcq     Zeichen mit CW-Code     cq   [1154]
cqcq
```

```
1440 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq      [2488]
cqcq
1450 CLS                                      [91]
1460 PRINT#1,"DRUECKEN SIE EIN BUCHSTABE O     [2755]
DER ZAHL !!!"
1470 LOCATE 1,1:PRINT"                    "   [3986]
:LOCATE 1,1:PRINT CHR$(24)"? "CHR$(24)" >"
;
1480 a$=INKEY$:IF a$="" THEN 1480             [1433]
1490 IF a$="\" THEN f=1:GOTO 1470             [2154]
1500 IF a$="*" THEN ff=1:GOTO 1470            [2391]
1510 IF a$="s" AND f=1 THEN l=91:PRINT"SK"    [2839]
;:GOTO 1610
1520 IF a$="a" AND f=1 THEN l=92:PRINT"AR"    [2840]
;:GOTO 1610
1530 IF a$="k" AND f=1 THEN l=93:PRINT"KA"    [4654]
;:GOTO 1610
1540 IF a$="e" AND f=1 THEN l=94:PRINT"EB"    [2208]
;:GOTO 1610
1550 IF a$="v" AND f=1 THEN l=95:PRINT"VE"    [3660]
;:GOTO 1610
1560 IF a$="i" AND f=1 THEN l=96:PRINT"IRR    [2413]
UNG";:GOTO 1610
1570 IF a$="7" AND ff=1 THEN 3190             [2005]
1580 IF a$="9" AND ff=1 THEN 3210             [1255]
1590 IF a$=" " THEN 1470                      [639]
1600 a$=UPPER$(a$):PRINT a$;:l=ASC(a$)        [1846]
1610 IF l>96 OR l<38 THEN 1470                [2246]
1620 PRINT" ";                                [673]
1630 FOR a=1 TO 9                             [767]
1640 IF m(l,a)=5 THEN a=9:lae=3:GOTO 1690     [2243]
1650 IF m(l,a)=1 THEN PRINT CHR$(144);        [2889]
1660 IF m(l,a)=3 THEN PRINT"-";               [1914]
1670 IF m(l,a)>0 THEN ton1=ton:lae=m(l,a)     [2150]
1680 SOUND 1,ton1,lae*dauer,15                [2169]
1690 SOUND 1,0,dauer,0                        [1409]
1700 NEXT                                     [350]
1710 f=0:PRINT                                [505]
1720 SOUND 1,0,3*dauer,0                      [1840]
1730 GOTO 1470                                [397]
1740 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq        [2488]
cqcq
1750 'cqcqcq          Morseband          cq   [811]
cqcq
1760 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq        [2488]
cqcq
1770 aut1=0:ff=0:DIM zuf(16,56)               [1951]
1780 CLS:CLS#1:b$="":a$=""                    [557]
1790 PRINT#1,CHR$(24)STRING$(56," ")          [1927]
1800 PRINT"-1- Ausgabe des CW 'Zeichens je     [3451]
 Tastendruck"
1810 PRINT:PRINT"-2- Festtexte anfertigen"     [3099]
1820 PRINT:PRINT"-3- Festtext einlesen"        [2165]
1830 PRINT:PRINT"-4- Files umbenennen"         [2178]
1840 a$=INKEY$:IF a$="" THEN 1840              [1456]
1850 IF a$="*" THEN f=1:GOTO 1840              [1641]
1860 IF a$="9" THEN f=0:ON ERROR GOTO 0:ER     [3532]
ASE zuf:GOTO 3210
1870 IF a$="1" THEN CLS:f7=0:aut1=0:PRINT#     [4507]
1,"Druecken sie eine Taste":GOTO 2880
1880 IF a$="2" THEN 1920                       [718]
1890 IF a$="3" THEN 2250                       [1013]
1900 IF a$="4" THEN 5450                       [766]
1910 GOTO 1840                                 [395]
1920 PRINT#1,"Name des Textes CWLERN*.555      [3681]
  '* A - Z ":CLS:CAT:LOCATE#1,40,1:INPUT#1
,txt$
1930 file$="cwlern"+txt$+".555"                [1228]
1940 PRINT#1,"Text bis 840 Zeichen":LOCATE     [5729]
#1,25,1:PRINT#1," ("file$")"
1950 CLS:LOCATE 1,18:PRINT"Nach Texteingab     [6567]
```

```
e Menue 'f7 oder 'f9  waehlen !!!"
1960 a1=1:a2=1:f=0                                  [606]
1970 LOCATE#1,44,1:PRINT#1,"Z=";a2;:PRINT#          [2850]
1,"B=";a1
1980 LOCATE a1,a2:PRINT"."                          [1990]
1990 a$=INKEY$:IF a$="" THEN 1990                   [1538]
2000 a$=UPPER$(a$)                                  [533]
2010 IF a$="*" THEN f=2:GOTO 1990                   [1651]
2020 IF a$="7" AND f=2 THEN f=0:zuf(a2,a1)          [2855]
=42:GOTO 2150
2030 IF a$="9" AND f=2 THEN f=0:ERASE zuf:          [3067]
PRINT#1,CHR$(24):GOTO 3210
2040 IF NOT INKEY(79) THEN a1=a1-1:LOCATE           [3532]
a1+1,a2:PRINT" ";:a$=" ":f=1
2050 IF a1<1 THEN a1=56:a2=a2-1                     [1283]
2060 IF a2<1 THEN a2=1:a1=1                         [1023]
2070 zuf(a2,a1)=ASC(a$)                             [1711]
2080 IF zuf(a2,a1)=127 THEN zuf(a2,a1)=32:          [3360]
LOCATE a1+1,a2:PRINT " ":a$=" "
2090 LOCATE a1,a2:PRINT a$;                         [1626]
2100 IF f=1 THEN f=0:GOTO 1970                      [1943]
2110 a1=a1+1                                        [536]
2120 IF a1=57 THEN a1=1:a2=a2+1                     [958]
2130 IF a2=16 THEN a2=15:a1=56                      [755]
2140 GOTO 1970                                      [337]
2150 'Speichern                                     [428]
2160 OPENOUT file$                                  [781]
2170 a1=1:a2=1                                      [444]
2180 PRINT#9,zuf(a2,a1)                             [1122]
2190 a1=a1+1                                        [536]
2200 IF a1=57 THEN a1=1:a2=a2+1                     [958]
2210 IF a2=16 THEN 2230                             [614]
2220 GOTO 2180                                      [343]
2230 CLOSEOUT                                       [902]
2240 GOTO 2390                                      [397]
2250 'Einlesen                                      [1453]
2260 CLS:CAT:INPUT#1,"filename CWLERN*.555          [5534]
 'Buchstabe: A-Z ";txt$
2270 file$="CWLERN"+txt$+".555"                     [1657]
2280 OPENIN file$                                   [1062]
2290 a1=1:a2=1                                      [444]
2300 INPUT#9,zuf(a2,a1)                             [855]
2310 IF zuf(a2,a1)=42 THEN 2360                     [1508]
2320 a1=a1+1                                        [536]
2330 IF a1=57 THEN a1=1:a2=a2+1                     [958]
2340 IF a2=16 THEN 2360                             [461]
2350 GOTO 2300                                      [351]
2360 CLOSEIN                                        [752]
2370 f7=1                                           [326]
2380 CLS                                            [91]
2390 'Ausgabe des Textes                            [2411]
2400 IF f7=0 THEN 1780                              [941]
2410 LOCATE 1,18:PRINT"Text im CW - Code a          [4324]
usgeben (j/n)"
2420 a$=INKEY$:IF a$="" THEN 2420                   [1516]
2430 a$=UPPER$(a$)                                  [533]
2440 IF a$="N" THEN 1780                            [883]
2450 IF a$="*" THEN f=1:CLS:GOTO 2420               [1757]
2460 IF a$="8" THEN f=0:n=8:GOTO 2570               [1593]
2470 IF a$="J" THEN 2490                            [789]
2480 GOTO 2420                                      [343]
2490 LOCATE 1,18:PRINT"Soll der Code  best          [3561]
aetigt werden (j/n)?"
2500 a$=INKEY$:IF a$="" THEN 2500                   [1516]
2510 a$=UPPER$(a$)                                  [533]
2520 best=0:CLS#1:LOCATE#4,1,1:PRINT#4,CHR          [2864]
$(127)
2530 IF a$="J" THEN best=1:feh=0                    [1306]
2540 IF best=1 THEN CLS:PRINT"Bei diesem M          [16431]
enuepunkt wird nach jeder Ausgabe des Code
s":PRINT"auf die richtige Bestaetigung gew
artet.":PRINT"Druecken Sie also die entspr
echene  Taste  zu dem Code."
2550 IF best=1 THEN PRINT"Dieser wird ansc          [10801]
hliessend geprueft und die  Anzahl  der":P
RINT"Fehler gezaehlt!":PRINT:PRINT"<ENTER>
":CALL &BB06
2560 b$=" ":GOSUB 2850                              [1072]
2570 aut1=1:a1=1:a2=1:a3=1:ff=0                     [1562]
2580 IF n=8 THEN PRINT#8:PRINT#8,"(c)DK9PM          [3893]
 HJZ   ";dat$;"   Morsetext :  ";
2590 PRINT#n,"V";:l=86:GOSUB 3040:PRINT#n,          [7298]
"V";:l=86:GOSUB 3040:PRINT#n,"V";:l=86:GOS
UB 3040:PRINT#n,"KA":l=93:GOSUB 3040:PRINT
#n,CHR$(13)
2600 l=zuf(a2,a1)                                   [1013]
2610 IF n=8 THEN 2640                               [738]
2620 IF l=32 THEN SOUND 1,0,2*dauer,0:b$=b          [4009]
$+"  ":PRINT" ";:GOTO 2730
2630 GOSUB 3040                                     [953]
2640 IF best=0 THEN PRINT#n,CHR$(zuf(a2,a1          [2688]
));
2650 best$="g":IF best=1 THEN best$=INKEY$          [4038]
:IF best$="" THEN 2650
2660 IF best=1 THEN FOR a=1 TO 50:NEXT              [2246]
2670 IF l=42 AND best=1 THEN GOSUB 2860:GO          [2420]
SUB 2840:GOTO 2380
2680 IF l=42 THEN GOSUB 2860:GOSUB 2830:l=          [5374]
42:PRINT#n:PRINT#n:n=0:LOCATE 1,18:INPUT"C
W Code nochmal (j/n)";a$
2690 a$=UPPER$(a$)                                  [533]
2700 IF l=42 AND a$="J" THEN CLS:n=0:GOTO           [2058]
2490
2710 IF l=42 AND a$="N" THEN n=0:aut1=0:GO          [2850]
TO 1780
2720 IF NOT INKEY(79) AND best=1 THEN GOSU          [6997]
B 2860:aut1=0:best=0:GOSUB 2840:GOTO 2380
2730 IF NOT INKEY(79) AND best=0 THEN GOSU          [4647]
B 2860:aut1=0:best=0:GOSUB 2830:GOTO 2380
2740 IF l=32 THEN 2790                              [1347]
2750 IF best=1 THEN LOCATE 1,18                     [1813]
2760 IF best=1 AND best$="" THEN 2750               [1788]
2770 IF best=1 THEN best$=UPPER$(best$):IF          [5619]
 CHR$(zuf(a2,a1))<>best$ THEN feh=feh+1
2780 a3=a3+1                                        [697]
2790 a1=a1+1                                        [536]
2800 IF a1=57 THEN a1=1:a2=a2+1                     [958]
2810 IF a2=16 THEN 1800                             [636]
2820 GOTO 2600                                      [339]
2830 IF n=0 THEN PRINT#1,"Es waren";a3-1;"          [6250]
Zeichen in";FNUhr;" Sekunden";:RETURN
2840 IF n=0 THEN PRINT#1,"Es waren";a3-1;"          [7327]
Zeichen und ";feh;"Fehler in";FNUhr;" Seku
nden";:RETURN
2850 t=TIME/300:DEF FNUhr=INT(TIME/300-t):          [2674]
CLS:RETURN
2860 PRINT" AR*";::l=92:GOSUB 3040:RETURN           [1640]
2870 aut=1                                          [314]
2880 a$=INKEY$:IF a$="" THEN 2880                   [1472]
2890 IF a$="*" THEN f=1                             [1257]
2900 IF a$="\" THEN ff=1:GOTO 2880                  [3008]
2910 IF ff=1 THEN 2960                              [583]
2920 a$=UPPER$(a$):PRINT a$;:LOCATE#4,1,1:          [5869]
PRINT#4,CHR$(24)a$CHR$(24);:l=ASC(a$)
2930 IF l=32 THEN SOUND 1,0,2*dauer,0:b$=b          [4513]
$+"  ":GOTO 2880
2940 IF NOT INKEY(79) THEN CLEAR INPUT:bes          [2652]
t=0:GOTO 2380
2950 IF f=1 AND l=57 THEN PRINT#1,CHR$(24)          [2540]
:ERASE zuf:GOTO 3210
2960 IF a$="s" AND ff=1 THEN l=91:ff=0              [2029]
2970 IF a$="a" AND ff=1 THEN l=92:ff=0              [2708]
```

```
2980 IF a$="k" AND ff=1 THEN l=93:ff=0      [2744]
2990 IF a$="e" AND ff=1 THEN l=94:ff=0      [2106]
3000 IF a$="v" AND ff=1 THEN l=95:ff=0      [1433]
3010 IF a$="i" AND ff=1 THEN l=96:ff=0      [2111]
3020 IF l=42 AND f=1 THEN a$=RIGHT$((a$),1 [2297]
):GOTO 2880
3030 IF l>96 OR l<38 THEN 2880              [2300]
3040 IF n=8 THEN RETURN                     [701]
3050 FOR a=1 TO 9                           [767]
3060 IF m(l,a)=5 THEN a=9:lae=3:GOTO 3110   [2462]
3070 IF m(l,a)=1 THEN b$=b$+CHR$(144)       [2124]
3080 IF m(l,a)=3 THEN b$=b$+"-"             [1765]
3090 IF m(l,a)>0 THEN ton1=ton:lae=m(l,a)   [2150]
3100 SOUND 1,ton1,lae*dauer,15              [2169]
3110 SOUND 1,0,dauer,0                      [1409]
3120 NEXT                                   [350]
3130 IF LEN(b$) >55 THEN b$=RIGHT$((b$),55 [2598]
)
3140 PRINT#1,b$                             [424]
3150 SOUND 1,0,3*dauer,0                    [1840]
3160 b$=b$+" "                              [416]
3170 IF aut1=1 THEN RETURN                  [1402]
3180 GOTO 2880                              [315]
3190 END                                    [110]
3200 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq [2488]
cqcq
3210 'cqcqcq        Einstellung         cq [1016]
cqcq
3220 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq [2488]
cqcq
3230 CLS:CLS#1:LOCATE#4,1,1:PRINT#4,CHR$(1 [2029]
27):aut1=1:wpm=0:ton=0
3240 CLEAR INPUT:INPUT#1,"Programm beenden [2174]
 (j/n)";end1$
3250 IF end1$="j" OR end1$="J" THEN END     [2151]
3260 PRINT"Bei <ENTER> Uebernahme der Stan [3909]
dardwerte"
3270 PRINT:CLEAR INPUT:PRINT#1,"BPM:20,30, [7063]
40,50,60,";:PRINT#1,CHR$(24)"75"CHR$(24);:
PRINT#1,",90,100,125,150,165,190,200 ";
3280 INPUT#1,wpm                            [373]
3290 IF wpm=20 THEN dauer=23                [1224]
3300 IF wpm=30 THEN dauer=16                [1359]
3310 IF wpm=40 THEN dauer=12                [1068]
3320 IF wpm=50 THEN dauer=9                 [1144]
3330 IF wpm=60 THEN dauer=8                 [1578]
3340 IF wpm=75 THEN dauer=6.3               [1444]
3350 IF wpm=90 THEN dauer=5.3               [1607]
3360 IF wpm=100 THEN dauer=4.5              [1687]
3370 IF wpm=125 THEN dauer=3.5              [1227]
3380 IF wpm=150 THEN dauer=3.3              [1181]
3390 IF wpm=165 THEN dauer=2.5              [1397]
3400 IF wpm=190 THEN dauer=2                [1150]
3410 IF wpm=200 THEN dauer=1                [1489]
3420 IF wpm=0 THEN wpm=75:dauer=6.3         [2486]
3430 CLEAR INPUT                            [184]
3440 PRINT:PRINT#1,"Ton : ";:PRINT#1,CHR$( [5652]
24)"800"CHR$(24);:PRINT#1,",850,950,1000,1
050 Hz";:INPUT#1,hz
3450 IF hz=800 THEN ton=78                  [837]
3460 IF hz=850 THEN ton=74                  [1017]
3470 IF hz=900 THEN ton=69                  [1838]
3480 IF hz=950 THEN ton=65                  [884]
3490 IF hz=1000 THEN ton=62                 [737]
3500 IF hz=0 THEN hz=800:ton=78             [765]
3510 a$=" ":f=0:CLS                         [1083]
3520 LOCATE#7,7,17:PRINT#7,CHR$(24)"ca."wp [3228]
m;:LOCATE#7,7,18:PRINT#7,"Hz "hz;CHR$(24)
3530 CLS#1:PRINT#1,CHR$(24)" Waehlen Sie!! [3282]
! "CHR$(24)
3540 PRINT:PRINT;CHR$(24)"- 1 - "CHR$(24)" [2153]
 Morseband"
3550 PRINT:PRINT;CHR$(24)"- 2 - "CHR$(24)" [5002]
 Buchstabe und Morseanzeige"
3560 PRINT:PRINT;CHR$(24)"- 3 - "CHR$(24)" [4823]
 Fuenfer Gruppen (alle Zeichen)"
3570 PRINT:PRINT;CHR$(24)"- 4 - "CHR$(24)" [5471]
 Fuenfer Gruppen (nach eigener Wahl)"
3580 PRINT:PRINT;CHR$(24)"- 5 - "CHR$(24)" [5472]
 Morsegenerator Junkerstaste"
3590 IF NOT INKEY(64) THEN CLEAR INPUT:GOT [2163]
O 1750
3600 IF NOT INKEY(65) THEN CLEAR INPUT:GOT [2291]
O 1450
3610 IF NOT INKEY(57) THEN CLEAR INPUT:GOT [2645]
O 3660
3620 IF NOT INKEY(56) THEN CLEAR INPUT:GOT [2091]
O 4280
3630 IF NOT INKEY(49) THEN CLEAR INPUT:GOT [3046]
O 5830
3640 GOTO 3590                              [491]
3650 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq [2488]
cqcq
3660 'cqcqcq  5er Gruppen alle Zeichen  cq [3653]
cqcq
3670 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq [2488]
cqcq
3680 CLS                                    [91]
3690 IF cw=1 THEN ERASE zuf                 [882]
3700 DIM zuf(676):cw=1                      [1002]
3710 PRINT#1,"5er Code Laden 'f7   oder     [2601]
Erzeugen <ENTER>"
3720 a$=INKEY$:IF a$="" THEN 3720           [1480]
3730 IF a$="*" THEN f=2:GOTO 3710           [1565]
3740 IF a$="7" AND f=2 THEN n=9:GOTO 4190   [2214]
3750 IF a$="9" AND f=2 THEN ERASE zuf:PRIN [3352]
T#1,CHR$(24):GOTO 3210
3760 CLS                                    [91]
3770 PRINT"WARTEN : Es werden 5er Gruppen   [6368]
von A-Z,0-1 und Satzzei -         chen erz
eugt!!"
3780 mi=38:ma=90                            [543]
3790 ii=675                                 [216]
3800 FOR i=1 TO ii                          [562]
3810 z=INT((1+ma-mi)*RND(1)+mi)             [1895]
3820 IF z=38 OR z=39 OR z=42 OR z=60 OR z= [3862]
62 OR z=64 THEN 3810
3830 zuf(i)=z:LOCATE 1,1:PRINT#1,zuf(i):NE [3628]
XT
3840 CLS:PRINT#1,"Drucken 'f8  oder  Abspe [10857]
ichern 'f7":PRINT:PRINT"Bei <ENTER> wird d
er CW Code ausgegeben!":PRINT:PRINT"Mit <D
EL>   Programm unterbrechen"
3850 n=0:CLEAR INPUT                        [990]
3860 a$=INKEY$:IF a$="" THEN 3860           [1340]
3870 n=0:PRINT#1,CHR$(24):b$=""             [1235]
3880 IF a$="*" THEN f=1:GOTO 3860           [1796]
3890 IF a$="8" AND f=1 THEN n=8             [2012]
3900 IF a$="7" AND f=1 THEN n=9:PRINT#1,CH [3228]
R$(24):GOTO 4090
3910 IF a$="9" AND f=1 THEN aut1=0:cw=0:ER [3448]
ASE zuf:PRINT#1,CHR$(24):GOTO 3210
3920 CLS#1                                  [373]
3930 PRINT#1,"Alle Zeichen gemischt !!!"    [3139]
3940 CLS:b=1:c=1                            [524]
3950 FOR a=1 TO 200:NEXT                    [1170]
3960 IF n=8 THEN PRINT#n:PRINT#n,"(c)DK9PM [3765]
 HJZ     ";dat$;"   Morsetext :   ";
3970 PRINT#n,"V";:l=86:GOSUB 3040:PRINT#n, [7298]
"V";:l=86:GOSUB 3040:PRINT#n,"V";:l=86:GOS
UB 3040:PRINT#n,"KA":l=93:GOSUB 3040:PRINT
#n,CHR$(13)
```

```
3980 FOR i=1 TO 675                                    [603]
3990 IF NOT INKEY(79) THEN CLEAR INPUT:aut             [4653]
1=0:cw=0:ERASE zuf:PRINT#1,CHR$(24):GOTO 3
210
4000 IF b=6 THEN b=1:PRINT#n," ";:b$=b$+"              [3220]
 ":SOUND 1,0,3*dauer,0
4010 PRINT#n,STRING$(1,zuf(i));                        [2587]
4020 IF c=45 THEN b=0:c=0:PRINT#n,CHR$(13)             [2282]
4030 l=zuf(i):GOSUB 3040                               [1813]
4040 b=b+1                                             [569]
4050 c=c+1                                             [701]
4060 NEXT                                              [350]
4070 PRINT#n,CHR$(13):PRINT#n,"AR";:PRINT"             [6794]
       <ENTER>":l=92:GOSUB 3040:PRINT#n,C
HR$(13):CALL &BB06:PRINT#1,CHR$(24)
4080 CLS:n=0:GOTO 3840                                 [955]
4090 'Auf Diskette speichern                           [1063]
4100 CLS:PRINT"WARTEN"                                 [681]
4110 OPENOUT "cwlern.5                                 [1144]
4120 FOR i=1 TO 675                                    [603]
4130 PRINT#9,zuf(i)                                    [1079]
4140 NEXT                                              [350]
4150 zuf(676)=42                                       [821]
4160 PRINT#9,zuf(676)                                  [1084]
4170 CLOSEOUT                                          [902]
4180 CLS:n=0:f=0:GOTO 3840                             [1042]
4190 'Von Diskette lesen                               [1144]
4200 PRINT"WARTEN"                                     [1300]
4210 OPENIN "cwlern.5                                  [965]
4220 FOR i=1 TO 676                                    [598]
4230 INPUT#9,zuf(i)                                    [808]
4240 NEXT                                              [350]
4250 CLOSEIN                                           [752]
4260 CLS:n=0:f=0:GOTO 3840                             [1042]
4270 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq               [2488]
cqcq
4280 'cqcqcq   Fuenfer Code nach Wahl   cq             [2595]
cqcq
4290 'cqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcqcq               [2488]
cqcq
4300 CLS:PRINT#1,CHR$(24)                              [648]
4310 IF cw=1 THEN ERASE zuf:ERASE a                    [1338]
4320 DIM zuf(676):DIM a(9,6):cw=1                      [1774]
4330 CLS                                               [91]
4340 PRINT#1,"5er Code Laden 'f7   oder                [2601]
Erzeugen <ENTER>"
4350 CLEAR INPUT                                       [184]
4360 a$=INKEY$:IF a$="" THEN 4360                      [1512]
4370 IF a$="*" THEN f=2:GOTO 4360                      [1658]
4380 IF a$="7" AND f=2 THEN n=9:GOTO 5130              [2138]
4390 IF a$="9" AND f=2 THEN aut1=0:cw=0:ER             [5078]
ASE zuf,a:PRINT#1,CHR$(24):GOTO 3210
4400 'Vorgeschlagene 5er Gruppen 1-9                   [1299]
4410 a(1,1)=65:a(1,2)=70:a(1,3)=75:a(1,4)=             [3253]
80:a(1,5)=85:a(1,6)=90
4420 a(2,1)=66:a(2,2)=71:a(2,3)=76:a(2,4)=             [2527]
81:a(2,5)=86:a(2,6)=69
4430 a(3,1)=67:a(3,2)=72:a(3,3)=77:a(3,4)=             [3887]
82:a(3,5)=87:a(3,6)=68
4440 a(4,1)=68:a(4,2)=74:a(4,3)=78:a(4,4)=             [2982]
83:a(4,5)=88:a(4,6)=65
4450 a(5,1)=69:a(5,2)=73:a(5,3)=79:a(5,4)=             [2763]
84:a(5,5)=89:a(5,6)=66
4460 a(6,1)=49:a(6,2)=51:a(6,3)=53:a(6,4)=             [2645]
55:a(6,5)=57:a(6,6)=50
4470 a(7,1)=50:a(7,2)=52:a(7,3)=54:a(7,4)=             [3437]
56:a(7,5)=48:a(7,6)=49
4480 a(8,1)=46:a(8,2)=44:a(8,3)=63:a(8,4)=             [3370]
61:a(8,5)=58:a(8,6)=63
4490 a(9,1)=40:a(9,2)=41:a(9,3)=43:a(9,4)=             [3285]
47:a(9,5)=45:a(9,6)=59
```

```
4500 FOR a1=1 TO 9                                     [1085]
4510 PRINT;a1"›  ";                                    [551]
4520 FOR a2=1 TO 6                                     [1192]
4530 PRINT CHR$(a(a1,a2))" ";                          [1887]
4540 NEXT a2                                           [460]
4550 PRINT" ‹"                                         [650]
4560 PRINT                                             [361]
4570 NEXT a1                                           [461]
4580 PRINT" 0 ›  vom Benutzer zu definiere             [3989]
n ‹  ";
4590 PRINT:PRINT:PRINT;" "CHR$(147);:INPUT             [5109]
" ›  Waehlen Sie 0 bis 9 ..... ";a1
4600 IF a1>0 THEN 4700                                 [1254]
4610 CLS:PRINT"6 Buchstaben,Zeichen oder S             [5465]
atzzeichen eingeben :":PRINT
4620 a2=1                                              [426]
4630 a$=INKEY$:IF a$="" THEN 4630                      [1546]
4640 IF a$="*" THEN f=1:GOTO 4630                      [1810]
4650 IF a$="9" THEN f=0:ERASE zuf:PRINT#1,             [5901]
CHR$(24):GOTO 3210
4660 a$=UPPER$(a$):PRINT#0,a$;                         [1539]
4670 a(0,a2)=ASC(a$)                                   [765]
4680 a2=a2+1                                           [709]
4690 IF a2<7 THEN 4630                                 [974]
4700 CLS:PRINT"WARTEN : Es werden 5er Grup             [6436]
pen mit dem Zufallsgenerator          nach
 Ihrer Wahl erzeugt !!!"
4710 mi=1:ma=6                                         [669]
4720 ii=675                                            [216]
4730 FOR i=1 TO ii                                     [562]
4740 z=INT((1+ma-mi)*RND(1)+mi)                        [1895]
4750 IF NOT INKEY(79) THEN CLEAR INPUT:PRI             [3487]
NT#1,CHR$(24):GOTO 4300
4760 zuf(i)=a(a1,z):LOCATE 1,1:PRINT#1,zuf             [3865]
(i):NEXT
4770 CLS:CLS#1:PRINT#1,a1;" › "CHR$(a(a1,1             [5674]
))CHR$(a(a1,2))CHR$(a(a1,3))CHR$(a(a1,4))C
HR$(a(a1,5))CHR$(a(a1,6))" ‹"
4780 CLS:PRINT#1,"Drucken 'f8  oder  Abspe             [10857]
ichern 'f7":PRINT:PRINT"Bei <ENTER> wird d
er CW Code ausgegeben!":PRINT:PRINT"Mit <D
EL>   Programm unterbrechen"
4790 CLEAR INPUT:n=0:b$=""                             [509]
4800 a$=INKEY$:IF a$="" THEN 4800                      [1536]
4810 IF a$="*" THEN f=1:GOTO 4800                      [1689]
4820 IF a$="8" AND f=1 THEN n=8                        [2012]
4830 IF a$="7" AND f=1 THEN n=9:GOTO 5010              [2312]
4840 IF a$="9" AND f=1 THEN aut1=0:cw=0:ER             [3199]
ASE zuf:PRINT#1,CHR$(24):ERASE a:GOTO 3210
4850 CLS:b=1:c=1                                       [524]
4860 FOR a=1 TO 200:NEXT                               [1170]
4870 PRINT;CHR$(24)" › "CHR$(a(a1,1))CHR$(             [5383]
a(a1,2))CHR$(a(a1,3))CHR$(a(a1,4))CHR$(a(a
1,5))CHR$(a(a1,6))" ‹ "CHR$(24)" ";
4880 IF n=8 THEN PRINT#n:PRINT#n,"(c)DK9PM             [3765]
 HJZ     ";dat$;"   Morsetext :   ";
4890 PRINT#n,"V";:l=86:GOSUB 3040:PRINT#n,             [7298]
"V";:l=86:GOSUB 3040:PRINT#n,"V";:l=86:GOS
UB 3040:PRINT#n,"KA":l=93:GOSUB 3040:PRINT
#n,CHR$(13)
4900 FOR i=1 TO 675                                    [603]
4910 IF NOT INKEY(79) THEN CLEAR INPUT:PRI             [3487]
NT#1,CHR$(24):GOTO 4300
4920 IF b=6 THEN b=1:PRINT#n," ";:b$=b$+"              [3220]
 ":SOUND 1,0,3*dauer,0
4930 PRINT#n,STRING$(1,zuf(i));                        [2587]
4940 IF c=45 THEN b=0:c=0:PRINT#n,CHR$(13)             [2282]
4950 l=zuf(i):GOSUB 3040                               [1813]
4960 b=b+1                                             [569]
4970 c=c+1                                             [701]
4980 NEXT                                              [350]
```

```
4990 PRINT#n,CHR$(13):PRINT#n,"AR";:PRINT" [6794]
      <ENTER>":l=92:GOSUB 3040:PRINT#n,C
HR$(13):CALL &BB06:PRINT#1,CHR$(24)
5000 CLS:n=0:GOTO 4780                    [946]
5010 'Diskette speichern                  [1739]
5020 CLS:PRINT"WARTEN"                    [681]
5030 OPENOUT "cwlern.55                   [1231]
5040 PRINT#9,a1                           [917]
5050 PRINT#9,a(a1,1),a(a1,2),a(a1,3),a(a1, [2162]
4),a(a1,5),a(a1,6)
5060 FOR i=1 TO 675                       [603]
5070 PRINT#9,zuf(i)                       [1079]
5080 NEXT                                 [350]
5090 zuf(676)=42                          [821]
5100 PRINT#9,zuf(676)                     [1084]
5110 CLOSEOUT                             [902]
5120 CLS:n=0:f=0:GOTO 4780                [823]
5130 ' Diskette lesen                     [1141]
5140 CLS:PRINT"WARTEN"                    [681]
5150 OPENIN "cwlern.55                    [1164]
5160 INPUT#9,a1                           [566]
5170 INPUT#9,a(a1,1),a(a1,2),a(a1,3),a(a1, [3006]
4),a(a1,5),a(a1,6)
5180 FOR i=1 TO 676                       [598]
5190 INPUT#9,zuf(i)                       [808]
5200 NEXT                                 [350]
5210 CLOSEIN                              [752]
5220 CLS:n=0:f=0:GOTO 4780                [823]
5230 'Uhr                                 [500]
5240 INPUT#0,"Datum: xx.xx.xx ";dat$      [2029]
5250 IF dat$="" THEN dat$=dat1$           [2428]
5260 IF LEN(dat$)>8 OR LEN(dat$)<8 THEN 52 [2303]
40
5270 PRINT:INPUT#0,"Uhrzeit in UT/GMT (Stu [5633]
nde,Minute)";stunde,minute
5280 minute=minute-1                      [719]
5290 CLS                                  [91]
5300 PRINT#7,CHR$(24)                     [812]
5310 minute=minute+1                      [676]
5320 IF minute >59 THEN minute=0:stunde=st [2488]
unde+1
5330 IF stunde >23 THEN stunde=0          [825]
5340 LOCATE#7,1,25:PRINT#7,CHR$(91)dat$CHR [3424]
$(93);
5350 IF stunde<10 THEN PRINT #7,USING " 0# [4076]
";stunde;:GOTO 5370
5360 PRINT #7,USING " ##";stunde;         [1877]
5370 IF minute<10 THEN PRINT #7,":";USING [2487]
"0# ";minute;::GOTO 5420
5380 PRINT#7,":";                         [619]
5390 PRINT #7,USING "## ";minute;         [678]
5400 stunde$=STR$(stunde)                 [1369]
5410 minute$=STR$(minute)                 [1229]
5420 PRINT#7,CHR$(24);                    [873]
5430 RETURN                               [555]
5440 'Files umbenennen zum bestaetigen    [1901]
5450 CLS:CLS#1:file1$="":file2$=""        [69]
5460 PRINT"Dieser Menuepunkt dient zum [4334]
 umbenennen von Files"
5470 PRINT"vom Typ (cwlern.5) (cwlern.55) [10535]
  in (cwlern.555) deren":PRINT"Dateninhal
t mit dem Zufallgenerator erzeugt wurden.!
!!"
5480 PRINT:PRINT"Diese Umbenennung soll da [6173]
s Einlesen unter Menuepunkt"
5490 PRINT"Festtext ermoeglichen um sich s [4428]
elbst zu testen."
5500 PRINT:PRINT"<ENTER>":CALL &BB06      [1823]
5510 CLS:PRINT"-1- cwlern.5 (alle Zeichen) [3141]
  umbenennen"
5520 PRINT:PRINT"-2- cwlern.55 (6 Zeichen) [2973]
  umbennen"
5530 PRINT:PRINT"-3- file loeschen"       [1779]
5540 PRINT:PRINT"-4- Morseband zurueck"   [2089]
5550 a$=INKEY$:IF a$="" THEN 5550         [1530]
5560 IF a$="1" THEN file1$="cwlern.5"     [2172]
5570 IF a$="2" THEN file1$="cwlern.55"    [2146]
5580 IF a$="3" THEN 5720                  [1117]
5590 IF a$="4" THEN 1780                  [565]
5600 LOCATE#1,24,1:PRINT#1,file1$         [1231]
5610 CLS:CAT                              [140]
5620 LOCATE 1,18:INPUT"Neuer Filename  'cw [5248]
lern*.555   * a bis z ";file2$
5630 file2$="cwlern"+file2$+".555":LOCATE [3157]
#1,1,1:PRINT#1,"Iren,'"file2$"',1"
5640 LOCATE #1,36,1:PRINT#1,"OK? (j/n)"   [2129]
5650 a$=INKEY$:IF a$="" THEN 5650         [1502]
5660 a$=UPPER$(a$)                        [533]
5670 IF a$="J" THEN 5700                  [929]
5680 IF a$="N" THEN 5450                  [835]
5690 GOTO 5650                            [409]
5700 |REN,file2$,file1$                   [1766]
5710 GOTO 5450                            [353]
5720 CLS:CAT                              [140]
5730 LOCATE 1,18:INPUT"Filename";file$    [1504]
5740 IF file$="" THEN 5450                [747]
5750 LOCATE 30,18:PRINT"Wirklich loeschen [2663]
 (j/n)"
5760 a$=INKEY$:IF a$="" THEN 5760         [1456]
5770 a$=UPPER$(a$)                        [533]
5780 IF a$="J" THEN |ERA,file$:GOTO 5450  [1995]
5790 GOTO 5450                            [353]
5800 'Fehlerabfangroutine                 [1642]
5810 PRINT:PRINT"Programm wird neu gestart [4175]
et  (ENTER)":CALL &BB06
5820 CLS:RUN                              [342]
5830 'Morsetaste Junkers                  [1254]
5840 CLS#1:PRINT#1,"'Morsegenerator zur B [6006]
enutzung einer Junkers-Taste'"
5850 CLS:PRINT"Taste an Joystick-Buchse 6 [6850]
und 8 anschliessen !!!":PRINT:PRINT"(ENTER
)":CALL &BB06:CLS
5860 CLS                                  [91]
5870 PRINT"Tasthaltung!Gewoehnen Sie sich [5075]
von Anfang an eine rich-"
5880 PRINT"Tasthaltung an.Dazu nehmen Sie [6538]
 den  Tastknopf zwischen"
5890 PRINT"Daumen (links vorne etwas un [5736]
terhalb ),  Zeigefinger"
5900 PRINT"(links hinten) und Mittelfinger [5655]
 (rechts hinten)."
5910 PRINT                                [361]
5920 PRINT"Jede unnatuerliche Haltung und [5804]
  jegliche Verkrampfung"
5930 PRINT"ist zu vermeiden.Ellenbogen auf [8621]
 den  Tisch auflegen und"
5940 PRINT"im Rhytmus der Zeichen schwinge [6223]
n.Wobei der Daumen alle"
5950 PRINT"Bewegungen mitzumachen hat."   [3860]
5960 PRINT:PRINT"Weitere Informatioen gibt [6053]
 Ihnen sicherlich jeden Funk -"
5970 PRINT"amateur !":PRINT:PRINT"(ENTER)" [4165]
:CALL &BB06:CLS
5980 SPEED KEY 1,2                        [1274]
5990 IF JOY(0)=16 THEN SOUND 1,69,1,15    [1644]
6000 IF JOY(0)<>16 THEN SOUND 1,0,1,0     [1528]
6010 IF NOT INKEY(79) THEN SPEED KEY 30,2: [1904]
GOTO 3510
6020 GOTO 5990                            [331]
```

# Tilgungsplan

**Noch nie war es so leicht, von der Bank oder sonstigen Geldinstituten Geld zu bekommen, selbst wenn das Konto nicht die nötige Deckung aufweist. Kredite werden häufig in Anspruch genommen, doch wie sieht es mit der monatlichen Belastung aus? Damit Sie Ihre Kosten bis ins letzte Kalkül übersichtlich berechnen können, folgt nun unser CPC-Tilgungsplan.**

Nach dem Starten des Programmes erscheint das Hauptmenü. Hier können Sie zunächst Adresseinträge vornehmen. Sie haben anschließend die Wahl zwischen einer Berechnung der gesamten Darlehenskosten oder der Erstellung eines kompletten Tilgungsplanes. Dazu geben Sie nur Ihre Schuldsumme, den Jahreszins sowie die gewünschte Zahlungsweise (monatlich oder vierteljährlich) an. In kürzester Zeit erhalten Sie den kompletten Tilgungsplan mit Ihren regelmäßigen Belastungen. Natürlich können Sie sich das Ganze auch auf Drucker ausgeben lassen und archivieren. Ansonsten ist das Programm selbsterklärend. (G.Renius/sr)

## für 464-664-6128

```
10 '********************************          [1285]
20 '**        TILGUGSPLAN           **        [1091]
30 '**  VON  G.RENIUS    1987      **        [1654]
40 '********************************          [1285]
50 CLS:MODE 2:BORDER 12                       [1009]
60 '                                          [117]
70 '****HAUPTMENUE****                        [825]
80 '                                          [117]
90 LOCATE 20,8:PRINT"********************** [2622]
**********************"
100 LOCATE 20,9:PRINT"***     TILGUNG VON G [3651]
.RENIUS  1987     ***"
110 LOCATE 20,10:PRINT"********************** [2698]
**********************"
120 LOCATE 20,11:PRINT"***                    [2734]
                  ***"
130 LOCATE 20,12:PRINT"***  ADRESSE EINGEB [3498]
EN          =a    ***"
140 LOCATE 20,13:PRINT"***                    [1903]
                  ***"
150 LOCATE 20,14:PRINT"***  TILGUGS-PLAN E [3320]
RSTELLEN    =b    ***"
160 LOCATE 20,15:PRINT"***                    [2712]
                  ***"
170 LOCATE 20,16:PRINT"***  DARLEHNS-KOSTE [3757]
N BERECHN.  =c    ***"
180 LOCATE 20,17:PRINT"***                    [2789]
                  ***"
190 LOCATE 20,18:PRINT"***  PROGRAMM-ENDE  [2148]
            =x    ***"
200 LOCATE 20,19:PRINT"***                    [2432]
                  ***"
210 LOCATE 20,20:PRINT"********************** [3370]
**********************"
220 LOCATE 39,23:PRINT"***"                   [1778]
230 LOCATE 35,24:PRINT">>>      <<<"          [1908]
240 LOCATE 39,25:PRINT"***"                   [1378]
250 LOCATE 39,24:INPUT;go$                    [1654]
260 IF go$="a" THEN CLS:nr=0:zs=0:sua=0:GO [2928]
TO 320
270 IF go$="b" THEN nr=0:zs=0:sua=0:CLS:GO [2471]
TO 530
280 IF go$="c" THEN CLS:zs=0:sua=0:CLS:GOT [2438]
O 2020
290 IF go$="x" THEN CLS:GOTO 2500             [1120]
300 IF go$<>"a/b/c/x" OR go$="" THEN GOTO  [2700]
250
310 '                                         [117]
320 '**** ADRESSE EINGEBEN ****               [1220]
330 '                                         [117]
340 LOCATE 20,7:PRINT"Adresse eingeben ?   [4969]
(Weiter mit ENTER)"
350 LOCATE 10,9:PRINT"Anrede        :"        [1927]
360 LOCATE 30,9:INPUT;an$:IF an$="" THEN G [2619]
OTO 50
370 LOCATE 10,11:PRINT"Name         :"        [1031]
380 LOCATE 30,11:INPUT; nam$                  [1369]
390 LOCATE 10,13:PRINT"Str.-Platz   :"        [2436]
400 LOCATE 30,13:INPUT; st$                   [1780]
410 LOCATE 10,15:PRINT"Haus- Nr. :"           [2200]
420 LOCATE 30,15:INPUT;hn                     [1572]
430 LOCATE 10,17:PRINT"Postleitzahl :"        [3521]
440 LOCATE 30,17:INPUT;plz$                   [915]
450 LOCATE 10,19:PRINT"Wohnort :"             [2298]
460 LOCATE 30,19:INPUT; wo$                   [1526]
470 LOCATE 10,22:PRINT"Eingabe richtig (j/ [3636]
n) :"
480 LOCATE 35,22:INPUT;eig$                   [1406]
490 IF eig$="j" THEN GOTO 550                 [842]
500 IF eig$="n" THEN CLS:GOTO 320             [2136]
510 IF eig$<>"j/n" OR eig$="" THEN GOTO 48 [1685]
0
520 '                                         [117]
530 '**** DRUCKER ABFRAGE ****                [1432]
540 '                                         [117]
550 LOCATE 10,24:PRINT"Tilgugsplan drucken [3571]
 (j/n):"
560 LOCATE 45,24:INPUT;td$                    [1345]
570 IF td$="j" THEN CLS:GOSUB 1440:GOSUB 1 [3383]
560:GOTO 610
580 IF td$="n" THEN CLS:GOTO 610              [2293]
590 IF td$<>"j/n" OR td$="" THEN GOTO 560  [1955]
600 '                                         [117]
610 '**** DATEN EINGABE ****                  [886]
620 '                                         [117]
630 LOCATE 10,7:PRINT"Monatl. oder 1/4 - j [5629]
aehrl. Zinsabrechnung ? (m/v)"
640 LOCATE 60,7:INPUT;abr$                    [1247]
650 IF abr$="m" THEN GOSUB 1280:GOTO 680   [1818]
660 IF abr$="v" THEN GOSUB 1280:GOTO 680   [2402]
670 IF abr$<>"m/v" OR abr$="" THEN GOTO 64 [1690]
0
680 LOCATE 20,11:PRINT"Schuldsumme            [2880]
  DM"
690 LOCATE 45,11:INPUT; sua                   [1102]
700 IF sua=0 THEN CLS:GOTO 70                 [2233]
710 IF abr$="m" THEN GOTO 730                 [1223]
720 IF abr$="v" THEN GOTO 740                 [2197]
730 LOCATE 20,13:PRINT"Monatl. Rueckzahlun [3656]
g  DM":GOTO 750
740 LOCATE 20,13:PRINT"1/4-Jaehrl.Rueckzah [3704]
l.DM":GOTO 750
750 LOCATE 45,13:INPUT; rz                    [1980]
760 IF rz<1 THEN PRINT:LOCATE 63,13:PRINT" [5286]
Bitte etwas mehr !":GOTO 750
770 LOCATE 20,15:PRINT"Zinssatz in            [2171]
  %"
780 LOCATE 45,15:INPUT; zs                    [1605]
790 LOCATE 20,20:PRINT"Eingabe richtig (j/ [3765]
n):"
800 LOCATE 45,20:INPUT; er$                   [1393]
810 IF er$="j" THEN GOSUB 1280:GOTO 840    [1960]
820 IF er$="n" THEN zs=0:GOSUB 1280:GOTO 6 [1935]
10
830 IF er$<>"j/n" OR er$="" THEN GOTO 800  [1452]
840 IF td$="j" THEN GOSUB 1610                [1163]
850 nr=1                                      [327]
860 IF abr$="m" THEN sun=sua-rz+((sua/100) [1967]
*zs/12)
870 IF abr$="v" THEN sun=sua-rz+((sua/100) [2217]
*zs/4)
880 t=sua-sun                                 [557]
890 z=rz-t                                    [858]
900 IF sun>sua OR sun=sua THEN PRINT:LOCAT [12681]
E 10,25:PRINT"Rueckzahlung zu gering !
     Darlehns-Konto nimmt zu !":FOR n=1 TO
 1500:NEXT:nr=0:zs=0:sua=0:dur$=INKEY$:GOS
UB 1300:GOTO 610
910 LOCATE 1,7:PRINT"Lfd.Nr.";                [1698]
920 LOCATE 10,7:PRINT Alter Saldo";           [2847]
930 LOCATE 29,7:PRINT"Zahlung";               [2212]
940 LOCATE 43,7:PRINT"Tilgung";               [1388]
950 LOCATE 57,7:PRINT"Zinsen";                [2450]
960 LOCATE 69,7:PRINT"Neuer Saldo"            [1921]
970 LOCATE 1,8:PRINT STRING$(80,"-")          [2596]
980 LOCATE 3,9:PRINT;nr;                      [2357]
990 LOCATE 8,9:PRINT USING"#######.## DM"; [1794]
sua;
1000 LOCATE 24,9:PRINT USING"######.## DM" [1402]
;rz;
```

```
1010 LOCATE 38,9:PRINT USING"######.## DM" [2481]
;t;
1020 LOCATE 51,9:PRINT USING"######.## DM" [3134]
;z;
1030 LOCATE 68,9:PRINT USING"######.## DM" [2403]
;sun
1040 IF td$="j" THEN GOSUB 1690             [1083]
1050 IF abr$="v" THEN GOTO 1070             [2197]
1060 IF sun<=rz THEN PRINT:PRINT STRING$(8 [11589]
0,"-"):PRINT TAB(3) nr+1;".und letzte Rate
";TAB(58)"Restschuld";USING"######.## DM";
sua-t+((sun/100)*zs/12):PRINT TAB(58)STRIN
G$(22,"="):GOTO 1210
1070 IF sun<=rz THEN PRINT:PRINT STRING$(8 [14862]
0,"-"):PRINT TAB(3) nr+1;".und letzte Quar
tals-Rate";TAB(58)"Restschuld";USING"#####
#.## DM";sua-t+((sun/100)*zs/4):PRINT TAB(
58)STRING$(22,"="):GOTO 1210

1080 IF dur$="s" THEN sua=sun:nr=nr+1:GOTO [2917]
 860
1090 IF dur$="e" THEN GOTO 1150             [1137]
1100 LOCATE 2,24:PRINT"Moechten Sie die na [8063]
echsten Raten einzeln oder im Schnelldurch
lauf berechnen ?"
1110 LOCATE 30,25:PRINT"(e/s)":LOCATE 40,2 [3385]
5:INPUT; dur$
1120 IF dur$="s" THEN GOTO 1080             [1615]
1130 IF dur$="e" THEN GOTO 1090             [1141]
1140 IF dur$<>"e/s" OR dur$="" THEN GOTO 1 [2898]
100
1150 LOCATE 1,24:PRINT STRING$(80,"-")      [1721]
1160 LOCATE 1,25:PRINT"naechste Rate berec [4011]
hnen  (j/n)     "
1170 LOCATE 40,25:INPUT; nm$                [1360]
1180 IF nm$="j" THEN sua=sun:nr=nr+1:GOSUB [2173]
 860:RETURN
1190 IF nm$="n" THEN GOTO 1210              [1194]
1200 GOTO 1170                              [305]
1210 LOCATE 1,24:PRINT STRING$(80,"-")      [1721]
1220 LOCATE 1,25:PRINT"Neue Berechnung      [2805]
       (j/n) ?  "
1230 LOCATE 40,25:INPUT; nb$                [1366]
1240 IF nb$="j" THEN nr=0:zs=0:sua=0:dur$= [4182]
INKEY$:CLS:GOTO 530
1250 IF nb$="n" THEN CLS:GOTO 70            [1331]
1260 GOTO 1230                              [365]
1270 '                                      [117]
1280 '**** BILDSCHIRM - AUFBAU ****         [2003]
1290 '                                      [117]
1300 CLS                                    [91]
1310 LOCATE 1,1:PRINT STRING$(80,"=")       [955]
1320 IF go$="b" THEN GOTO 1340              [1016]
1330 IF go$="c" THEN GOTO 1350              [768]
1340 LOCATE 26,3:PRINT" * TILGUNGS-PLAN *  [3021]
":GOTO 1360
1350 LOCATE 22,3:PRINT" * DARLEHENS-KOSTEN [2670]
 * "
1360 LOCATE 32,4:PRINT"Jahres-Zinssatz ";z [2658]
s;" %"
1370 IF abr$="m" THEN GOTO 1390             [1371]
1380  IF abr$="v" THEN GOTO 1400            [2219]
1390 LOCATE 2,4:PRINT"Darlehen = ";USING"# [7646]
######.## DM";sua:LOCATE 59,4:PRINT"Monatl
.Zinsabrechnung":GOTO 1410
1400 LOCATE 2,4:PRINT"Darlehen = ";USING"# [7947]
######.## DM";sua:LOCATE 59,4:PRINT"1/4-Ja
ehrl.Zinsabrechnung":GOTO 1410

1410 LOCATE 1,5:PRINT STRING$(80,"=")       [2090]
1420 RETURN                                 [555]
1430 '                                      [117]
1440 '**** ADRESSE DRUCKEN ****             [1786]
1450 '                                      [117]
1460 LOCATE 22,25:PRINT" * D R U C K E R    [4629]
 E I N S C H A L T E N *"
1470 FOR N=1 TO 1500                        [950]
1480 NEXT                                   [350]
1485 IF anname$="" THEN RETURN              [937]
1490 PRINT#8,CHR$(18);CHR$(27);"3";CHR$(24 [3875]
);TAB(6);an$
1500 PRINT#8,TAB(5);nam$                    [1348]
1510 PRINT#8,TAB(5);st$;hn                  [1186]
1520 PRINT#8,TAB(5);plz$;TAB(10);wo$        [2538]
1530 PRINT#8                                [343]
1540 RETURN                                 [555]
1550 '                                      [117]
```

```
1560 '**** KOPF DRUCKEN ****                [710]
1570 '                                      [117]
1580 PRINT#8,CHR$(18);CHR$(27);"3";CHR$(12 [6071]
);TAB(25);" * T I L G U N G S - P L A N *"
:PRINT#8
1590 RETURN                                 [555]
1600 '                                      [117]
1610 '**** ZINSSATZ DRUCKEN ****            [2274]
1620 '                                      [117]
1630 PRINT#8,CHR$(27);"3";CHR$(12);TAB(33) [3490]
;"Jahreszins :";zs;" %"
1640 IF abr$="m" THEN GOTO 1660             [1381]
1650 IF abr$="v" THEN GOTO 1670             [2316]
1660 PRINT#8,TAB(30);"Monatl. Zinsabrechnu [3308]
ng":RETURN
1670 PRINT#8,TAB(25);"Vierteljaehrl. Zinsa [5120]
brechnung":RETURN
1680 '                                      [117]
1690 '**** DATEN DRUCKEN ****               [904]
1700 '                                      [117]
1710 PRINT#8,CHR$(15),TAB(15);STRING$(115, [3599]
"-")
1720 PRINT#8,TAB(17);"Lfd. Nr.: ";          [1975]
1730 PRINT#8,TAB(31);"Alter Saldo ";        [2128]
1740 PRINT#8,TAB(50);"Zahlung ";            [2017]
1750 PRINT#8,TAB(65);"Tilgung ";            [2342]
1760 PRINT#8,TAB(81);"Zinsen ";             [2805]
1770 PRINT#8,TAB(96);"Neuer Saldo ";        [2929]

1780 PRINT #8,TAB(120);"Bezahlt am "        [1790]
1790 PRINT#8:                               [532]
1800 PRINT#8,TAB(15);nr;". Rate";           [1620]
1810 PRINT#8,TAB(29);USING"#######.## DM "  [2029]
;sua;
1820 PRINT#8,TAB(45);USING"######.## DM ";  [1910]
rz;
1830 PRINT#8,TAB(60);USING"######.## DM ";  [1550]
t;
1840  PRINT#8,TAB(75);USING"######.## DM "  [3485]
;z;
1850  PRINT#8,TAB(95);USING"######.## DM "  [2530]
;sun
1860 IF abr$="m" THEN GOTO 1880             [1385]
1870 IF abr$="v" THEN GOTO 1890             [2256]
1880 IF sun <=rz THEN PRINT#8,TAB(15);STRI [12445]
NG$(115,"-"):PRINT#8,CHR$(18):PRINT#8,TAB(
9)nr+1;". und letzte Monats-Rate";TAB(54)"
Restschuld";USING"######.## DM";sua-t+((su
n/100)*zs/12):PRINT#8,TAB(54)STRING$(22,"=
"):GOTO 1900
1890 IF sun <=rz THEN PRINT#8,TAB(15);STRI [11854]
NG$(115,"-"):PRINT#8,CHR$(18):PRINT#8,TAB(
9)nr+1;". und letzte Quartals-Rate";TAB(54
)"Restschuld";USING"######.## DM";sua-t+((
sun/100)*zs/4):PRINT#8,TAB(54);STRING$(22,
"=")
1900 IF sun <=rz THEN GOSUB 1920 ELSE RETU [2158]
RN
1910 '                                      [117]
1920 PRINT#8:PRINT#8:PRINT#8,TAB(24)"D A H [5637]
 R L E H E N S - K O S T E N":PRINT#8,TAB(
9) STRING$(66,"-"):PRINT#8
1930 IF abr$="m" THEN GOTO 1950             [1419]
1940  IF abr$="v" THEN GOTO 1970           [2232]
1950 IF abr$="m" THEN gesum=nr*rz + sun +  [16319]
( ( sun / 100 ) * zs / 12):PRINT#8,TAB(18)
nr;"Monatsrate(n)  a.";TAB(45)USING"######
.## DM";rz:PRINT#8:PRINT#8;TAB(16)" + 1
   Rate = Rest    =";TAB(45)USING"######.##
 DM";sun+((sun/100)*zs/12):PRINT#8:GOTO 19
60
1960 PRINT#8,TAB(25)"Gesamtkosten =";TAB(4 [7255]
4)USING"#######.## DM";gesum:PRINT#8,TAB(2
5) STRING$(32,"="):GOTO 1050
1970 IF abr$="v" THEN gesum=nr*rz + sun +  [13113]
( ( sun / 100 ) * zs / 4):PRINT#8,TAB(18);
nr;"   Quart.Rate(n) a.";TAB(45)USING"####
##.## DM";rz:PRINT#8:PRINT#8;TAB(18)" + 1
  Rate = Rest    =";TAB(45)USING"######.##
DM";sun+((sun/100)*zs/4):PRINT#8:GOTO 1980
1980  PRINT#8,TAB(25)"Gesamtkosten =";TAB( [7887]
45)USING"#######.## DM";gesum:PRINT#8,TAB(
25) STRING$(32,"="):GOTO 1050
1990 '                                      [117]
2000 '**** DARLEHENS-KOSTEN ****            [608]
2010 '                                      [117]
2020 '**** DATENEINGABE ****                [886]
```

```
2030 '                                          [117]
2040 LOCATE 10,7:PRINT"Monatl. oder 1/4-Ja [5674]
ehrl.Zinsabrechnung ? (m/v)"
2050 LOCATE 60,7:INPUT; abr$                    [1247]
2060 IF abr$="m" THEN GOSUB 1280:GOTO 2090 [1778]
2070 IF abr$="v" THEN GOSUB 1280:GOTO 2090 [2114]
2080 GOTO 2050                                  [361]
2090 LOCATE 20,11:PRINT"Gewuenschte Darleh [4191]
enssumme  DM :"
2100 LOCATE 55,11:INPUT; sua                    [1551]
2110 IF sua=0 THEN CLS:GOTO 70                  [2233]
2120 IF abr$="m" THEN GOTO 2140                 [1333]
2130 IF abr$="v" THEN GOTO 2150                 [2260]
2140 LOCATE 20,13:PRINT"Monatl.Rueckzahlun [3969]
g      DM:":GOTO 2160
2150 LOCATE 20,13:PRINT"Vierteljaehrl.Ruec [5423]
kzahlung      DM:":GOTO 2160
2160 LOCATE 55,13:INPUT; rz                     [1287]
2170 IF rz<1 THEN LOCATE 63,13:PRINT"Bitte [3971]
 etwas mehr !":GOTO 2160
2180 LOCATE 20,15:PRINT"Zinssatz in             [2401]
          % :"
2190 LOCATE 55,15:INPUT; zs                     [1125]
2200 LOCATE 10,22:PRINT"Eingabe Richtig (j [2097]
/n) :"
2210 LOCATE 35,22:INPUT; eing$                  [1503]
2220 IF eing$="j" THEN CLS:GOSUB 1280:GOTO [2145]
 2260
2230 IF eing$="n" THEN CLS:GOSUB 1280:GOTO [1643]
 2020
2240 GOTO 2210                                  [361]
2250 '                                          [117]
2260 '**** DATEN FUER DARLEHENSOKOSTEN BER [2110]
ECHNEN ****
2270 '                                          [117]
2280 nr=1                                       [327]
2290 IF zs=0 THEN sun=sua-rz:GOTO 2340      [1851]
2300 IF abr$="m" THEN sun=sua-rz+((sua/100 [2090]
)*zs/12):GOTO 2320
2310 IF abr$="v" THEN sun=sua-rz+((sua/100 [3018]
)*zs/4):GOTO 2320
2320 IF sun>sua OR sun=sua THEN LOCATE 10, [8881]
25:PRINT"Rueckzahlung zu gering !
Darlehenskonto nimmt zu !":GOTO 2020
2330 LOCATE 33,9:PRINT"Bitte Warten !":PRI [3159]
NT:PRINT
2340 t=sua-sun                                  [557]
2350 z=rz-t                                     [858]
2360 IF sun>=rz THEN sua=sun:nr=nr+1:gesum [3925]
=nr*rz+z:GOTO 2290
2370 IF abr$="m" THEN GOTO 2390                 [1443]
2380 IF abr$="v" THEN GOTO 2410                 [2280]
2390 gesum=nr*rz+sun+((sun/100)*zs/12):PRI [12727]
NT TAB(18) nr;"   Monats-Rate(n) a.";TAB(4
5)USING "######.## DM";rz:PRINT:PRINT TAB(
18)"+ 1  Rate = Rest = ";TAB(45) USING "##
####.## DM";sun+((sun/100)*zs/12):PRINT:GO
TO 2400
2400 PRINT TAB(25)"Gesamtkosten = ";TAB(44 [4061]
) USING"#######.##  DM";gesum:PRINT:GOTO 2
430
2410 gesum=nr*rz+sun+((sun/100)*zs/4):PRIN [15198]
T TAB(18) nr;"   QUART.-Rate(n) a.";TAB(45
)USING "######.## DM";rz:PRINT:PRINT TAB(1
8)"+ 1 Rate = Rest = ";TAB(45) USING "####
##.## DM";sun+((sun/100)*zs/4):PRINT:GOTO
2420
2420 PRINT TAB(25)"Gesamtkosten = ";TAB(44 [4061]
) USING"#######.##  DM";gesum:PRINT:GOTO 2
430
2430 PRINT TAB(25) STRING$(32,"=")              [845]
2440 LOCATE 1,24:PRINT STRING$(80,"-")          [1721]
2450 LOCATE 1,25:PRINT"Neue Berechnung (j/ [2841]
n) ?          "
2460 LOCATE 40,25:INPUT; nb$                    [1366]
2470 IF nb$="j" THEN CLS:nr=0:zs=0:sua=0:G [1803]
OTO 2020
2480 IF nb$="n" THEN CLS:GOTO 70                [1331]
2490 GOTO 2460                                  [367]
2500 LOCATE 36,13:PRINT"Ende ":END              [1069]
```

## An alle Sternenkrieger Labyrinthspezialisten und sonstige Spiele-Fans

Wer kennt das nicht, nach langen Stunden intensiven Spielens ist es gelungen, alle Probleme zu knacken, und das Spiel bis zum Schluß zu meistern.

Viele andere Spieler waren nicht so clever wie Sie und hängen irgendwo fest. Eine Situation, in der man, wie Sie sicherlich wissen, nur zu leicht verzweifelt.

Wenn Sie als erfolgreicher Spieler uns Ihre Lösungen mitteilen, können wir zusammen allen Hilfesuchenden helfen.

Also, halten Sie Ihre Spieletips nicht hinter dem Berg.

Schreiben Sie uns

**DMV**
*Daten & Medien*
*Verlaggesellschaft mbH*
*Postfach 250 · Fuldaer Str. 6*
*3440 Eschwege*
*z.H. Herrn Stiller*

Wir freuen uns
schon heute auf Ihren Brief,
und mit uns
viele, viele Spiele-Fans.

# DEFPRINT

**DEFPRINT ist eine Kombination aus Zeicheneditor und Druckertreiber. Das Programm dient dazu, eigene Zeichensätze zu entwerfen, zu speichern, zu überarbeiten und in den Drucker zu laden.**

Ohne ein solches Hilfsprogramm ist das kaum zu schaffen. In vielen Handbüchern fehlen die nötigen Informationen. Im übrigen ist die Definition eigener Druckerzeichen schwieriger als das Entwerfen von Bildschirmzeichen, wegen der Regel, daß nicht zwei Nadeln überlappend gesetzt werden dürfen. Aus diesem Grunde gibt es auch keinen Algorithmus zum Invertieren von Druckerzeichen; man muß verschiedene Möglichkeiten durchprobieren und nach künstlerischen Gesichtspunkten entscheiden. Diesem Zweck dient der Editor DEFPRINT. Er erlaubt Probedrucke und Veränderungen des Entwurfs, bevor ein Zeichen in den erzeugten Zeichensatz übernommen wird. Vorhandene Zeichensätze können wieder eingeladen und editiert werden.
Das Programm ist getestet auf den Druckern Epson FX-85 und STAR SG-15. Es sollte daher auf den meisten 9-Nadel-Druckern laufen. Die beiden Drucker unterscheiden sich in den Grafiksteuerzeichen. Für den STAR muß eine markierte Programmzeile gelöscht oder die Variable »dr« auf Null gesetzt werden. Diese Druckerkennung wird mit den erzeugten Zeichensätzen abgespeichert. Ein Epson-File läuft demnach nur auf einem Epson Drucker richtig. Vollständig ausnutzen kann man es nur, wenn der Centronics-Ausgang des CPC-Computers auf 8 Bit erweitert ist, wie es in zahlreichen Bastelanleitungen beschrieben wurde. Sonst lassen sich nur die Zeichen mit den ASCII-Codes < 128 umdefinieren und man kann nur 8 Nadeln des Druckkopfs ansprechen. DEFPRINT ist auf dem Schneider 464 geschrieben. Nach meiner Absicht soll es auf allen Computern der CPC-Reihe laufen. Ich weiß aber nicht, ob der im Programm verwendete Patch für den 8-Bit-Betrieb auf dem CPC664/6128 richtig arbeitet. Wie im Programm vermerkt, entstammt er der Zeitschrift CPC International Nr. 11/86. Als Ergänzung des Programms ist die Routine DEFINST zu betrachten. Sie lädt einfach einen gespeicherten Zeichensatz in den Drucker, ohne ihn zu bearbeiten. Man kann sie als Vorspann zu Programmen verwenden, die einen anwenderdefinierten Zeichensatz benötigen.

## Die Programme

### DEFPRINT

1. Drucker und Computer aus- und wieder einschalten
2. RUN« defprint

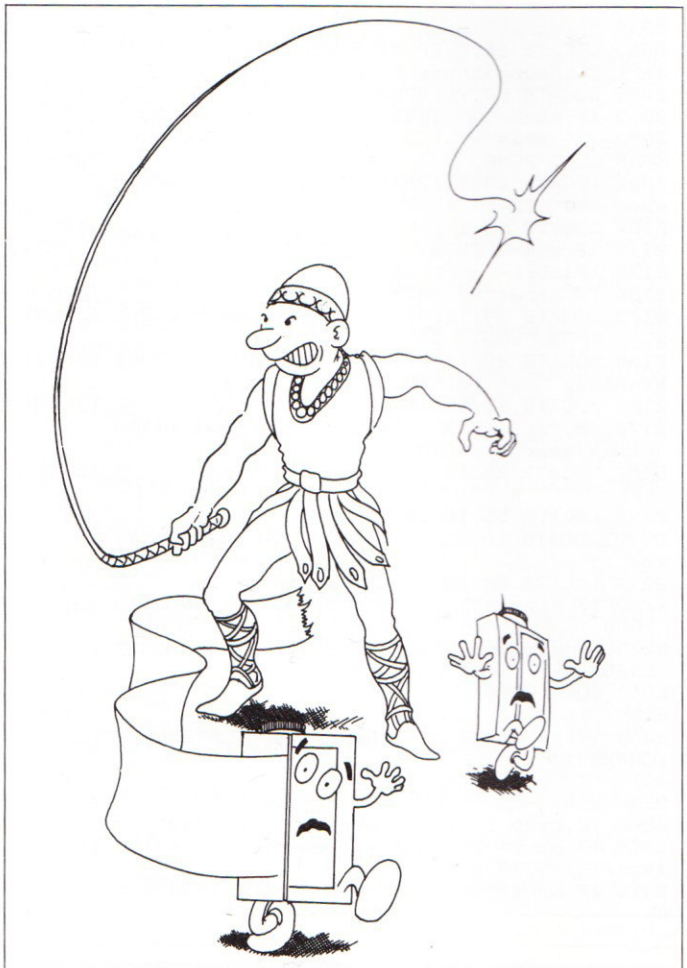
Der Benutzer wird vom Programm geführt. Neben der programminternen Bedienungsanleitung, die man sich auch ausdrucken lassen kann, bedarf es keiner weiteren Erläuterungen. Auf die Frage nach der Proportionalbreite eines definierten Zeichens sollte in der Regel der voreingestellte Wert (volle Breite) übernommen werden. Zur Realisierung von Proportionalschrift muß das Druckerhandbuch zu Rate gezogen werden.
Erläuterungen zum Programm und Hilfen für eventuelle Anpassungen an andere Drucker finden sich im Listing. Beim Abtippen können sämtliche REMarks weggelassen werden.

### DEFINST

1. RUN« definst

Das Programm ruft eine Symboldatei mit dem Namen DEFDAT auf. Natürlich kann in Zeile 240 auch ein anderer Name eingesetzt werden. Selbstdefinierte Zeichen können nicht im NLQ-Modus gedruckt werden. Jedenfalls ist das bei Epson Druckern so.

(Dieter Taube)



```
100 '************************************ [1543]
*
110 '*                                    [175]
*
120 '*         DEFPRINT   V.1.1           [1761]
*
130 '*         ================           [175]
*
140 '*    Druckerzeichen definieren       [1870]
*
150 '*                                    [175]
*
160 '*        fuer Schneider CPC          [1497]
*
170 '* und EPSON FX-85 oder STAR SG-10/15  [2275]
*
180 '*     (C) Dieter Taube, Rinteln      [1628]
*
190 '*                                    [175]
*
200 '************************************ [1543]
*
210 '                                     [117]
220 DEFSTR b:DEFINT i-z                   [458]
230 '                                     [117]
240 '-------------------                  [1051]
250 ' Drucker-Anpassung                   [1913]
260 '-------------------                  [1051]
270 '                                     [117]
280 dr=1:' Fuer Star-Drucker diese Zeile l [2496]
oeschen
290 '                                     [117]
300 name$(1)="EPSON":name$(0)="STAR "     [2838]
310 kop$(1)=CHR$(27)+":"+STRING$(3,0):kop$ [3785]
(0)=CHR$(27)+"*"+CHR$(0)
320 user$=CHR$(27)+CHR$(36+dr)+CHR$(1)    [2047]
```

```
330 defi$(1)=CHR$(27)+"&"+CHR$(0):defi$(0) [2850]
=CHR$(27)+"*"+CHR$(1)
340 stz$(1)=CHR$(27)+"6":stz$(0)="" [1844]
350 ' [117]
360 '------------------ [1051]
370 ' Initialisierungen [1441]
380 '------------------ [1051]
390 ' [117]
400 '8.Bit-Patch von A.Hartmeier, aus CPC- [3817]
International 11/86:
410 FOR i=&B072 TO &B08A:READ j:POKE i,j:N [2539]
EXT
420 POKE &BD2B,195:POKE &BD2C,114:POKE &BD [2626]
2D,176
430 DATA 254,128,56,18,197,213,1,0,246,17, [4526]
0,32,237,81,205,136,176,237,89,209,193,201
,207,242,135
440 'vorgeschlagene Anpassung an CPC664/61 [3960]
28 (nicht getestet):
450 IF PEEK(6)<>128 THEN POKE &B08A,136 EL [3004]
SE 480
460 IF PEEK(6)=123 THEN POKE &B089,11 ELSE [3162]
 POKE &B089,27
470 ' [117]
480 MODE 1:PRINT CHR$(23)CHR$(0) [2060]
490 SYMBOL AFTER 123 [1457]
500 SYMBOL 123,108,0,120,12,124,204,118 [2198]
510 SYMBOL 124,102,0,60,102,102,102,60 [1824]
520 SYMBOL 125,102,0,102,102,102,102,62 [2270]
530 SYMBOL 126,12,50,50,62,50,60,48,48 [1645]
540 SYMBOL 254,0,0,34,20,0,20,34 [1962]
550 SYMBOL 255,0,0,28,62,62,62,28 [1903]
560 e$=" "+CHR$(8)+CHR$(242)+CHR$(9)+CHR$( [7473]
243)+CHR$(11)+CHR$(240)+CHR$(10)+CHR$(241)
+"X"+CHR$(224)+CHR$(127)+CHR$(17)
570 DIM z(12,8),d(11),dat(255,11) [1153]
580 BEEP=CHR$(7):GET=&BB4B [991]
590 LOCATE 13,16:PRINT BEEP"Anleitung ? [3084]
n"CHR$(8);
600 GOSUB 1060:CALL GET:IF NOT INKEY(45)TH [4645]
EN GOSUB 2140 ELSE CLS
610 WINDOW#1,10,15,7,14 [1435]
620 WINDOW#2,17,40,7,14:PEN#2,2 [1640]
630 WINDOW#3,2,40,23,25 [906]
640 ' [117]
650 '---------------- [874]
660 ' Eingabe-Editor [1101]
670 '---------------- [874]
680 ' [117]
690 IF INP(&F500)=90 THEN LOCATE 9,12:PRIN [5699]
T CHR$(24)" Drucker einschalten ! "CHR$(24
)BEEP
700 PRINT#8,kop$(dr);:'    Drucker-Zeichen [2739]
satz in RAM kopieren
710 PRINT#8,user$;:'       RAM-Zeichensatz [2821]
 einschalten
720 PRINT#8,stz$(dr);:'    Steuerzeichen a [3371]
usdrucken, nur Epson
730 CLS:a$=STRING$(40,154) [1702]
740 PRINT a$"* SELBST-DEFINIERTE ZEICHEN F [4373]
UER "name$(dr)" *"a$
750 LOCATE 2,23:PRINT BEEP"Alten Zeichensa [6355]
tz laden (j/n) ?  n"CHR$(8);
760 GOSUB 1080:IF q$="j"THEN GOSUB 1950 [2054]
770 CLS#2:CLS#3 [637]
780 PRINT#2,TAB(7)"Nadeln setzen","mit Pfe [10650]
iltasten und COPY  oder Joystick & Fire":P
RINT#2:PRINT#2,TAB(4)"Matrix invertieren":
PRINT#2,TAB(8)"mit CTRL Q
790 PRINT#2:PRINT#2,TAB(6)BEEP"Beenden mit [1584]
 DEL
800 GOSUB 1160:' Raster zeichnen [2505]
810 TAG:GOSUB 1000:' Markierung setzen [2229]
820 CALL &BB7E:' Cursor aus [1960]
830 ' [117]
840 ' Beginn Eingabeschleife: [1625]
850 ' [117]
860 CALL GET:a=INSTR(e$,INKEY$) [1775]
870 IF a<2 THEN 860 ELSE GOSUB 1040:' Curs [2770]
orkreuz loeschen
880 ON a-1 GOTO 890,890,900,900,910,910,92 [3033]
0,920,930,930,1300,1100
890 x=MAX(x-1,1):GOTO 950:'   links [3555]
900 x=MIN(x+1,11):GOTO 950:'  rechts [2509]
910 y=MAX(y-1,1):GOTO 950:'   hoch [2074]
920 y=MIN(y+1,8):GOTO 950:'   runter [2200]
930 IF z(x-1,y)+z(x+1,y)<>0 THEN SOUND 1,9 [2931]
00:GOSUB 1040:GOTO 860:' Ueberlappung
940 SOUND 1,90,10:z(x,y)=1-SGN(z(x,y)):GOS [4898]
UB 1030:GOTO 860:' Nadel ein/aus
950 GOSUB 1000:GOTO 860 [1350]
960 ' [117]
970 ' Unterroutinen: [1239]
980 ' [117]
990 'Cursor bewegen: [956]
1000 MOVE(x-1)*8+1,-(y-1)*16-1:IF z(x,y)=0 [5027]
 THEN PRINT CHR$(254);
1010 RETURN [555]
1020 'Nadel ein/aus, Cursorkreuz: [2526]
1030 MOVE(x-1)*8+1,-(y-1)*16-1:PRINT CHR$( [3501]
255);
1040 IF z(x,y)=0 THEN MOVER-16,0:PRINT CHR [3803]
$(254);
1050 'Ton halten: [952]
1060 WHILE SQ(1)>127:WEND:RETURN [1992]
1070 'Ein-Zeichen-Eingabe: [2244]
1080 WHILE SQ(1)>127:WEND:CALL GET:q$=LOWE [4731]
R$(INKEY$):PRINT q$;:RETURN
1090 'Matrix invertieren: [2348]
1100 ERASE z,d:TAGOFF:PRINT CHR$(23)CHR$(0 [3127]
):DIM z(12,8),d(11):GOSUB 1160
1110 PRINT CHR$(23)CHR$(1):LOCATE 1,1:TAG [3057]
1120 FOR y=1 TO 8:FOR x=1 TO 11 STEP 2 [2536]
1130 z(x,y)=1:MOVE(x-1)*8+1,-(y-1)*16-1:PR [2588]
INT CHR$(255);
1140 NEXT x,y:x=1:y=1:GOTO 860 [1344]
1150 'Pin-Raster zeichnen: [1526]
1160 ORIGIN 16,306,16,112,306,178:CLG [1298]
1170 FOR i=0 TO 96 STEP 16:MOVE i,0:DRAW i [2782]
,-128:NEXT
1180 FOR i=-128 TO 0 STEP 16:MOVE 0,i:DRAW [2206]
 96,i:NEXT
1190 'Cursor-Anfangsposition: [725]
1200 x=1:y=1 [850]
1210 'XOR-Grafikmodus setzen: [1157]
1220 PRINT CHR$(23)CHR$(1);:RETURN [2523]
1230 ' [117]
1240 '------------------------ [1287]
1250 ' Drucker-Programmierung [2366]
1260 '------------------------ [1287]
1270 ' [117]
1280 ' fehlende Parameter berechnen [1701]
1290 ' [117]
1300 TAGOFF:CALL &BB7B [1607]
1310 CLS#3:PRINT BEEP;:INPUT#3,"ASCII-Code [4344]
 des neuen Zeichens :    ",a0
1320 IF a0<1 OR a0>255 THEN 1310 [1423]
1330 LOCATE 2,24:PRINT BEEP"Hat es Unterl{ [5416]
nge (j/n) ?          n"CHR$(8);
1340 GOSUB 1080:u=-(q$<>"j") [1682]
1350 LOCATE 2,25:PRINT BEEP"Proportionalbr [5856]
eite Standardwert ?  j"CHR$(8);
1360 GOSUB 1080:IF q$<>"n"THEN a=0:e=11:GO [2729]
TO 1380
1370 LOCATE 2,25:PRINT CHR$(20)BEEP;:INPUT [5930]
"erste,letzte Spalte (0...11): ",a,e
```

```
1380 f=128*u+16*a+e:' Code fuer Hoehe und  [3252]
Breite des Zeichens
1390 CLS#3                                [371]
1400 'Spaltensummen:                      [1198]
1410 sum=0:FOR x=1 TO 11:d(x)=0           [902]
1420 FOR y=1 TO 8:d(x)=d(x)+z(x,y)*2^(8-y) [5690]
:NEXT:' gesetzte Nadeln in 11 Spalten
1430 sum=sum+d(x):NEXT                    [1065]
1440 IF sum=0 THEN dat(a0,0)=0:GOTO 1630  [1602]
1450 GOSUB 1480:GOTO 1560                 [1402]
1460 '                                    [117]
1470 'Zeichen plotten als Merkhilfe:      [3120]
1480 vtab=VPOS(#1):PRINT#1,USING"###";a0  [1687]
1490 CALL &BBBA:ORIGIN 208,321-vtab*16    [1773]
1500 FOR x=1 TO 11:FOR y=1 TO 8:IF z(x,y)= [3800]
1 THEN PLOT x,-2*y
1510 NEXT y,x                             [397]
1520 RETURN                               [555]
1530 '                                    [117]
1540 ' Definition an Drucker              [503]
1550 '                                    [117]
1560 GOSUB 1570:GOSUB 1760:GOTO 1600      [2495]
1570 PRINT#8,defi$(dr)STRING$(2,a0)CHR$(f) [3164]
;:' Definition eines Zeichens
1580 FOR i=1 TO 11:PRINT#8,CHR$(d(i));:NEX [2191]
T
1590 RETURN                               [555]
1600 LOCATE 2,24:PRINT BEEP"Probeausdruck  [4960]
(j/n) ?  n"CHR$(8);
1610 GOSUB 1080:IF q$<>"j"THEN 1630       [2252]
1620 PRINT#8,STRING$(5,a0);:FOR i=1 TO 5:P [6511]
RINT#8," "CHR$(a0);:NEXT:PRINT#8,"Test
1630 CLS#3:LOCATE 5,24:PRINT BEEP"Ergebnis [4275]
 speichern (j/n) ?    n"CHR$(8);
1640 GOSUB 1080:IF q$="j"THEN GOSUB 1800  [2060]
1650 CLS#3:LOCATE 5,24:PRINT BEEP"Neues Ze [3072]
ichen (j/n) ?    j"CHR$(8);
1660 GOSUB 1080:IF q$="n"THEN x=1:y=1:ORIG [4075]
IN 16,306,16,112,306,178:GOTO 810
1670 CLS#3:LOCATE 1,1                     [316]
1680 PRINT CHR$(23)CHR$(0):ERASE z,d:DIM z [3735]
(12,8),d(11):GOTO 800
1690 '                                    [117]
1700 '--------------------                [1177]
1710 ' Ergebnis speichern                 [823]
1720 '--------------------                [1177]
1730 '                                    [117]
1740 ' 1. Datenpuffer des Programms:      [2176]
1750 '                                    [117]
1760 dat(a0,0)=f:FOR i=1 TO 11:dat(a0,i)=d [4699]
(i):NEXT:RETURN
1770 '                                    [117]
1780 ' 2. Sicherung auf Diskette:         [1788]
1790 '                                    [117]
1800 IF datei$<>""THEN 1820               [917]
1810 CLS#3:LOCATE 5,24:PRINT BEEP;:INPUT"U [6082]
nter welchem Dateinamen ? ",datei$
1820 OPENOUT datei$                       [738]
1830 PRINT#9,dr                           [1038]
1840 FOR ascii=1 TO 255                   [1056]
1850 IF dat(ascii,0)=0 THEN 1880:' keine A [1776]
enderung
1860 PRINT#9,ascii                        [1315]
1870 FOR i=0 TO 11:PRINT#9,dat(ascii,i):NE [3512]
XT
1880 NEXT ascii                           [989]
1890 CLOSEOUT:RETURN                      [1447]
1900 '                                    [117]
1910 '-------------------------           [1409]
1920 ' alten Zeichensatz laden            [923]
1930 '-------------------------           [1409]
1940 '                                    [117]
1950 LOCATE 2,24:PRINT BEEP;:INPUT"Welche  [3999]
Datei laden ? ",altdat$
1960 LOCATE#2,2,2:PRINT#2,"Anzeige unterdr [3938]
}cken
1970 PRINT#2," mit niedergehaltener":PRINT [4402]
#2," Leertaste ...
1980 OPENIN altdat$                       [605]
1990 INPUT#9,i                            [683]
2000 WHILE NOT EOF                        [1840]
2010 INPUT#9,a0,f:dat(a0,0)=f             [1333]
2020 FOR i=1 TO 11:INPUT#9,d(i):dat(a0,i)= [2762]
d(i):NEXT
2030 GOSUB 1570:' Zeichen an Drucker      [1823]
2040 IF NOT INKEY(47)THEN 2070:' ueberspri [4284]
ngen mit Leertaste
2050 FOR x=1 TO 11:FOR y=1 TO 8:z(x,y)=-(( [3390]
d(x)AND 2^(8-y))<>0):NEXT y,x
2060 GOSUB 1480:' eingelesene Zeichen anze [1944]
igen
2070 WEND                                 [390]
2080 ERASE z:DIM z(12,8):RETURN           [1201]
2090 '                                    [117]
2100 '-----------                         [597]
2110 ' Anleitung                          [439]
2120 '-----------                         [597]
2130 '                                    [117]
2140 MODE 2:INK 1,20:LOCATE 1,3:d=0       [2669]
2150 RESTORE 2190:FOR i=1 TO 20:READ a$:PR [2797]
INT#d,a$:NEXT
2160 LOCATE 1,25:PRINT CHR$(24)SPC(24)"* C [6119]
OPY: Diesen Text zum Drucker *"SPC(22)CHR$
(24)BEEP;
2170 GOSUB 1060:CALL GET:IF NOT INKEY(9)TH [4926]
EN d=8:WIDTH 255:GOTO 2150
2180 MODE 1:INK 1,24:RETURN               [1717]
2190 DATA "  DEFPRINT ist ein Zeichenedito [15054]
r f}r EPSON- und STAR-Drucker. Besitzer ei
nes","  STAR-Druckers m}ssen vor dem Progr
ammstart die Basic-Zeile 280 l{schen.",
2200 DATA Bei jedem Speicher- oder Ladevor [10765]
gang wird der gesamte Vorrat an definierte
n,Zeichen gesichert und der Drucker entspr
echend programmiert. Das Programm kann
2210 DATA deshalb jederzeit mit ESC oder R [7834]
ESET abgebrochen werden. Danach stehen auf
 dem,Drucker die neuen Zeichen zur Verf}gu
ng.
2220 DATA "Definierte Zeichen k}nnen nacht [21306]
r{glich ge{ndert werden. Die }bernahme ein
es","Leerzeichens mit DEL restauriert die
urspr}ngliche Form. Beachten Sie, da~ nich
tjeder ASCII-Code mit einem druckbaren Zei
chen belegt werden kann. Hier hilft"
2230 DATA Probieren oder das Studium des D [14477]
ruckerhandbuchs. Ohne das 8.Bit des Centro
nics-,"Ausgangs k}nnen Zeichen mit ASCII-C
odes>127 nicht definiert werden. So bleibt
",der Befehl ESC'6'(Epson) zum Ausdruck de
r Steuerzeichen 128-159 wirkungslos.
2240 DATA "Wenn Sie Ihren Schneider noch n [12443]
icht umgel}tet haben und nur }ber eine 7-B
it-","Schnittstelle verf}gen, dann wird au
ch die obere Nadelreihe nicht ausgedruckt.
"
2250 DATA Au~erdem werden alle Zeichen zwe [8209]
i Pixelreihen zu tief gesetzt (Unterl{nge)
.,Das mu~ beim Setzen der Nadeln ber}cksic
htigt werden.,
2260 DATA "  Die mit DEFPRINT erstellten D [18780]
ruckercodes werden auf Diskette abgelegt.
Zur","  Einbindung in eigene Programme die
nt die kurze Routine DEFINST, welche dem",
"  eigentlichen Programm einfach vorangest
ellt wird."
```

```
100 ' DEFINST  V.1.1                                [649]
110 ' ==============                                [1050]
112 '                                               [117]
115 ' fuer Schneider CPC                            [494]
120 '                                               [117]
130 DEFINT a-z:DIM d(11)                            [763]
140 '                                               [117]
150 '------------------------------------ [3711]
------------------------------------
160 ' 8.Bit-Patch von A.Hartmeier, veroeff [4809]
enlicht in CPC-Intern. 11/86 S.24 :
170 '                                               [117]
180 FOR i=&B072 TO &B08A:READ j:POKE i,j:N [2539]
EXT
190 POKE &BD2B,195:POKE &BD2C,114:POKE &BD [2626]
2D,176
200 DATA 254,128,56,18,197,213,1,0,246,17, [4526]
0,32,237,81,205,136,176,237,89,209,193,201
,207,242,135
202 'eventuell anpassen an CPC664/6128:     [2424]
203 IF PEEK(6)<>128 THEN POKE &B08A,136 EL [2878]
SE 230
204 IF PEEK(6)=123 THEN POKE &B089,11 ELSE [3162]
 POKE &B089,27
210 '------------------------------------ [3711]
------------------------------------
220 '                                               [117]
230 IF INP(&F500)=90 THEN CLS:PRINT CHR$(7 [4849]
)"Drucker einschalten !
240 OPENIN"defdat                                   [1510]
245 INPUT#9,dr                                      [815]
250 IF dr=0 THEN 320                                [335]
260 'Druckbefehle fuer Epson:                       [1796]
270 PRINT#8,CHR$(27)":"STRING$(3,0);:' Dru [4884]
cker-Zeichensatz in RAM kopieren
280 PRINT#8,CHR$(27)"6";:'              Ste [3648]
uerzeichen als Druckzeichen behandeln
290 PRINT#8,CHR$(27)"%1";:'             def [2898]
inierten Zeichensatz einschalten
300 GOTO 350                                        [470]
310 'Druckbefehle fuer Star:                        [2106]
320 PRINT#8,CHR$(27)"*"CHR$(0);:'       Dru [2904]
cker-Zeichensatz in RAM kopieren
330 PRINT#8,CHR$(27)"$1";:'             def [3607]
inierten Zeichensatz einschalten
340 '                                               [117]
350 WHILE NOT EOF                                   [1840]
360 'Einlesen:                                      [1385]
370 INPUT#9,a0,f:' Ascii-Code und Informat [5279]
ion ueber Hoehe & Breite des Zeichens
380 FOR i=1 TO 11:INPUT#9,d(i):NEXT:' gese [4558]
tzte Nadeln in 11 Spalten
390 'Uebergabe an den Drucker:                      [1866]
400 IF dr=0 THEN PRINT#8,CHR$(27)"*"CHR$(1 [4008]
)CHR$(a0)CHR$(a0)CHR$(f);:GOTO 410
405 PRINT#8,CHR$(27)"&"CHR$(0)CHR$(a0)CHR$ [3611]
(a0)CHR$(f);
410 FOR i=1 TO 11:PRINT#8,CHR$(d(i));:NEXT [2191]
420 WEND                                            [390]
430 CLOSEIN                                         [752]
440 '                                               [117]
450 '--- RUN"programm ---                           [2299]
460 '                                               [117]
470 WIDTH 255:FOR i=128 TO 255:PRINT#8,CHR [5038]
$(i);:NEXT:PRINT#8,"  * TESTAUSDRUCK *
```


Für unsere ständige CPC-Rubrik suchen wir noch

# Programme, Tips + Tricks

zur Veröffentlichung. Honorar nach Vereinbarung.

Einsendung an:

**DMV Daten & Medien Verlagsgesellschaft mbH**
**Fuldaer Straße 6, 3440 Eschwege**

# Timeout

**Automatisches Abschalten des Bildschirms bei längerer Nichtbenutzung zur Schonung der Bildröhre**

Es gibt eine Reihe von Computeranwendungen, bei denen das System stunden- oder auch tagelang läuft. Das dürfte für viele Anwendungeh im Bürobereich zutreffen, aber auch etwa für Mailboxen. Falls nun am System zum Beispiel eine Stunde lang nichts gearbeitet wird (Mittagspause!), zeigt der Bildschirm die ganze Zeit dasselbe an. Das kann, vor allem bei größerer Helligkeit zu Schäden führen, weil sich der Elektronenstrahl »einbrennt«. Professionelle Terminals schalten daher nach einigen Minuten Untätigkeit die Bildröhre ab; bei der ersten Aktivität ist sie aber wieder »voll da«.
Das gleiche Verhalten ist nun auch auf den CPCs verfügbar. Die RSX-Befehlsgruppe TIMEOUT sorgt mit folgenden drei Befehlen dafür:

!TIMEOUT,n
!DARK
!NORMAL

!TIMEOUT,n setzt die Wartezeit, nach der der Bildschirm abzudunkeln ist. Der Bildschirm wird automatisch dunkel, sobald seit dem letzten Tastendruck, seit der letzten Bildschirmausgabe, seit dem letzten TIMEOUT-Befehl sowie seit dem letzten NORMAL-Befehl mindestens n Sekunden verstrichen sind. (1 <= n <=65535. 65535 Sekunden = ca. 18 Stunden). n = 0 stellte den Effekt wieder ab.
Der Bildschirm wird wieder hell durch einen Tastendruck, eine Bildschirmausgabe oder einen TIMEOUT- oder NORMAL-Befehl.

!DARK schaltet den Bildschirm dunkel. Über das Hellwerden siehe oben. Falls Sie DARK nicht vom Programm ausführen lassen, sondern direkt eingeben, so wird der Befehl zwar durchgeführt, die folgende Ausgabe des »Ready« schaltet aber den Bildschirm sofort wieder hell, so daß keine Wirkung zu erkennen ist. Das läßt sich umgehen durch eine Verzögerungsschleife:

!DARK:FOR i=1 TO 5000:NEXT

!NORMAL schaltet den Bildschirm wieder hell. Die Anwendung von DARK und NORMAL ist wohl nur aus einem Programm heraus sinnvoll, etwa zum verdeckten Zeichnen von Graphiken.

## Eingabe des Programms

Tippen Sie das Listing ab und SAVEn Sie es ab.

Starten Sie es mit RUN. Der Maschinencode wird geladen und initialisiert. Ab jetzt sind die Befehle verfügbar. Das Programm löscht sich selbst.

Diese Version legt den Maschinencode unmittelbar unter HIMEM ab. Sie können das ändern, indem Sie die Zeilen 110 und 530 umbauen.

Sie können auch den reinen Maschinencode abspeichern, indem Sie eine Zeile

515 PRINT h+1:SAVE»TIMEOUT.BIN«,B,h+1,552,h+1

### für 464-664-6128

```
10 '************************************    [1460]
15 '*                                  *    [175]
20 '*                                  *    [175]
25 '*          T I M E O U T           *    [550]
30 '*                                  *    [175]
35 '*             C  1987              *    [71]
40 '*                                  *    [175]
45 '*           Georg Beran            *    [596]
50 '*                                  *    [175]
55 '************************************    [1460]
60 '                                        [117]
65 'REM-Statements brauchen nicht abge-     [4730]
  tippt werden
70 'Vorliegendes Programm legt MC-Code      [7067]
  direkt unter HIMEM ab (vgl. Zeile
110). Ablage an anderen Stellen ist     mo
eglich:
75 'Zeile 120: Ablage von h+1 bis inkl.     [8434]
  h+552. Bereich h+1 bis inkl. h+&67
ist nach der Initialisierung wieder     fr
ei (vgl. Zeile 530). Nettoplatzbe-    darf
 daher 552 - &67 = 449 Byte.
80 'Diese m}ssen zwischen &4000 und         [3063]
  &BFFF liegen.
85 '                                        [117]
100 SYMBOL AFTER 256                        [1408]
110 MEMORY HIMEM-552:h=HIMEM                [1025]
120 FOR i=h+1 TO h+552:READ a$:POKE i,VAL(  [3130]
"&"+a$):NEXT
130 DATA 3E,07,11,92,A0,21,61,A0,4E,23,46,  [3572]
23,E5,69,60,01,03,00,ED,B0
140 DATA E1,3D,20,F0,06,05,21,6B,A0,11,83,  [3678]
A0,1A,13,4F,1A,13,D5,59,57
```

```
150 DATA 13,4E,23,7E,23,E5,69,67,23,ED,A0,  [3801]
03,7E,E6,3F,12,E1,D1,10,E4
160 DATA 06,07,21,61,A0,11,75,A0,E5,4E,23,  [2400]
66,69,36,C3,23,1A,77,23,13
170 DATA 1A,77,13,E1,23,23,10,EC,21,89,A0,  [3701]
01,D3,A0,C3,D1,BC,D3,BD,06
180 DATA BB,09,BB,18,BB,1B,BB,38,BC,32,BC,  [2289]
32,BC,35,BC,3B,BC,00,00,00
190 DATA 00,00,00,00,00,00,00,00,00,00,00,  [1962]
00,00,00,00,00,00,00,00,00
200 DATA 00,00,00,00,00,00,00,00,00,00,00,  [1962]
00,00,00,00,00,00,00,00,00
210 DATA 00,00,00,00,00,00,00,00,00,00,00,  [1488]
00,00,00,00,00,81,4A,A1,00
220 DATA 00,00,00,00,00,00,00,00,00,81,61,  [2811]
A1,00,54,49,4D,45,4F,55,D4
230 DATA 44,41,52,CB,4E,4F,52,4D,41,CC,00,  [2940]
C1,A0,18,14,00,18,09,00,F3
240 DATA CD,44,A1,28,37,18,2D,F3,CD,44,A1,  [2720]
20,4E,18,6E,3D,C0,DD,7E,00
250 DATA DD,46,01,21,8F,A0,77,23,70,B0,20,  [3283]
0E,F3,CD,44,A1,28,03,CD,B9
260 DATA A1,CD,31,A1,18,35,F3,CD,44,A1,28,  [3627]
03,CD,B9,A1,CD,3F,A1,18,00
270 DATA 21,8F,A0,11,8D,A0,7E,ED,A0,B6,28,  [3525]
0D,ED,A0,21,A7,A0,11,32,00
280 DATA 42,4B,C3,E9,BC,21,A7,A0,C3,EC,BC,  [3667]
21,B4,A0,11,05,00,18,ED,21
290 DATA B4,A0,18,F0,AF,21,91,A0,BE,C9,ED,  [3373]
4B,8D,A0,ED,A1,ED,43,8D,A0
300 DATA E8,CD,44,A1,C0,CD,85,A1,CD,31,A1,  [3772]
18,D6,CD,44,A1,28,D9,3E,15
310 DATA CD,1E,BB,20,A3,3E,17,CD,1E,BB,C8,  [3025]
18,9B,32,91,A0,F3,D9,51,CB
320 DATA 91,ED,49,D9,21,67,A0,AF,C9,3E,FF,  [2975]
CD,75,A1,F5,E5,CD,00,00,E1
330 DATA 70,23,71,23,F1,F5,E5,01,00,00,CD,  [3509]
```

einfügen. Das können Sie dann laden mit LOAD" TIMEOUT.BIN:CALL h+1,h+1 ist der Wert vom SAVE her. Sie müssen ihn aufschreiben.

Unter der »Bildschirmausgabe« ist jede Ausgabe zu verstehen, die über die Firmware-Indirection TXT WRITE CHAR (BDD3) geleitet wird. Aus BASIC ist das vor allem das PRINT-Kommando (ausgenommen reine Steuerzeichensequenzen). Graphikbefehle zählen in diesem Zusammenhang n i c h t als Bildschirmausgabe. Sie können also »verdeckt« zwar Linien zeichnen, aber keine Texte ausgeben (außer mit TAG; das zählt dann als Graphik).

Der Bildschirm wird abgedunkelt, indem die derzeit gültigen Werte für die 16 Inks und den Border abgefragt und gespeichert werden. Anschließend werden die Inks auf 0 (schwarz) gesetzt.

Entsprechend wird der Bildschirm hell geschaltet, indem die Ink-Werte aus dem Zwischenspeicher geholt und neu gesetzt werden. Auf diese Art wird der Inhalt des Bildschirmspeichers durch die Abdunklung überhaupt nicht verändert. Auch Hardcopyroutinen arbeiten daher korrekt (Ausnahme: wenn eine Hardcopyroutine zur Ermittlung der Grautöne auf die geltenden Ink-Werte zugreift, wird sie nur schwarz in schwarz drucken; in diesem Fall sollten Sie vorher !TIMEOUT,0 setzen).

Falls während einer Dunkelperiode INK- oder BORDER-Kommandos gegeben werden (von einem Programm aus), dann werden die neuen Werte einfach in den Zwischenspeicher geschrieben und beim späteren Hellschalten von dort geholt. Das Kommando wirkt also verdeckt.

Bekanntlich kann am CPC jede Ink als blinkend definiert werden. Tatsächlich aber blinken ständig alle Inks. Normalerweise sind aber beide Blinkfarben gleich, so daß das Blinken nicht auffällt. Nun führt aber die Firmware die Änderung einer Ink nicht sofort durch, sondern erst beim nächsten Blinken. Falls Sie daher die Blinkperiode (mit SPEED INK) verlängert haben, kann sich auch das Abdunkeln bzw. Hellschalten des Bildschirms entsprechend verzögern.

Während Kassettenoperationen ist TIMEOUT lahmgelegt, weil dabei der interne Zeitgeber stillsteht.

## Für den Assemblerprogrammierer

TIMEOUT arbeitet mit zwei Ereignissen, EV1 und EV2 genannt. Ereignisse sind das Maschinencodeäquivalent zu den BASIC-Befehlen EVERY und AFTER.

EV1 wird einmal pro Sekunde aufgerufen und zählt einen Zähler herab. Wenn dieser auf Null kommt, wird der Bildschirm verdunkelt, EV1 abgestellt und EV2 gestartet.

EV« läuft immer, wenn der Bildschirm dunkel ist. Die Routine wird zehnmal je Sekunde aufgerufen und fragt die SHIFT- und CTRL-Tasten ab, da diese Tasten von den normalen Firmwarefunktionen KM WAIT CHAR etc. nicht erfaßt werden. Falls eine gedrückt ist, wird der Bildschirm hell geschaltet, EV2 abgestellt und EV1 gestartet.

Zur automatischen Aufhellung werden die Firmwareroutinen KM WAIT CHAR (BB06), KM READ CHAR (BB09), KM WAIT KEY (BB18), KM READ KEY (BB1B) sowie IND TXT WRITE CHAR (BDD3) auf Ersatzroutinen umgeleitet. Die Ersatzroutinen führen die Originalroutine der Firmware durch, hellen den Bildschirm auf, stellen EV2 ab und schalten EV1 ein.

Wenn bereits bei der Initialisierung von TIMEOUT eine Ersatzroutine für eine dieser fünf Funktionen vorliegt, dann wird statt der Originalroutine diese aufgerufen.

Damit INK- und BORDER-Befehle auch im abgedunkelten Zustand korrekt durchgeführt werden, sind die Firmwareroutinen SCR SET INK (BC32) und SCR SET BORDER (BC38) ebenfalls umgeleitet. Die Ersatzroutine prüft, ob der Bildschirm gerade dunkel oder hell ist, und führt je nach dem Ergebnis die Original-Firmewareroutine aus oder schreibt die neuen INK-Werte in den Zwischenspeicher, der beim Abdunkeln angelegt wurde. Beim (irgendwann folgenden) Aufhellen werden daher automatisch die neuen Werte verwendet.

Die Firmwareroutinen SCR GET INK und SCR GET BORDER sind nicht auf entsprechende Art umgeleitet. Dadurch liefern sie während einer Abdunklung falsche Werte. Die Implementierung ist zwar leicht, kostet aber Speicherplatz und ist meines Erachtens unnötig, weil es derartige Funktionen, etwa eine Farbe=INK(2), in BASIC ohnehin nicht gibt.

Sie dürfen vor der Initialisierung von TIMEOUT keine Ersatzroutinen für diese vier Firmwarefunktionen eingestellt haben. TIMEOUT geht nämlich davon aus, daß diese Routinen im unteren ROM liegen, was bei einem Ersatz nicht der Fall sein wird.

DARK dunkelt den Bildschirm ab und schaltet EV2 ein sowie EV1 aus.

NORMAL hellt den Bildschirm auf und schaltet EV1 ein sowie EV2 aus.

Bei jedem »Einschalten« von EV1 wird zunächst der Zähler auf den Anlaufwert gesetzt (der mit; TIMEOUT,n übergeben wurde). Falls dieser nicht 0 ist, wird EV1 danach eingeschaltet. Wenn er 0 ist, also TIMEOUT abgestellt ist, wird EV1 ausgeschaltet.

(G. Beran)

```
00,00,E1,F1,3C,FE,10,38,E6
340 DATA E5,CD,00,00,E1,70,23,71,01,00,00, [3371]
CD,00,00,D9,4A,ED,49,D9,FB
350 DATA C9,AF,CD,75,A1,F5,46,23,4E,23,E5, [3127]
CD,00,00,E1,F1,3C,FE,10,38
360 DATA F0,46,23,4E,18,DD,CD,95,A0,F5,C5, [4020]
D5,E5,F3,CD,44,A1,28,03,CD
370 DATA B9,A1,CD,3F,A1,CD,18,A1,FB,E1,D1, [3442]
C1,F1,37,C9,CD,9B,A0,18,E1
380 DATA CD,98,A0,D0,18,DB,CD,92,A0,18,D6, [4005]
CD,9E,A0,18,F3,F3,CD,44,A1
390 DATA CA,A1,A0,21,87,A0,70,23,71,FB,C9, [3072]
F3,57,CD,44,A1,7A,CA,A4,A0
400 DATA 21,67,A0,E6,0F,87,5F,16,00,19,18, [1240]
E6
410 RESTORE 430:n=64:'Eintragen der ab-     [8838]
  soluten Adressen. Beim Abtippen Auf-
```

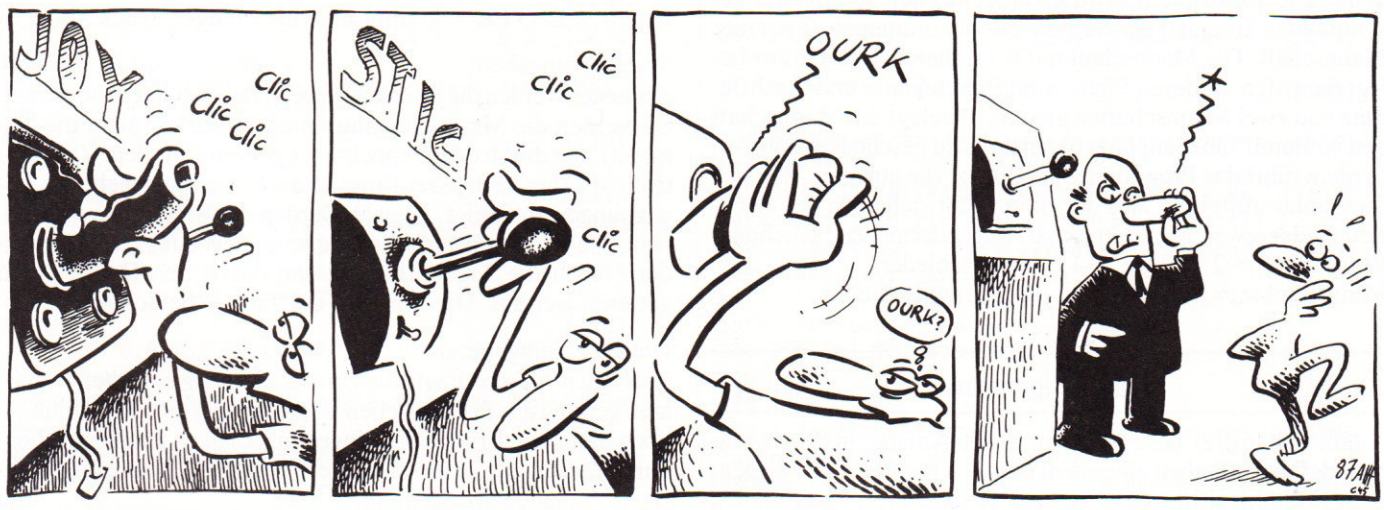
```
passen auf Abfolge von Hex- und De-     zi
malzahlen.
420 FOR i=1 TO n:READ ad,w:ad=ad+h:w=w+h+1 [4041]
:hi=INT(w/256):lo=w-hi*256:POKE ad,lo:POKE
 ad+1,hi:NEXT
430 DATA 4,&92,7,&61,28,&6b,31,&83,64,&61, [3505]
67,&75,90,&89,93,&d3,118,&1fa
440 DATA 120,&1d2,122,&1f4,124,&1ef,126,&1 [4407]
ff,128,&204,130,&213,132,&1af
450 DATA 134,&19a,136,&1c3,138,&18c,140,&1 [3169]
a5,178,&14a,191,&161,212,&c1
460 DATA 222,&144,230,&144,245,&8f,255,&14 [2765]
4,260,&1b9,263,&131,269,&144
470 DATA 274,&1b9,277,&13f,282,&8f,285,&8d [4098]
,296,&a7,307,&a7,313,&b4,321,&b4
480 DATA 327,&91,333,&8d,339,&8d,343,&144, [3033]
347,&185,350,&131,355,&144
```

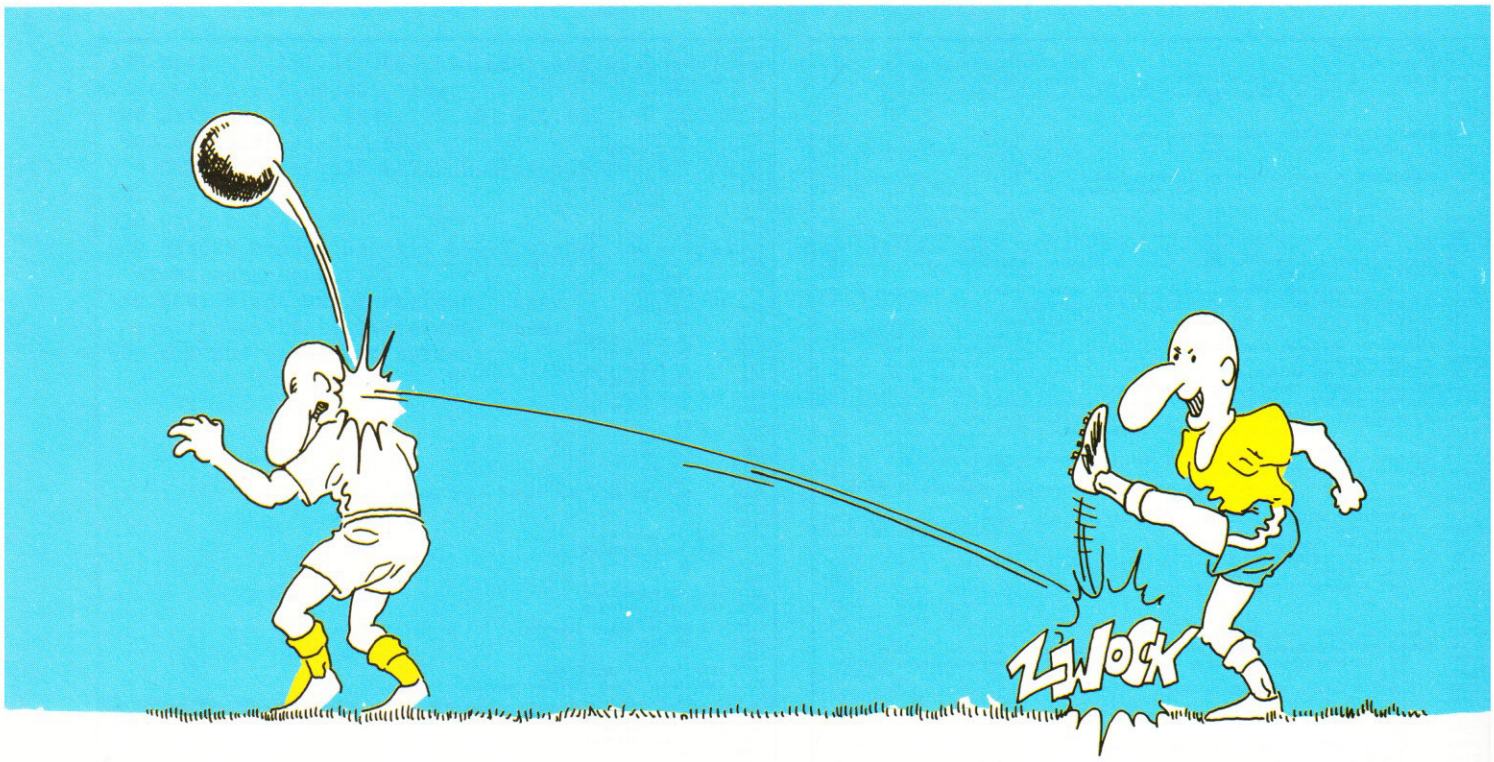
```
490 DATA 375,&91,386,&67,393,&175,444,&175 [2572]
,468,&95,476,&144,481,&1b9
500 DATA 484,&13f,487,&118,497,&9b,502,&98 [2708]
,508,&92,513,&9e,519,&144
510 DATA 522,&a1,525,&87,535,&144,539,&a4, [1706]
542,&67
520 CALL h+1                                [203]
530 MEMORY h+&67:'h+1 bis h+&67 wird        [3243]
  nicht mehr benoetigt
540 NEW:'Dient zur Vermeidung einer         [7728]
  zweifachen Initialisierung. Daher
vor dem RUN unbedingt SAVEen!
```

```
10 ;TIMEOUT. Abdunkeln des Bildschirms bei längerer Untätigkeit
120 ;
130     ORG 0A000H
132 ;Initialisierung:
134 ;Kopieren von 7 Einträgen der Hauptsprungtabelle
140 INIT LD A,7
150     LD DE,JPTAB ;Dort kommen sie hin
160     LD HL,HERTAB ;Hier steht, welche Einträge
170 LOOP1 LD C,(HL)
180     INC HL
190     LD B,(HL)
200     INC HL
210     PUSH HL ;zeigt auf nächsten Eintrag in HERTAB
220     LD L,C
230     LD H,B ;Adresse des Eintrags in der Sprungtabelle
240     LD BC,3
250     LDIR ;3 Byte kopieren
260     POP HL
270     DEC A
280     JR NZ,LOOP1 ;nächster Eintrag
290 ;
300 ;   Einige Routinen werden später direkt im ROM aufgerufen. Herausziehen
310 ;   der Adressen und Eintragen an den Stellen im Code, wo sie später
320 ;   aufgerufen werden.
330     LD B,5
340     LD HL,HERTAB+10 ;Hier steht, welche Routinen
350     LD DE,SETTAB ;enthält die Adressen, von wo sie später aufgerufen werden
360 LOOP2 LD A,(DE)
370     INC DE
380     LD C,A
390     LD A,(DE) ;AC enthält Eintrag von SETTAB
400     INC DE
410     PUSH DE ;zeigt auf nächsten Eintrag in SETTAB
420     LD E,C
430     LD D,A ;= Adresse des CALLs
440     INC DE ; 1 Byte weiter
450     LD C,(HL)
460     INC HL
470     LD A,(HL) ;AC enthält Eintrag von HERTAB
480     INC HL
490     PUSH HL ;zeigt auf nächsten Eintrag in HERTAB
500     LD L,C
510     LD H,A ;= Adresse des Eintrags in der Hauptsprungtabelle
520     INC HL ; 1 Byte weiter
530     LDI ;Low-Byte übertragen
540     INC BC ;LDI hat BC dekrementiert
550     LD A,(HL) ;High-Byte
560     AND 3FH ;ROMstate-Bits löschen
570     LD (DE),A ;Adresse ist übertragen
580     POP HL
590     POP DE
600     DJNZ LOOP2 ;nächster Eintrag
610 ;
620 ;   Eintragen von Indirections:
630     LD B,7 ;Anzahl
640     LD HL,HERTAB ;Dort steht, welche
650     LD DE,INDTAB ;enthält die Adressen der Ersatzroutinen
660 LOOP3 PUSH HL ;zeigt auf Eintrag in HERTAB
670     LD C,(HL)
680     INC HL
690     LD H,(HL)
700     LD L,C ;HL enthält Eintrag von HERTAB
710     LD (HL),0C3H ;JP .. eintragen
720     INC HL
730     LD A,(DE)
740     LD (HL),A ;1.Byte
750     INC HL
760     INC DE
770     LD A,(DE)
780     LD (HL),A ; und 2. Byte kopieren
790     INC DE ;zeigt auf nächsten Eintrag in INDTAB
800     POP HL
810     INC HL
820     INC HL ;zeigt auf nächsten Eintrag in HERTAB
830     DJNZ LOOP3
840 ;
850 ;RSX Einloggen
860     LD HL,BYTE4
870     LD BC,TABLE
880     JP LOGEXT
882 ;Initialisierung der Ereignisblocke mit KL INIT EVENT ist überflüssig, da
884 ;die Blocke bereits die nötigen Informationen enthalten. KL INIT EVENT tut
886 ;auch nichts anderes, als die übergebenen Werte in den Block zu schreiben.
890 ;
891 ;Tabellen:
892 ;Die 1. sieben Einträge von HERTAB geben an, welche Einträge der Haupt-
894 ;sprungtabelle zu kopieren sind. Sie werden kopiert nach JPTAB, JPTAB+3,...
900 HERTAB DEFW TXTIND
910     DEFW WAITCH
920     DEFW READCH
930     DEFW WAITK
940     DEFW READK
950 ;HERTAB+10. Die 5 Einträge ab jetzt geben an, welche Einträge der Haupt-
952 ;sprungtabelle im ROM aufgerufen werden (teilweise überlappung mit oben)
960     DEFW SETBOR
970     DEFW SETINK
980     DEFW SETINK ;2 Mal
990     DEFW GETINK
1000    DEFW GETBOR
1010 ;
1015 ;Adressen der Ersatzroutinen für die 1. sieben Routinen aus HERTAB
1020 INDTAB DEFW IND0
1030     DEFW IND1
1050     DEFW IND2
1060     DEFW IND3
1070     DEFW IND4
1080     DEFW IND5
1090     DEFW IND6
1100 ;
1110 ;Adressen der Stellen im Code, von denen die 5 Routinen ab HERTAB+10
1115 ;aufgerufen werden
1120 SETTAB DEFW SETBO1
1130     DEFW SETIN1
1140     DEFW SETIN2
1150     DEFW GETIN1
1160     DEFW GETBO1
1170 ;
1180 COUNT DEFW 0 ;Enthält Anzahl der Sekunden bis zu einem Abdunkeln
1190 BYTE4 EQU COUNT-4 ;4-Byte Bereich für RSX
1200 NOMTAB EQU BYTE4-34 ;enthält nach Abdunkeln die Werte für 16 Inks +
1210 ;                    Border (34 Byte  = (16+1)*2  ;2 Farben je Ink)
1220 ;    (beides liegt parallel zu INIT f.)
1230 ;
1240 HIGHCT DEFW 0 ;enth. Anlaufwert, auf den (COUNT) nach Ablauf gesetzt wird
1250 DKFLAG DEFB 0 ;ist 0, wenn Bildschirm Hell; ist FF, wenn dunkel
1260 JPTAB DEFS 21 ; 7*3
1265 ;
1270 EV1BLK DEFS 6 ;braucht die Firmware zur Ereignisverwaltung
1280     DEFS 2
1290     DEFB 0
1300     DEFB 81H ; = 1000 0001
1310     DEFW EV1RT
1320     DEFB 0
1325 ;
1330 EV2BLK DEFS 8
1340     DEFB 0
1350     DEFB 81H ;s.o.
1360     DEFW EV2RT
1370     DEFB 0
1375 ;
1377 ;RSX- Namenstabelle
1380 NTABLE DEFM "TIMEOU"
1390     DEFB 0D4H ;T
1400     DEFM "DAR"
1410     DEFB 0CBH ;K
1420     DEFM "NORMA"
1430     DEFW 0CCH ;L    defW, daher Abschluß mit 00
1432 ;
1434 ;RSX - Befehlstabelle
1440 TABLE DEFW NTABLE
1450     JR TIMOUT ;Sprung zur 1. Routine
1460     NOP ;Lückenfüller, weil die Firmware 3 Byte lange Sprünge erwartet
1470     JR DARK
1480     NOP
1485 ;Beginn der 3. Routine
1490 NORMAL DI ;zwischen FLGTST und NORMEX soll kein TIMEOUT "zuschlagen"
1500     CALL FLGTST ;Bildschirm hell?
1510     JR Z,EV1ON ; Ja
1520     JR W9
1530 ;
1540 DARK DI ;vgl. NORMAL
1550     CALL FLGTST
1560     JR NZ,EV2ON ;ist dunkel
1570     JR W2
1580 ;
1590 TIMOUT DEC A ; 1 Parameter?
1600     RET NZ ;nein
1610     LD A,(IX+0)
1620     LD B,(IX+1) ;Parameter in AB
1630     LD HL,HIGHCT
1640     LD (HL),A
1650     INC HL
1660     LD (HL),B ;Eintragen
1670     OR B ;0?
1680     JR NZ, W3 ;Nein
1690     DI ;vgl. NORMAL
1700     CALL FLGTST
1710     JR Z, W4 ;ist hell
1720     CALL NORMEX ;hell schalten
1730 W4  CALL EV1OFF ;weil der Parameter 0 war
1740 W5  JR EV2OFF
1750 W3  DI
1760     CALL FLGTST
1770     JR Z,W6 ;ist hell
1780 W9  CALL NORMEX
1790 W6  CALL EV2OFF
1800     JR EV1ON ;Parameter war <>0
1810 ;
1812 ; schaltet EV1 ein
1820 EV1ON LD HL,HIGHCT ; Anlaufwert übertragen
1830     LD DE,COUNT
1840     LD A,(HL)
1850     LDI ; 1.Byte. HL zeigt aufs 2.
1860     OR (HL) ; 0?
1870     JR Z,EV1OFF ;ja; EV1 sicherheitshalber abschalten
1880     LDI ; 2. Byte übertragen
1890     LD HL,EV1BLK
1900     LD DE,50D ;1 mal je Sekunde
1910 W7  LD B,D
1920     LD C,E
1930     JP ADDTIC ; Ereignis eintragen
1940 ;
1945 ; EV1 ausschalten
1950 EV1OFF LD HL,EV1BLK
```

```
1960 W8  JP DELTIC
1970 ;
1980 EV2ON LD HL,EV2BLK ;EV2 einschalten
1990     LD DE,5 ;alle 5/50 = 1/10 sec
2000     JR W7
2010 ;
2020 EV2OFF LD HL,EV2BLK ;EV2 ausschalten
2030     JR W8
2040 ;
2050 FLGTST XOR A ;Testet Dkflag. Z, wenn hell
2060 ;                          NZ, wenn dunkel
2070     LD HL,DKFLAG
2080     CP (HL)
2090     RET
2100 ;
2105 ;Ereignisroutine von EV1. Wird 1 mal je Sekunde aufgerufen
2110 EV1RT LD BC,(COUNT)
2120     CPI ;Ersatz für DEC BC. (DEC BC setzt keine Flags)
2130     LD (COUNT),BC
2140     RET PE ;Zähler <> 0
2150 ;
2155 ;Zähler=0; Bildschirm abdunkeln
2160     CALL FLGTST
2170     RET NZ ;ist eh dunkel
2180 W2  CALL DARKEX ;Abdunklung durchführen
2190     CALL EV1OFF
2200     JR EV2ON ;  DI wie bei DARK und NORMAL ist überflüssig, weil die
2210 ;               Routine sich nicht selbst unterbricht
2220 ;
2225 ;Ereignisroutine von EV2. wird 10 mal je Sekunde aufgerufen.
2230 EV2RT CALL FLGTST
2240     JR Z,EV2OFF ;ist hell. EV2 abschalten, obwohl es abgeschaltet sein
2245 ;                   müßte.
2250     LD A,21D ;SHIFT
2260     CALL TESTK
2270     JR NZ,W9 ;gedrückt
2280     LD A,23D ;CTRL
2290     CALL TESTK
2300     RET Z ;auch nicht gedrückt
2310     JR W9
2320 ;
2325 ;Teil von DARKEX und NORMEX
2330 ROM LD (DKFLAG),A ;Bildschirmstatus abspeichern
2340     DI ;Unterbrechungen müssen bis auf weiteres gesperrt bleiben.
2350     EXX
2360     LD D,C ;alten ROMstate sichern
2370     RES 2,C ;L-ROM freischalten
2380     OUT (C),C
2390     EXX
2400     LD HL,NOMTAB ;Tabelle mit den Ink-Werten
2410     XOR A ;Ink Nr. 0
2420     RET
2430 ;
2440 ;Abdunkeln des Bildschirms
2450 DARKEX LD A,0FFH ; neuer Status
2460     CALL ROM
2470 LOOP4 PUSH AF ;INK Nr.
2480     PUSH HL
2490 GETIN1 CALL 0 ;Get Ink (hier wird von INIT die Rom-Adresse von SCR GET INK
2495 ;                  eingetragen)
2500     POP HL
2510     LD (HL),B ;1.Farbe
2520     INC HL
2530     LD (HL),C ;2.Farbe
2540     INC HL
2550     POP AF ;Ink Nr.
2560     PUSH AF
2570     PUSH HL
2580     LD BC,0 ;Ink auf schwarz setzen
2590 SETIN1 CALL 0 ;set ink (s.o.)
2600     POP HL
2610     POP AF
2620     INC A ;nächste Ink
2630     CP 10H ;schon fertig?
2640     JR C,LOOP4 ;nein
2650     PUSH HL
2660 GETB01 CALL 0 ;Get Border
2670     POP HL
2680     LD (HL),B
2690     INC HL
2700     LD (HL),C
2710     LD BC,0    ;s.o.
2720 SETB01 CALL 0 ;Set Border
2730     EXX ;ROMstate wiederherstellen
2740     LD C,D ;alter ROMstate
2750     OUT (C),C
2760     EXX
2770     EI ;erst jetzt sind Unterbrechungen wieder zulässig
2780     RET
2790 ;
2800 ;schaltet den Bildschirm hell
2810 NORMEX XOR A ;vgl. DARKEX
2820     CALL ROM
2830 LOOP5 PUSH AF
2840     LD B,(HL)
2850     INC HL
2860     LD C,(HL)
```

```
2870     INC HL
2880     PUSH HL
2890 SETIN2 CALL 0 ;Set Ink
2900     POP HL
2910     POP AF
2920     INC A
2930     CP 10H
2940     JR C,LOOP5
2950     LD B,(HL)
2960     INC HL
2970     LD C,(HL)
2980     JR SETB01
2990 ;
2995 ; Ersatzroutinen:
2997 ; für KM Wait Char
3000 IND1 CALL JPTAB+3 ;Original Firmware-Routine
3010 W10 PUSH AF
3020     PUSH BC
3030     PUSH DE
3040     PUSH HL
3050     DI ;vgl. NORMAL
3060     CALL FLGTST
3070     JR Z,W11 ;ist hell
3080     CALL NORMEX
3090 W11 CALL EV2OFF
3100     CALL EV1ON
3110     EI
3120     POP HL
3130     POP DE
3140     POP BC
3150     POP AF
3160     SCF ;KM Wait Char kehrt mit C zurück
3170     RET
3180 ;
3185 ;für KM Wait Key
3190 IND3 CALL JPTAB+9
3200     JR W10
3210 ;
3215 ;für KM Read Char
3220 IND2 CALL JPTAB+6
3230 W12 RET NC ;kein Zeichen
3240     JR W10
3250 ;
3255 ;für IND TXT Write Char
3260 IND0 CALL JPTAB ;Zeichen ausgeben
3270     JR W10
3280 ;
3285 ;für KM Read Key
3290 IND4 CALL JPTAB+12
3300     JR W12
3310 ;
3315 ;für SCR Set Border
3320 IND5 DI
3330     CALL FLGTST
3340     JP Z,JPTAB+15 ;ist hell; SCR Set Border
3350     LD HL,NOMTAB+32
3360 W13 LD (HL),B ;1.Farbe
3370     INC HL
3380     LD (HL),C ;2.Farbe
3390     EI
3400     RET
3410 ;
3415 ;für SCR Set Ink
3420 IND6 DI
3430     LD D,A ;Ink Nr.
3440     CALL FLGTST
3450     LD A,D
3460     JP Z,JPTAB+18 ;SCR Set Ink
3470     LD HL,NOMTAB
3480     AND 0FH ;auf gültigen Wert bringen
3490     ADD A,A ;*2
3500     LD E,A
3510     LD D,0
3520     ADD HL,DE ;+NOMTAB
3530     JR W13 ;eintragen
3540 ;
3545 ;verwendete Firmware-Routinen
3550 LOGEXT EQU 0BCD1H ;KL Log Extension
3560 WAITCH EQU 0BB06H ;KM Wait Char
3570 READCH EQU 0BB09H ;KM Read Char
3580 WAITK  EQU 0BB18H ;KM Wait Key
3590 READK  EQU 0BB1BH ;KM Read Key
3600 TESTK  EQU 0BB1EH ;KM Test Key
3610 SETBOR EQU 0BC38H ;SCR Set Border
3620 GETBOR EQU 0BC3BH ;SCR Get Border
3630 SETINK EQU 0BC32H ;SCR Set Ink
3640 GETINK EQU 0BC35H ;SCR Get Ink
3650 ADDTIC EQU 0BCE9H ;KL Add Ticker
3660 DELTIC EQU 0BCECH ;KL Del Ticker
3670 TXTIND EQU 0BDD3H ;IND Txt Write Char
3680 ;
3690     END INIT
```






# Bundesliga

**Bei diesem Programm dreht es sich um die Berechnung der Tabelle der 1. Fußballbundesliga während oder nach dem Spieltag.**

## Die Tabellenberechnung

Bei jedem Spiel werden 2 Plus- und 2 Minuspunkte vergeben. Der Gewinner des Spiels erhält 2 Pluspunkte, der Verlierer 2 Minuspunkte. Bei einem Unentschieden teilen sich die Mannschaften die Punkte: jede erhält einen Plus- und einen Minuspunkt.

Für den Tabellenplatz ist in erster Linie das Punktekonto entscheidend. Je mehr Pluspunkte eine Mannschaft hat, um so mehr Siege kann sie verzeichnen und um so höher steigt sie in der Tabelle. Haben zwei Mannschaften den gleichen Punktestand(Plus- und Minuspunkte der beiden Mannschaften sind gleich), so ist für den Tabellenplatz die Tordifferenz entscheidend. Die Tordifferenz wird gebildet aus der Summe der geschossenen Tore und der Summe der bekommenen Tore einer Mannschaft. Die Mannschaft mit der höheren Tordifferenz belegt dann den vorderen Platz. Sind Punktekonto und Tordifferenz von zwei Mannschaften gleich, so belegt die Mannschaft den vorderen Tabellenplatz, die mehr Tore geschossen hat. Außerdem führt das Programm eine Bilanz, die aufzeigt, wie sich die Punkte aufteilen. Aus der Bilanz läßt sich auch der Punktestand der jeweiligen Mannschaft folgendermaßen berechnen:

Pluspunkte = 2 * Siege + 1 * Unentschieden
Minuspunkte = 2 * Niederl. + 1 * Unentschieden

## Das Programm

...läßt sich in drei Teile gliedern: In den Anfang, in dem Variablen definiert, Farben eingestellt und Zeichen definiert werden, in das Hauptmenü, von wo aus die 7 verschiedenen Optionen aufgerufen werden, und in die Unterprogramme.

Hinweis: Wer das Programm ein zweites Mal durch »Run« starten möchte wird Pech haben, denn der Computer protestiert gegen das SYMBOL AFTER in der Zeile 20. Dies liegt an den Zeilen 240-260 die die Ausgabe von Daten auf Cassette beschleunigen. Besitzer von Diskettenlaufwerken können sich diese Zeilen sparen. Wer das Programm also ein zweites Mal starten will, sollte »RUN 20« oder »GOTO 500« eingeben, wobei bei dem GOTO sämtliche Daten erhalten bleiben.

## Die 7 Optionen

Mannschaftsnamen eingeben:
jede Mannschaft besitzt eine Nummer, damit beim Abfragen der Spielpaarungen nicht so viel eingegeben werden muß. Jeder Nummer kann nun ein Name zugeordnet werden, der dann auch in der Tabelle auftaucht. Die Namen können jederzeit verändert werden. Dies hat keinen Einfluß auf das Programm. Beim Eingeben der Mannschaftsnamen ist folgende Form einzuhalten:

Nummer<:>Name<ENTER>

Durch <ENTER> kommt man ins Menue zurück.

Spieltag eingeben:
Als erstes werden die Spielpaarungen (Begegnungen) abgefragt. Es werden die Mannschaftsnummern erwartet (nicht die Namen!!), die durch den Doppelpunkt getrennt werden. Zur Kontrolle werden nach jeder Eingabe die bisher eingegebenen Begegnungen angezeigt. Danach werden die Ergebnisse eingegeben. Diese können anschließend verändert oder gelöscht werden. Auch die Ergebnisse müssen durch den Doppelpunkt getrennt werden. Danach wird die Tabelle berechnet.

Die Einzeleingabe:
ist ähnlich aufgebaut wie die Option »Spieltag eingeben«. Auch hier geht es um das Eingeben von Ergebnissen. Man muß jedoch nicht alle Ergebnisse eingeben, sondern kann zwischendurch die Tabelle berechnen, sie ausdrucken, sowie die bisherigen Eingaben abspeichern.

Daten Laden / Abspeichern:
Beim Abspeichern werden die Daten der Tabelle, sowie der momentane Stand der Einzeleingabe abgespeichert. Anhand des Punktekontos der Mannschaft Nr. 1 wird der Spieltag ermittelt. Daraus ergibt sich der Dateiname: »<Spieltag>.SPT«. Beim Laden wird nach dem Spieltag gefragt, damit die richtige Datei geladen wird. Dabei ist am 18. Spieltag eine 17 einzugeben, da der 17. Spieltag der zuletzt gespeicherte sein muß.

Tabelle ausgeben:
Die Tabelle kann auf dem Bildschirm und auf dem Drucker ausgegeben werden. Es werden Platz, Name, Punkte, Tore, Tordifferenz und Bilanz jeder Mannschaft ausgedruckt.

Werte der Tabelle korrigieren:
Hier können die Zahlenwerte der Tabelle verändert bzw. neu eingegeben werden. Punkte und Tore sind wie üblich durch den Doppelpunkt, die Werte der Bilanz jedoch durch das Komma zu trennen.
Wer das Programm nach eigenen Vorstellungen erweitern oder verändern will kann das gerne tun. Er kann dabei auf einige Unterprogramme zurückgreifen, die ihm das Leben leichter machen:
3700 Ja-Nein-Abfrage
Die Tasten <J> und <N> werden solange abgefragt, bis eine von ihnen gedrückt wird. Die Variable jn erhält dann den Wert 0 für »Ja« und den Wert 1 für »Nein«.

4000 Windows definieren
Der Bildschirm wird in 3 Windows eingeteilt: #2 ist die erste Zeile (für eine Überschrift) #1 sind die Zeilen 2–11 (zum Ausdrucken von Ergebnissen, Mannschaftsnamen o.ä.) #0 sind die Zeilen 12–25

4100 String zerpflücken
Nach Aufruf des Unterprogrammes erwartert der Computer einen String, der anschließend in den Teil vor und in den Teil nach dem ersten Doppelpunkt geteilt wird. Ist bei Einsprung die Variable se<>0, so läßt er auch Strings als Eingabe zu, die er als b$ und d$ zurückgibt. Anderenfalls akzeptiert er nur Zahlen, die er als e% und z% zurückgibt.
Das Hauptmenue kann durch beliebig viele neue Optionen erweitert werden.

Hinweis zum Eintippen des Programms
Zeilen die mit »PRINT #2,« anfangen sind Überschriften. Sie können invers dargestellt werden indem man gleich nach dem ersten und vor dem letzten Anführungszeichen ein Steuerzeichen einfügt, das man über die Tastenkombination <CTRL>+<X>erhält.

## Variablenliste

Dimensionierte Variablen:

| | |
|---|---|
| Name$ | Name einer Mannschaft |
| ppu | Pluspunkte einer M. |
| mpu | Minuspunkte einer M. |
| gto | Geschossene Tore einer Mannschaft |
| bto | Bekommene Tore einer M. |
| bs | Anzahl der Siege einer Mannschaft (für die Bilanz) |
| bu | Anz. der Unentschieden einer Mannschaft |
| bn | Anz. der Niederlagen einer Mannschaft |
| platz | Nummer einer Mannschaft die einen bestimmten Tabellenplatz belegt. platz(1) enthält z.B. die Nummer der Mannsch. die den ersten Platz in der Tabelle belegt. |
| epu | Bezieht sich auf die Einzeleingabe. epu(1) ist 1, wenn für die Mannschaft Nr. 1 ein Ergebnis eingegeben wurde. |
| kon | Hängt mit der Eingabe der Spielpaarungen zusammen. kon (1) enthält die Nummer der Mannschaft die im ersten Spiel als erste eingegeben wurde, kon (2) diejenige, die als zweite eingegeben wurde |
| tg+tb | enthalten die Ergebnisse der Spiele. tg (kon (1)) wären die Tore die die Mannschaft Nr. 1 in ihrem Spiel geschossen hätte. tg steht für die geschossenen, tb für die bekommenen Tore. |

Wichtige nicht dimensionierte Variablen:

| | |
|---|---|
| ez% | = 1, wenn die Spielpaarungen eingegeben wurden. =0, wenn sie noch nicht eingegeben wurden. |
| mnr% | Mannschaftsnummer |

(Peter Kannowade)

**für 464-664-6128**

```
10 '                                          [117]
20 '                                          [117]
30 '     P R O G R A M M      Z U R    B E  [2694]
R E C H N U N G   DER
40 'T A B E L L E     D E R   1 . F U ~ B  [1212]
A L L B U N D E S L I G A
50 '                                          [117]
60 '                                          [117]
70 SYMBOL AFTER 90                            [1214]
80 SYMBOL 123,72,0,120,12,124,204,118         [2137]
90 SYMBOL 124,36,0,60,102,102,102,60,0        [1575]
100 SYMBOL 125,36,0,102,102,102,102,62,0      [2369]
110 SYMBOL 126,28,34,98,110,98,98,108,96      [2459]
120 MODE 2                                    [513]
130 'INK 0,0:BORDER 0:INK 1,13:'Nur f}r Fa    [2376]
rbmonitore
140 BORDER 0:INK 1,26:PAPER 0:'Nur f}r Gr}    [3025]
nmomitore
150 PRINT"   Bundesligaprogramm":PRINT        [2449]
160 PRINT " geschrieben von :                 [1300]
170 PRINT:PRINT                               [743]
180 PRINT"********************************    [1876]
**********"
190 PRINT"**                                  [1579]
        **"
200 PRINT"**            Peter Kannowade       [3514]
        **"
210 PRINT"**                                  [1579]
        **"
220 PRINT"********************************    [1876]
**********"
230 PRINT:PRINT:PRINT "Herzlichen Dank an     [7824]
Markus Ortloff f}r seine Unterst}tzung !!"
240 '  Achtung  !!!!!   Programm kann nach    [6763]
 dem Laden nur einmal durch "Run" ge-
            startet werden .  Danach mit "
goto 500" oder "run 20" beginnen .
250 FOR a=1 TO 3000:NEXT                      [1580]
260 '                                         [117]
270 '                                         [117]
280 ' deutsche tastatur   : Umlaute durch     [3584]
contol+a,o oder u
290 '                          sz durch       [518]
```

```
contol+"
300 KEY DEF 69,1,97,65,123                    [1068]
310 KEY DEF 34,1,111,79,124                   [1227]
320 KEY DEF 42,1,117,85,125                   [1767]
330 OPENOUT "dummy"                           [987]
340 MEMORY HIMEM-1                            [259]
350 CLOSEOUT                                  [902]
360 DIM name$(18),ppu(18),mpu(18),gto(18),    [7430]
bto(18),bs(18),bu(18),bn(18),platz(18),epu
(18),kon(18),tg(18),tb(18)
370 FOR a=1 TO 18                             [950]
380 '   Startwerte fuer Tabelle               [781]
390 platz(a)=a                                [353]
400 NEXT                                      [350]
410 KEY 138,":":'Punkttaste auf dem Ziffer    [3716]
nblock mit dem Doppelpunkt belegen
420 '                                         [117]
430 '}berschriften k|nen invers dargestell    [10783]
t werden , indem man nach dem An-    f}hr
ungszeichen am Anfang und vor dem am Schlu
ss das Steuerzeichen  ctrl+X      einf}gt
.
440 '                                         [117]
450 '                                         [117]
460 '                                         [117]
470 '       MENUE                             [1057]
480 '                                         [117]
490 '                                         [117]
500 MODE 2                                    [513]
510 PRINT CHR$(24)"                           [6737]
 Fu~ball-Tabellen Berechnung
            "CHR$(24)
520 LOCATE 1,4                                [620]
530 PRINT:PRINT "Tabelle ausgeben -------- [3558]
------------  T"
540 PRINT:PRINT "Einzeleingabe ----------- [3725]
------------  E"
550 PRINT:PRINT "Mannschaftsnamen eingeben [3141]
 ------------  M"
560 PRINT:PRINT "Spieltag eingeben ------- [3505]
------------  S"
570 PRINT:PRINT "Daten abspeichern ------- [3460]
------------  A"
580 PRINT:PRINT "Daten laden ------------- [3465]
------------  L"
590 PRINT:PRINT "Werte der Tabelle korrigi [4432]
eren --------- K"
600 a$=INKEY$:a$=UPPER$(a$)                   [1404]
610 IF a$="" THEN GOTO 600                    [699]
620 IF a$="T" THEN GOSUB 780                  [1849]
630 IF a$="E" THEN GOSUB 1020                 [1637]
640 IF a$="M" THEN GOSUB 1310                 [1416]
650 IF a$="S" THEN GOSUB 1490                 [1162]
660 IF a$="A" THEN GOSUB 2360                 [1253]
670 IF a$="K" THEN GOSUB 3760                 [1148]
680 IF a$="L" THEN GOSUB 2550                 [1909]
690 GOSUB 3300                                [965]
700 GOTO 500                                  [401]
710 '                                         [117]
720 '                                         [117]
730 '  AB HIER UNTERPROGRAMME                 [1757]
740 '                                         [117]
750 '                                         [117]
760 '                                         [117]
770 '                                         [117]
780 '                                         [117]
790 '                                         [117]
800 '   Tabelle drucken                       [1023]
810 '                                         [117]
820 '                                         [117]
830 MODE 2                                    [513]
840 GOSUB 3300:' Tastenpuffer loeschen        [2309]
850 INPUT "DRUCKEN AUF DRUCKER (D)  ODER A [5169]
UF BILDSCHIRM  (B) ";e$:e$=UPPER$(e$):IF e
$="D" THEN aggnr=8
860 IF e$="B" THEN aggnr=0                    [361]
870 IF e$<>"D" AND e$<>"B" THEN RETURN        [1535]
880 CLS #0                                    [374]
890 IF aggnr=8 THEN PRINT #8,CHR$(14);"Tab [6396]
elle des ";(ppu(1)+mpu(1))/2;".Spieltags"
900 PRINT "Tabelle des ";(ppu(1)+mpu(1))/2 [3192]
;".Spieltags
910 a$=""                                     [388]
920 PRINT "Platz Mannschaft                   [5545]
          +P   -P    GT    BT        Si  U
n  Ni"
930 IF aggnr=8 THEN PRINT #8,"Platz  Manns [6039]
chaftsname                        Punkte
    Tore    diff.     Si   Un   NI"
940 FOR a=1 TO 18                             [950]
950 z=platz(a)                                [1040]
960 IF aggnr=8 THEN PRINT #8,USING "##\ \\ [11199]
                                   \\ \##\
\##\   \###\ \###\ \+##\    \##\ \##\ \##"
;a,a$,name$(z),a$,ppu(z)," : ",mpu(z),a$,g
to(z)," : ",bto(z),a$,gto(z)-bto(z),a$,bs(
z),a$,bu(z),a$,bn(z)
970 PRINT USING "##\ \\                       [10100]
          \\ \##\ \##\ \###\ \###\\+##\
\##\\##\\##";a,a$,name$(z),a$,ppu(z)," : "
,mpu(z),a$,gto(z)," : ",bto(z),a$,gto(z)-b
to(z),a$,bs(z),a$,bu(z),a$,bn(z)
980 NEXT                                      [350]
990 CALL &BB18                                [389]
1000 RETURN                                   [555]
1010 '                                        [117]
1020 '                                        [117]
1030 '  Einzeleingabe                         [1648]
1040 '                                        [117]
1050 '                                        [117]
1060 MODE 2                                   [513]
1070 GOSUB 2760                               [899]
1080 GOSUB 3300                               [965]
1090 PRINT #2,CHR$(24)"     Einzeleingabe     [1575]
     "CHR$(24)
1100 PRINT CHR$(24)"  Eingabe der Begegnun [4195]
gen  "CHR$(24)
1110 IF ez%=1 THEN GOTO 1140                  [1277]
1120 ez%=1:ERASE tg,tb,kon,epu:DIM tg(18), [2770]
tb(18),kon(18),epu(18)
1130 GOSUB 1600                               [859]
1140 CLS                                      [91]
1150 PRINT:PRINT " Ergebnis einer Begegnun [4228]
g eingeben   ------- E"
1160 PRINT:PRINT " Momentanen Tabellenstan [5167]
d berechnen  ------- B"
1170 PRINT:PRINT " Tabelle ausgeben        [3948]
             ------- T"
1180 PRINT:PRINT " R}ckkehr ins Menue      [3903]
             ------- R"
1190 PRINT:PRINT " Neue Spielpaarungen ein [5122]
geben        ------- S"
1200 IF INKEY(58)=0 THEN up=1:CLS:CLS #1:G [3278]
OSUB 2970
1210 IF INKEY(54)=0 THEN up=1:GOSUB 2070   [2194]
1220 IF INKEY(51)=0 THEN up=1:GOSUB 780    [894]
1230 IF INKEY(60)=0 THEN up=1:ez%=0:GOTO 1 [1358]
020
1240 IF INKEY(50)=0 THEN RETURN            [840]
1250 IF up=0 THEN GOTO 1200                [1228]
1260 up=0                                  [212]
1270 GOSUB 3450                            [1039]
1280 GOSUB 3330                            [975]
1290 PRINT #2,CHR$(24)"     Einzeleingabe  [1575]
     "CHR$(24)
1300 GOTO 1140                             [311]
```

```
1310 '                                          [117]
1320 '                                          [117]
1330 '                                          [117]
1340 '     Mannschaftsnamen eingeben            [3004]
1350 '                                          [117]
1360 '                                          [117]
1370 GOSUB 3300                                 [965]
1380 CLS                                        [91]
1390 GOSUB 2760:' Mannschaftsnamen ausgebe [3908]
n
1400 PRINT #2,CHR$(24)"    Mannschaftsname [3106]
n {ndern    "CHR$(24)
1410 PRINT "Geben Sie die Mannschaftsnamen [6451]
 bitte in der Form NUMMER:NAME [Enter] ein
"
1420 PRINT "Zum Menue zurueck mit [ENTER]" [2807]
1430 se=1                                       [283]
1440 GOSUB 3550                                 [979]
1450 IF le=0 THEN RETURN                        [1251]
1460 IF e%>18 THEN GOTO 1310                    [464]
1470 name$(e%)=d$                               [585]
1480 GOTO 1310                                  [436]
1490 '                                          [117]
1500 '                                          [117]
1510 ' Spieltag eingeben                        [391]
1520 '                                          [117]
1530 '                                          [117]
1540 ERASE kon:DIM kon(18):ez%=0                [775]
1550 MODE 2                                     [513]
1560 GOSUB 3300                                 [965]
1570 GOSUB 2760:' Mannschaftsnamen darstel [3468]
len
1580 PRINT #2,CHR$(24)"    Spieltag eingeb [6329]
en    ";CHR$(24),"              ";CHR$(24)
;"    Eingeben der Begegnungen    "CHR$(24
)
1590 PRINT "Geben Sie alle Spielpaarungen  [11462]
ein,indem Sie die oben angegebenen Nummern
 verwen- den . Trennen Sie bitte durch den
 Doppelpunkt .":CALL &BB18
1600 FOR a=1 TO 17 STEP 2                       [965]
1610 GOSUB 1720                                 [819]
1620 GOSUB 3550                                 [979]
1630 IF e%<1 OR e%>18 OR z%<0 OR z%>18 OR  [3508]
e%=z% THEN PRINT "Eingabefehler  !!":GOTO
1620
1640 b=0                                        [312]
1650 b=b+1:IF b=19 THEN GOTO 1680               [1366]
1660 IF kon(b)=e% OR kon(b)=z% THEN PRINT  [12530]
"Eine der Mannschaften wurde schon einmal
eingegeben.Wiederholen Sie bitte die   Ein
gabe":GOTO 1620
1670 GOTO 1650                                  [313]
1680 kon(a)=e%                                  [795]
1690 kon(a+1)=z%                                [668]
1700 NEXT                                       [350]
1710 a=17                                       [535]
1720 CLS                                        [91]
1730 FOR c=1 TO a STEP 2                        [1319]
1740 PRINT USING "\                             [4590]
      \\   \\
        \";name$(kon(c))," :  ",name$(ko
n(c+1))
1750 NEXT                                       [350]
1760 IF kon(18)=0 THEN RETURN                   [875]
1770 PRINT "eingabe richtig  ??"                [1344]
1780 GOSUB 3200:' ja/nein-Abfrage               [1902]
1790 IF jn=1 THEN GOTO 1490                     [907]
1800 '                                          [117]
1810 IF ez%=1 THEN RETURN:'Ende der Mannsc [5053]
hafteseingabe ; wenn ez%=1 , dann
                                R}cksprung
 zur Einzeleingabe
```

```
1820 '                                          [117]
1830 PRINT #2,CHR$(24)"   Spieltag Eingebe [5982]
n    ";CHR$(24);"         ";CHR$(24);"   E
rgebnisse Eingeben   "CHR$(24)
1840 CLS                                        [91]
1850 FOR b=1 TO 17 STEP 2                       [1349]
1860 a=kon(b)                                   [1192]
1870 c=kon(b+1)                                 [1551]
1880 PRINT USING "\                             [4728]
      \\ \\                          \\
\";name$(a),":",name$(c);
1890 GOSUB 3550                                 [979]
1900 a%=e%:b%=z%                                [710]
1910 tg(a)=a%:tb(a)=b%:tg(c)=b%:tb(c)=a%   [2041]
1920 gto(a)=gto(a)+a%                           [978]
1930 gto(c)=gto(c)+b%                           [221]
1940 bto(a)=bto(a)+b%                           [700]
1950 bto(c)=bto(c)+a%                           [1169]
1960 IF a%>b% THEN ppu(a)=ppu(a)+2:mpu(c)= [4584]
mpu(c)+2:bs(a)=bs(a)+1:bn(c)=bn(c)+1
1970 IF a%<b% THEN mpu(a)=mpu(a)+2:ppu(c)= [2658]
ppu(c)+2:bn(a)=bn(a)+1:bs(c)=bs(c)+1
1980 IF a%=b% THEN ppu(a)=ppu(a)+1:ppu(c)= [6250]
ppu(c)+1:mpu(a)=mpu(a)+1:mpu(c)=mpu(c)+1:b
u(a)=bu(a)+1:bu(c)=bu(c)+1
1990 NEXT                                       [350]
2000 FOR a=1 TO 18:epu(a)=1:NEXT                [1469]
2010 CLS                                        [91]
2020 PRINT #2,CHR$(24);"*********               [5123]
          Die Ergebnisse
       **********"CHR$(24)
2030 GOSUB 3330                                 [975]
2040 PRINT "Eingabe richtig ? "                 [2220]
2050 GOSUB 3200                                 [953]
2060 IF jn=1 THEN PRINT #2:GOSUB 2920           [1889]
2070 '                                          [117]
2080 '                                          [117]
2090 '   ordnen                                 [445]
2100 '                                          [117]
2110 '                                          [117]
2120 ' Bubblesort                               [857]
2130 '                                          [117]
2140 CLS #1:CLS                                 [554]
2150 PRINT " Aktuelle Tabelle wird berechn [4026]
et"
2160 PRINT " Bitte warten Sie einen Moment [2875]
"
2170 '                                          [117]
2180 'teil 1 :Nach Punkten                      [1591]
2190 '                                          [117]
2200 FOR b=17 TO 1 STEP -1                      [1651]
2210 FOR a=1 TO b                               [626]
2220 IF ppu(platz(a))<ppu(platz(a+1)) THEN [5295]
 tr=platz(a):platz(a)=platz(a+1):platz(a+1
)=tr
2230 NEXT a,b                                   [355]
2240 '                                          [117]
2250 ' teil 2 :  Bei Punktgleichstand nach [2309]
 Tordifferenz
2260 '                                          [117]
2270 ' und bei gleicher Differenz nach ges [2665]
chossenen Toren
2280 '                                          [117]
2290 FOR b=17 TO 1 STEP -1                      [1651]
2300 FOR a=1 TO b                               [626]
2310 tdiff=(gto(platz(a))-bto(platz(a)))-( [10117]
gto(platz(a+1))-bto(platz(a+1))):IF ppu(pl
atz(a))=ppu(platz(a+1)) AND tdiff<0 THEN t
r=platz(a):platz(a)=platz(a+1):platz(a+1)=
tr
2320 IF tdiff=0 AND ppu(platz(a))=ppu(plat [6780]
z(a+1)) AND gto(platz(a+1))>gto(platz(a))
THEN tr=platz(a):platz(a)=platz(a+1):platz
```

```
(a+1)=tr
2330 NEXT a,b                                   [355]
2340 IF ez%=0 THEN FOR a=1 TO 18:epu(a)=0: [2603]
NEXT
2350 RETURN                                     [555]
2360 '                                          [117]
2370 '                                          [117]
2380 ' Daten abspeichern                        [1712]
2390 '                                          [117]
2400 '                                          [117]
2410 MODE 2: GOSUB 3450                         [1498]
2420 PRINT #2,CHR$(24)"  Daten abspeichern [2753]
  "CHR$(24)
2430 spt=(ppu(1)+mpu(1))/2                      [1253]
2440 a$=STR$(spt)+".Spt"                        [827]
2450 OPENOUT a$                                 [765]
2460 FOR a=1 TO 18                              [950]
2470 PRINT #9,Name$(a)                          [1400]
2480 NEXT                                       [350]
2490 PRINT #9,ez%                               [1127]
2500 FOR a=1 TO 18                              [950]
2510 PRINT #9,ppu(a),mpu(a),gto(a),bto(a), [6515]
bs(a),bu(a),bn(a),platz(a),tg(a),tb(a),kon
(a),epu(a)
2520 NEXT                                       [350]
2530 CLOSEOUT                                   [902]
2540 RETURN                                     [555]
2550 '                                          [117]
2560 '                                          [117]
2570 ' Daten  laden                             [1528]
2580 '                                          [117]
2590 '                                          [117]
2600 MODE 2:GOSUB 3450                          [1498]
2610 PRINT #2,CHR$(24)"  Daten laden  "CHR [1802]
$(24)
2620 INPUT "Der wievielte Spieltag";spt         [2804]
2630 spt=INT(spt)                               [1361]
2640 a$=STR$(spt)+".spt"                        [828]
2650 PRINT "searching for ";a$                  [2233]
2660 OPENIN a$                                  [715]
2670 FOR a=1 TO 18                              [950]
2680 LINE INPUT #9,name$(a)                     [1480]
2690 NEXT                                       [350]
2700 INPUT #9,ez%                               [805]
2710 FOR a=1 TO 18                              [950]
2720 INPUT #9,ppu(a),mpu(a),gto(a),bto(a), [6988]
bs(a),bu(a),bn(a),platz(a),tg(a),tb(a),kon
(a),epu(a)
2730 NEXT                                       [350]
2740 CLOSEIN                                    [752]
2750 RETURN                                     [555]
2760 '                                          [117]
2770 '                                          [117]
2780 '  Unterprogramm : Ausgabe der Mannsc [4609]
haftsnamen auf den Bildschirm
2790 '                                          [117]
2800 '                                          [117]
2810 GOSUB 3450:' Windows                       [1457]
2820 CLS #1                                     [373]
2830 FOR a=1 TO 18 STEP 2                       [1336]
2840 x=x+1                                      [853]
2850 LOCATE #1,1,x                              [816]
2860 PRINT #1,a;"  ";name$(a)                   [1298]
2870 LOCATE #1,40,x                             [696]
2880 PRINT #1,a+1;"  ";name$(a+1)               [1312]
2890 NEXT                                       [350]
2900 x=0                                        [351]
2910 RETURN                                     [555]
2920 '                                          [117]
2930 ' Behandlung von Eingabefehlern  bzw. [3703]
 Einzeleingabe
2940 '                                          [117]
2950 CLS                                        [91]
2960 GOSUB 3300                                 [965]
2970 GOSUB 3330                                 [975]
2980 INPUT "Nummer der Begegnung deren Erg [8396]
ebnis ge|ndert werden soll [ ENTER ] wenn
fertig  ";n
2990 IF n<0 OR n>9 THEN PRINT "Eingabefehl [3366]
er":GOTO 2980
3000 IF n=0 THEN RETURN                         [681]
3010 a=kon(n*2-1):b=kon(n*2)                    [1099]
3020 PRINT name$(a);" : ";name$(b);"   ";t [1684]
b(b);" : ";tg(b)
3030 IF epu(a)=0 THEN epu(a)=1:epu(b)=1:GO [2043]
TO 3090
3040 IF tg(a)>tb(a) THEN bs(a)=bs(a)-1:bn( [3436]
b)=bn(b)-1:ppu(a)=ppu(a)-2:mpu(b)=mpu(b)-2
3050 IF tg(a)<tb(a) THEN bn(a)=bn(a)-1:bs( [6316]
b)=bs(b)-1:ppu(b)=ppu(b)-2:mpu(a)=mpu(a)-2
3060 IF tg(a)=tb(a) THEN bu(a)=bu(a)-1:bu( [5434]
b)=bu(b)-1:ppu(a)=ppu(a)-1:mpu(a)=mpu(a)-1
:ppu(b)=ppu(b)-1:mpu(b)=mpu(b)-1
3070 gto(a)=gto(a)-tg(a):bto(a)=bto(a)-tb( [5272]
a):gto(b)=gto(b)-tg(b):bto(b)=bto(b)-tb(b)

3080 PRINT "Zum L|schen des alten Ergebnis [4268]
ses nur  [ Enter ]   dr|cken !"
3090 PRINT "Richtiges Ergebnis (a:b)"           [2930]
3100 GOSUB 3550                                 [979]
3110 a%=e%:b%=z%                                [710]
3120 IF le=0 THEN epu(a)=0:epu(b)=0:GOTO 2 [1290]
920
3130 tg(a)=a%:tb(a)=b%:tg(b)=b%:tb(b)=a%        [1922]
3140 gto(a)=gto(a)+a%:bto(a)=bto(a)+b%:gto [4864]
(b)=gto(b)+b%:bto(b)=bto(b)+a%
3150 IF a%>b% THEN bs(a)=bs(a)+1:bn(b)=bn( [3648]
b)+1:ppu(a)=ppu(a)+2:mpu(b)=mpu(b)+2
3160 IF a%<b% THEN bn(a)=bn(a)+1:bs(b)=bs( [3952]
b)+1:mpu(a)=mpu(a)+2:ppu(b)=ppu(b)+2
3170 IF a%=b% THEN bu(a)=bu(a)+1:bu(b)=bu( [5913]
b)+1:ppu(a)=ppu(a)+1:mpu(a)=mpu(a)+1:ppu(b
)=ppu(b)+1:mpu(b)=mpu(b)+1
3180 CLS                                        [91]
3190 GOTO 2920                                  [339]
3200 '                                          [117]
3210 '                                          [117]
3220 'ja-nein Abfrage                           [1368]
3230 '                                          [117]
3240 '                                          [117]
3250 ' jn=0  f|r "Ja"   ;   jn=1 f|r "Nein [2306]
"
3260 '                                          [117]
3270 IF INKEY(45)=0 THEN jn=10:GOTO 3300        [1552]
3280 IF INKEY(46)=0 THEN jn=1:GOTO 3300         [1252]
3290 GOTO 3200                                  [441]
3300 a$=INKEY$                                  [278]
3310 a$=""                                      [388]
3320 RETURN                                     [555]
3330 '                                          [117]
3340 '                                          [117]
3350 '  Ausdrucken der Ergebnisse               [798]
3360 '                                          [117]
3370 '                                          [117]
3380 CLS #1                                     [373]
3390 FOR a=1 TO 18 STEP 2                       [1336]
3400 IF epu(kon(a))=0 THEN a$=" --":b$=" - [1958]
-"
3410 IF epu(kon(a))=1 THEN a$=STR$(tg(kon( [3336]
a))):b$=STR$(tb(kon(a)))
3420 PRINT #1,USING "#\\\                       [6089]
        \\ \\
   \\\\ \\\\ \";(a+1)/2,".)",name$(kon(a))
," :",name$(kon(a+1)),"",a$," : ",b$
3430 NEXT                                       [350]
3440 RETURN                                     [555]
```

```
3450 '                                          [117]
3460 '                                          [117]
3470 '  Windows definieren                      [933]
3480 '                                          [117]
3490 '                                          [117]
3500 MODE 2                                     [513]
3510 WINDOW #0,1,80,12,25                       [1162]
3520 WINDOW #1,1,80,2,11                        [974]
3530 WINDOW #2,1,80,1,1                         [1086]
3540 RETURN                                     [555]
3550 '                                          [117]
3560 '                                          [117]
3570 ' string "zerpfl}cken"                     [1578]
3580 '                                          [117]
3590 ' e$ in der Form e%:z% wird in e% und      [2653]
 z% zerlegt
3600 '                                          [117]
3610 LINE INPUT e$                              [1250]
3620 d$="":b$="":a$=" "                         [1038]
3630 le=LEN(e$)                                 [631]
3640 FOR p=1 TO le                              [1015]
3650 a$=MID$(e$,p,1)                            [929]
3660 w%=ASC(a$)                                 [341]
3670 IF se=0 THEN IF w%<48 OR w%>59 THEN P      [4661]
RINT "Zahleneingabe nicht in der Form erla
ubt  !!":GOTO 3610
3680 IF a$=":" THEN b$=d$:d$="":GOTO 3710       [1278]
3690 IF a$="" THEN GOTO 3720                    [844]
3700 d$=d$+a$                                   [449]
3710 NEXT p                                     [364]
3720 e%=VAL(b$)                                 [257]
3730 z%=VAL(d$)                                 [350]
3740 se=0                                       [280]
3750 RETURN                                     [555]
3760 '                                          [117]
3770 '                                          [117]
3780 ' Tabelle korrigieren                      [1562]
3790 '                                          [117]
3800 '                                          [117]
3810 MODE 2                                     [513]
3820 GOSUB 3300                                 [965]
3830 GOSUB 3450                                 [1039]
3840 GOSUB 2760                                 [899]
3850 PRINT #2,"Korrektur der Tabelle            [1698]
3860 PRINT "Geben sie die Nummer der Manns      [12970]
chaft ein deren Tabellenwerte ge{ndert wer
den     sollen (0 bzw. ENTER f}r R}ckkehr
ins Menue)":INPUT mnr%
3870 IF mnr%<0 OR mnr%>18 THEN PRINT "Numm      [4731]
er unzul{ssig . Bitte Eingabe wiederholen
.":GOTO 3860
3880 IF mnr%=0 THEN RETURN                      [728]
3890 CLS                                        [91]
3900 CLS #1                                     [373]
3910 PRINT #2,"Korrektur der Werte von ";n      [2307]
ame$(mnr%)
3920 PRINT:PRINT "Tore {ndern  ----------       [3267]
- T"
3930 PRINT:PRINT "Punkte {ndern ----------      [2288]
- P"
3940 PRINT:PRINT "Bilanz {ndern ----------      [2592]
- B"
3950 PRINT:PRINT "Ende der Korrektur -----      [2905]
- [Enter]"
3960 IF INKEY(51)=0 THEN GOTO 4020              [927]
3970 IF INKEY(27)=0 THEN GOTO 4120              [1280]
3980 IF INKEY(54)=0 THEN GOTO 4230              [1628]
3990 IF INKEY(18)=0 THEN GOTO 3760              [653]
4000 GOSUB 3300                                 [965]
4010 GOTO 3960                                  [481]
4020 '                                          [117]
4030 ' Tore {ndern                              [919]
4040 '                                          [117]
```

```
4050 CLS                                        [91]
4060 PRINT "Geben Sie die Tore in der Form      [5770]
    geschossene Tore  :   bekommene Tore
  ein !!"
4070 PRINT "Tore momentan :      ";gto(mnr      [3127]
%);" : ";bto(mnr%)
4080 GOSUB 3550                                 [979]
4090 gto(mnr%)=e%                               [571]
4100 bto(mnr%)=z%                               [1043]
4110 GOTO 3890                                  [471]
4120 '                                          [117]
4130 ' punkte {ndern                            [395]
4140 '                                          [117]
4150 CLS                                        [91]
4160 PRINT "Geben sie die Punkte in der Fo      [4858]
rm   Pluspunkte  :  Minuspunkte     ein  !
!"
4170 PRINT "Momentaner Punktestand :    ";      [5305]
ppu(mnr%);" : ";mpu(mnr%)
4180 GOSUB 3550                                 [979]
4190 ppu(mnr%)=e%                               [677]
4200 mpu(mnr%)=z%                               [1054]
4210 GOSUB 4320                                 [1065]
4220 CALL &BB18:GOTO 3890                       [1137]
4230 '                                          [117]
4240 ' Bilanz {ndern                            [1281]
4250 '                                          [117]
4260 CLS                                        [91]
4270 PRINT "Geben Sie die Bilanz in der Fo      [7705]
rm Siege,Unentschieden,Niederlagen ein."
4280 INPUT bs(mnr%),bu(mnr%),bn(mnr%)           [1804]
4290 GOSUB 4320                                 [1065]
4300 CALL &BB18                                 [389]
4310 GOTO 3890                                  [471]
4320 '                                          [117]
4330 ' Testen ob Punktekonto mit Bilanz }b      [3589]
ereinstimmt oder nicht
4340 '                                          [117]
4350 psum=bs(mnr%)*2+bu(mnr%)                   [1695]
4360 nsum=bn(mnr%)*2+bu(mnr%)                   [934]
4370 IF ppu(mnr%)=psum AND mpu(mnr%)=nsum       [4992]
THEN PRINT "Punkte und Bilanz stimmen }ber
ein "
4380 IF ppu(mnr%)<>psum THEN PRINT "WARNUN      [6112]
G : Pluspunkte und Bilanz stimmen nicht }b
erein"
4390 IF mpu(mnr%)<>nsum THEN PRINT "WARNUN      [6579]
G : Minuspunkte und Bilanz stimmen nicht }
berein"
4400 RETURN                                     [555]
```




HEINRICH STILLER 85



# SPIEL O DROME

**Es gibt Spiele und Spiele und auch solche Spiele, die sich standhaft weigern den Spieler weiter zu lassen als in das zweite Bild oder Level.**

**Allen denjenigen die eben mit dem einen oder anderen Spiel Ihre Probleme haben, ist dieser Teil unseres Sonderheftes gewidmet.**

Auf den folgenden Seiten werden Sie Tips und Tricks zu allerlei Spielen finden.

Ob Sie nun in einem Textadventure, einem Action Spiel oder was auch immer, festsitzen, blättern Sie diese Seiten durch, vielleicht finden Sie hier eben jenen Tip der Ihnen weiterhilft.

Ohne viel Federlesen wollen wir dann auch gleich anfangen und zwar mit..

## Die Erbschaft.

**Der erste Teil**

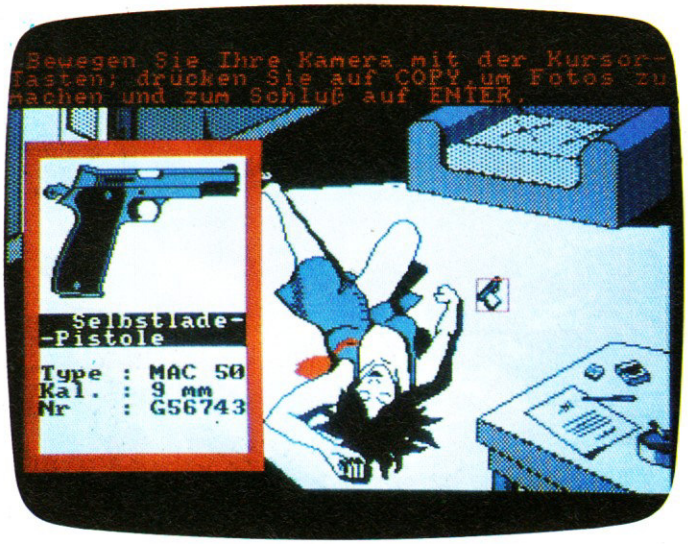
Zuallererst sollten Sie sich Ihr Apartment einmal genauer ansehen. Dort finden Sie eine ganze Menge Gegenstände, die Sie sich irgendwo ausgeliehen haben.
Auf dem Weg aus dem Haus heraus werden Sie von eben jenen Leuten angehalten, die nun von Ihnen den entliehenen Gegenstand wiederhaben wollen.
In der folgenden Tabelle sehen Sie genau welches Ding zu welcher Person gehört.

| | |
|---|---|
| Füller | großer Mann |
| Juwel | schüchterner Mann |
| Pistole | Mann im Trenchcoat |
| Trompete | Neger |
| Geld | Kaufmann |
| Kaktus | Mann mit Schweinskopf |

Das Geld für den Kaufmann finden Sie übrigens in einem Versteck im Schrank. Haben Sie diese Dinge eingesammelt, gehen Sie in das Treppenhaus, dort werden Sie einen nach dem anderen treffen, und können ihnen die geliehenen Dinge wiedergeben.

Haben Sie alle zufriedengestellt, gehen Sie zurück in Ihr Apartment. Dort angekommen rüsten Sie sich mit Ihrem Pass, dem Kamm und der Banane aus und sehen zu, daß Sie so schnell wie möglich wieder auf die Straße kommen. Am besten benutzen Sie die Treppe.

Im Erdgeschoß angekommen, benutzen Sie die Tür und notieren sich außerdem die Telefonnummer des Taxiunternehmens. Ab zum Telefon, dort wählen Sie die Nummer fürs Taxi. Jetzt nur noch




*Als Kommisar hat man's nicht leicht. Noch dazu, wenn es gilt, einen mysteriösen Mordfall zu klären. Die Dame Vera Cruz wurde das Opfer einer abscheulichen Bluttat. Sie müssen nun in Korrespondenz mit den Polizeicomputern im ganzen Land den Kreis der Verdächtigen immer weiter eingrenzen. Sollte Ihnen das Probleme bereiten, versuchen Sie es doch mal mit unseren Tips.*

warten bis die Taxe kommt, einsteigen und zum Flughafen fahren.

**Der zweite Teil.**
Wenn Sie den Flughafen betreten, stiehlt ein Penner ihren Pass. Den müssen Sie natürlich zurückbekommen bevor Sie ins Flugzeug können.
Der Penner hält sich zumeist in der Nähe der Flughafentoiletten auf. Wenn Sie ihn gefunden haben, geben Sie ihm Geld und er wird Ihnen Ihren Pass zurückgeben. Geben Sie ihm nochmals Geld, wird er Ihnen eine nette Geschichte über seinen Bruder in Las Vegas erzählen. Jetzt sollten Sie noch ein Sandwich essen und die Zeitschrift Mon Tricot kaufen.
Nun begeben Sie sich zu den Schaltern, zeigen dort Ihren Pass vor und warten bis 11 Uhr 19. Dann gehen Sie durch die fünfte Tür. Während des Fluges werden Sie die Bekanntschaft eines Luftpiraten machen. Um ihn zu besänftigen, geben Sie ihm die Zeitschrift. In Las Vegas angekommen, werden Sie einem Rocker begegnen. Geben Sie ihm die Banane.
Als nächstes werden Sie einem Skinhead über den Weg laufen. Dieser läßt sich mit dem Kamm zufriedenstellen. Sie sollten diesen Beiden keinesfalls zweimal über den Weg laufen, dann nämlich haben Sie verloren.
Um dahin zu kommen wo Sie hin wollen, müssen Sie den Bus Nr.6 nehmen. Doch Vorsicht die 6 am Bus ist umgekippt und sieht fast wie eine 9 aus, aber eben nur fast.

**Der dritte Teil**
Im Büro des Notars können Sie eine ganze Menge Zeug stehlen und es an den Gangster, der Russisches Roulette spielt, verkaufen. Sie werden Zeuge eines Striptease, die Dame die sich da entblättert ist die Stewardeß aus dem Flugzeug. Sie wird Ihnen Geld dafür geben, daß Sie niemandem erzählen was sie für einen Nebenjob hat. Nun müssen Sie nur noch so lange spielen, bis Sie eine Million Dollar gewonnen haben. Voila.

## Wer ermordete Vera Cruz

**Ermittlungen in der Wohnung von Vera Cruz:**

Außer den Gegenständen, die gut sichtbar im Raum herumliegen, befindet sich noch ein schwarzer Knopf im Zimmer. Er liegt im Schatten, links oberhalb der Handtasche. Anhand der nachstehenden Liste können Sie entnehmen ob Sie auch alle Indizien gefunden haben.

1. Pistole
2. Patronenhülse
3. Brief
4. Rothmans (Zigaretten)
5. Knopf
6. Faden (unter einem Fingernagel der linken Hand)
7. Handtasche
8. Notizbuch (dessen Inhalt erfahren Sie durch nochmaliges Drücken von Copy)
9. Aschenbecher
10. Linker Arm (Spuren)

**Ermittlungen im Präsidium:**
Die folgende Liste zeigt an, bei welchem Amt oder Polizeirevier Sie sich nach welchen Dingen zu erkundigen haben bzw. welche Fremduntersuchungen Sie in die Wege leiten müssen.

1. Autopsie Vera Cruz
2. Aussage Nadine Lafeuille Balaystr.
3. Anfrage CRRJ Lyon »Der Gelockte«
4. Aussage Abdulah Hocine Gasthof Pappel
5. Anfrage CRRJ Lyon »Der Zigeuner«
6. Anfrage GIE Galmier Philibert Ziegler
7. Anfrage PRIS St.-Paul Philibert Ziegler
8. CRRJ Lyon Eva Delarue
9. Aussage Eva Delarue Gasthof Fernfahrer
10. CRRJ Lyon Gilles Blanc
11. Anfrage CIAT Lyon Gilles Blanc
12. Aussage Hubert Delroche Etiene
13. Anfrage CIAT Etienne Hubert Delroche
14. Aussage Philibert Ziegler Carnot Platz
15. Vergleiche Philibert Ziegler
16. Aussage Hausmeisterin Forest
17. Aussage Gilles Blanc
18. Vergleiche Gilles Blanc
19. Untersuchung Graphologie Brief Gilles Blanc
20. Anfrage CRRJ Lyon Stanislas Kowalski
21. Anfrage GIE Mont Prison Stanislas Kowalski
22. Aussage Stanislas Kowalski E. Zolastr.
23. Anfrage CRRJ Lyon Pistole MAC 50 Kal. 9mm, Nr. g56743
24. Anfrage GIE Clermont Pistole
25. Anfrage PREF. Lyon X 9111 Cd 69 (Ausprobieren)
26. Anfrage CRRJ Lyon Philippe Blanc
27. Anfrage BDRJ Etienne Philippe Blanc
28. Anfrage CIAT Paris Pilippe Blanc
29. Aussage Philippe Blanc Bellecour
30. Vergleiche Philippe Blanc
31. Aussage Philibert Ziegler Carnotplatz
32. Aussage Stanislas Kowalski, E. Zolastr
33. Aussage Gilles Blanc Bahnhofstr.
34. Aussage Philippe Blanc Bellecour
35. Verhaftung Philippe Blanc

Nach dieser Prozedur ist Ihnen der Mörder ins Netz gegangen.

## Der kleine Kartograph

Sicherlich ist Ihnen beim ersten Durchblättern des Heftes schon aufgefallen, daß wir auch diesmal eine ganze *Menge Karten* und Lagepläne zu Spielen haben. Dazu möchte ich, in eigener Sache sozusagen, folgende Anmerkungen machen.

*Ein Teil der Karten bezieht sich auf Texte die Sie in diesem Sonderheft finden.*

Die anderen Karten stehen für sich selbst.

Leider war es aus layouttechnischen Gründen nicht immer möglich, Text und Karte zusammenzubringen. Um allerdings Verwirrungen vorzubeugen, ist jede Karte beschriftet.

## ACADEMY

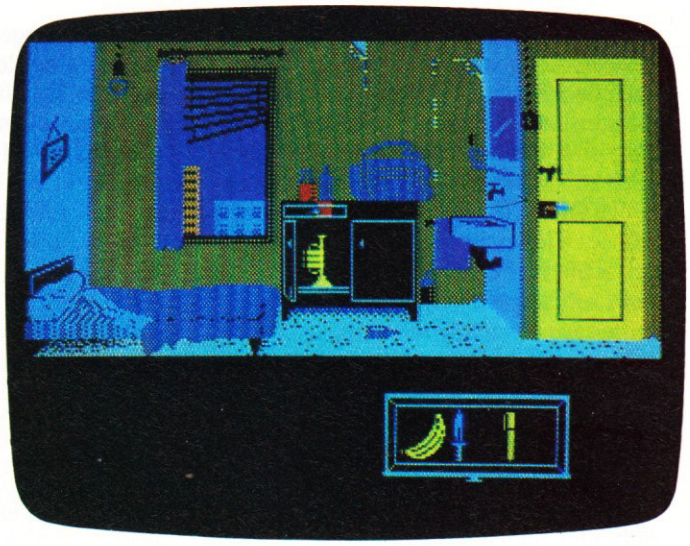
Für alle tapferen GAL.Corp. Skimmer Piloten hier einige Hilfen für das erste Missionspaket von Academy. Zuerst einmal etwas Grundlegendes: Die hier beschriebenen Skimmer waren mir treue Kampfgefährten, aber es gibt bestimmt Leute, die andere Typen bevorzugen.
Die Bauanleitungen sind also nichts Zwingendes. Außerdem sollte man jeden Skimmer erst bauen, wenn man die vorherige Mission geschafft hat, da man vier Skimmer braucht, jedoch nur drei selbstgebaute gleichzeitig im Speicher halten kann. Doch nun frisch ans Werk. Zuersteinmal den Kadettennamen eingeben und bei Bedarf die Tasten neu definieren.
Nun den ersten Skimmer zusammenbauen, der für die Mission »Wenn es sich bewegt...« gebraucht wird.

**Teile:**
Scanner
Kompaß-ADF
IR-Einheit
Vier Raketen
Laserintensität MED
Hauptmotoren MED
Schilde HIGH
Steuereinheit MED

**a) Name: Apokalypse**
Ist dies geschehen, stürzen wir uns ins Kampfgeschehen. Die Schwierigkeit bei dieser Mission liegt eigentlich darin, daß man sich oft gegen zwölf oder mehr Hunters und Konsorten zur Wehr setzen muß. Es kommt also hauptsächlich auf die Leistung des Skimmers und auf das Geschick des Piloten an.
Nach ein paar Versuchen wird Ihnen diese Mission gelungen sein. Für Mission Nummer zwei brauchen wir wieder einen anderen Skimmer.

**Teile:**
Scanner
Kompaß-ADF
Sprung/Tür Einheit
Vier AMM's
Vier Zeitbomben
Laserintensität MED
Hauptmotoren MED
Schilde HIGH
Steuereinheit MED

**b) Name: Tango**
Diese Mission ist schon etwas schwieriger, denn auch im Kampf mit einer Armee von Huntern darf man nur einen Schuß auf eine Sprungfläche abgeben, da man sonst ein ganzes Stückchen fliegen darf und einem dabei meistens der Sprit ausgeht. Noch ein paar Tips: Bei einem Angriff auf die Roboterfabriken muß man ständig bereit sein, eine AMM oder eine Zeitbombe abzufeuern.



*Daß die Schnorrer- und Pumperei ein Geschäft ist, das gelernt sein will, muß man als Spieler bei »Die Erbschaft« recht bald feststellen. Eine ganze Schar von Gläubigern versucht Sie daran zu hindern, in Las Vegas eine Millionenerbschaft anzutreten.*

Man fliegt genau zwischen den beiden Fabriken hindurch und legt neben den beiden Gebäuden eine der Zeitbomben ab. Jetzt nur noch mit vollem Tempo in Sicherheit bringen. Weiterhin feuern die Fabriken oft und gerne Raketen auf den Skimmer ab.
Da er nicht mehr als fünf Stück aushält, feuert man sofort, nachdem die Meldung »Roberrakete« (Na, einen Schreibfehler konnten sich die CRL Jungs wohl nicht verkneifen) erschienen ist, eine AMM ab. Ist dies alles geschehen, kommen wir sogleich zur dritten Mission.
Für Mission Drei brauchen wir einen Spezialskimmer. Er besteht aus folgenden Einzelteilen.

**Teile:**
Scanner
Kompaß-ADF
IR-Einheit
Vier Zeitbomben
Laserintensität HIGH
Hauptmotoren HIGH
Schilde HIGH
Steuereinheit MED

**c) Name: Windwraith**
Bei dieser Mission ist es nicht empfehlenswert, die Leuchttürme abzuschießen, da sie einem den Weg zum Reaktor weisen. Ab und an treffen wir einige gemeine Zeitgenossen: Die Fortresses.
Diese schießen schon außerhalb der eigenen Laserreichweite. Ist man nahe genug am Reaktor, den Skimmer so ausrichten, daß das ADF genau in der Mitte steht, Zeitbombe zünden, und mit Höchstgeschwindigkeit weg. Kommt der GLV in Sicht, andocken. Und schon sind wir bei der letzten Mission des ersten Paketes angekommen. Der Skimmer für diese Mission muß folgendermaßen aussehen.

**Teile**
Scanner
Kompaß
Acht Raketen
Laserintensität HIGH
Hauptmotoren HIGH
Schilde HIGH
Steuereinheit HIGH

**d) Name: Foxtrott**
Hier muß man ein Minenfeld durchfliegen, um letztendlich am GLV anzudocken. Das Minenfeld ist ganz klar durch die regelmäßig stehenden Minen zu erkennen. Entdecken Sie einmal eine Unregelmäßigkeit: Warten bis sie nahe genug beim Skimmer ist und eine Rakete abfeuern. Bis jetzt ist mir kein Feind untergekommen, der eine AMM abgeschossen hat, also stehen die Erfolgschancen dementsprechend hoch.

Sind Ihnen einmal die AMM's ausgegangen, müssen Sie zum Laser greifen, was Ihnen bei seiner Intensität auch nicht schwer fallen wird. So, und mit ein wenig Glück haben Sie auch das hinter sich gebracht und Sie können Paket zwei laden. Hierzu will ich Ihnen aber keine Tips geben um Ihnen den Spaß nicht vollständig zu verderben.

Ich drücke Ihnen aber die Daumen, daß Sie es zu Elite-Skimmer-Piloten schaffen.

## MERCENARY

Nun folgen einige Tips für all diejenigen, die auf Targ notgelandet sind und dort in irgendeiner Weise Probleme haben. Ganz zu Anfang sollten Sie sich die Arbeit machen, die Instruktionen genau zu lesen. Dort sind einige wichtige Tips versteckt.

Im eigentlichen Spiel sollten Sie immer ein Augé auf die Meldungen Bensons haben. Wenn Sie dann den Dominion Dart bestiegen haben, sollten Sie beim Überfliegen von Targ eine Flughöhe von 100 bis 250 Metern einhalten. In dieser Höhe

sind Bodenstrukturen und Bauwerke am besten erkennbar. Lernen Sie Ihren Dart nach dem Kompaß zu manövrieren. Die meisten interessanten Plätze die es auf Targ gibt, befinden sich in der Stadt, jedoch außerhalb in den Ödländern gibt es einige Orte, die aufzusuchen sich lohnen würde. Vielleicht sollten Sie einmal in der Nähe von LOC 81–35 suchen.

Bei Luftkämpfen in oder über der Stadt sollten Sie vorsichtig mit dem Laser umgehen und darauf achten, keines der Gebäude zu treffen. Wenn die nämlich genug Schüsse abbekommen haben, stürzen sie ein. Dies ist zwar ein kapitaler Effekt der auch lustig anzusehen ist, jedoch werden Sie sogleich als Agressor deklariert und haben in Zukunft noch viel mehr Ärger mit den Patrouilliengleitern. Es ist recht hilfreich sich während des Spiels eigene Landmarken zu schaffen. Es erleichtert die Orientierung mitunter ungemein.

Wenn Sie bereits einige Gebäude untersucht haben, werden Sie sicherlich den einen oder anderen Gegenstand mit sich führen der auf den ersten Blick keine Aufgabe erfüllt. Landen Sie und legen einfach einen solchen entbehrlichen Gegenstand ab. Wenn Sie sich an die angegebene Flughöhe halten, werden Sie die Gegenstände auch aus der Luft erkennen und sich an ihnen orientieren können.

In der Nähe von LOC 09–06 befindet sich ein Aufzug mit dem Sie in die unterirdischen Zonen von Targ vordringen können. Landen Sie so zentral wie möglich im Aufzugskäfig. Durch Drücken von E gelangen Sie nun in den Hangar unter dem Aufzug. Verlassen Sie dort Ihr Schiff um nun die unterirdische Anlage zu erkunden. Beim Herumwandern werden Sie eine ganze Reihe von Gegenständen finden. Die meisten davon identifiziert Benson sobald sie aufgenommen wurden. Aber Sie sollten sich überlegen welche der Gegenstände Sie wirklich brauchen. Beim Untersuchen des Komplexes werden Sie auch einen Palyar Planungsraum finden, in dem Sie einige interessante Informationen erhalten. Mercenary hält, außer den beschriebenen, noch viele weitere Überraschungen für den Spieler bereit.

Mit den Tips auf diesen Seiten sollten Sie es eigentlich geschafft haben, mit den Steuerungen und dem Spielprinzip von Mercenary klarzukommen. Alles Weitere wollen wir aber Ihnen überlassen.

## SPELLBREAKER

### Teil 1 – Der Traum

Zu Beginn des Spieles steht man im Twistet Forest und wird sogleich von einem Höllenhund angegriffen. NO bietet einen Fluchtweg an der sofort eingeschlagen werden sollte. Flüchtet man nicht auf der Stelle, so ist das auch nicht weiter tragisch. Man wird zwar sogleich von dem Höllenhund zerfleischt, aber das bedeutet nur das Ende eines Alptraums, in dem man sich zu Beginn des Spiels befindet. Allerdings kann dieser Traum genutzt werden um die Gegend, in der sich der wesentliche Teil des Spieles abspielt, gefahrlos zu erkunden. Wenn Sie dazu keine Lust haben, dann können Sie sich an dieser Stelle getrost auffressen lassen.

### Teil 2
### Wie man den ersten Tag überlebt

Zuerst einmal ist es ratsam, aus dem Bett aufzustehen. Von dort aus gehen Sie nach Osten in den Hallway, nach Süden bis zur Lobby und nach Westen in den Storeroom. Hier nehmen Sie alles mit und gehen nach Osten.

Im folgenden Raum werfen Sie das Matchbook in den Behälter der sich da Receptacle zu nennen pflegt. Nach einigen Eingaben klingelt es an der Haustür. Wenn dies passiert, öffnen Sie den Behälter und entnehmen ihm das Orange Vial. Der Rest ist für den weiteren Spielverlauf nicht wichtig. In der Library und im Hellister Quarter liegen der MEEF beziehungsweise der GASPAR Spell herum. Diese werden ge-GNUSOt. Nun schaut man im Belboz Qarter hinter das »Beautifull woven wall hanging« und nimmt den Schlüssel der sich dort befindet.

Öffnen Sie die Schublade des Schreibtisches. Mit der »Tiny Box« verfahren sie gleichermaßen. Darin befindet sich ein Amulett, das Sie sich um den Hals hängen sollten. Die Intensität der Leuchtkraft des Amuletts verhält sich proportional zur Entfernung zu Belboz.

Dem gefundenen Journal können Sie mit dem Schlüssel zuleiberücken. Beim Lesen des Journals werden Sie einen Code entdecken. Dieses Codewort ist der Name eines jener Ungetüme, die Ihnen im GUE begegnen können. Man schaut nun im »Infotator« (dieser liegt der Packung bei) unter dem entsprechenden Ungeheuer nach und merkt sich die Farbkombination am Ende der Beschreibung. Nun begibt man sich in den Cellar und »presst« die Knöpfe am »trunk« in Reihenfolge der Farbkombination.

Gehen Sie gerade dabei sehr gewissenhaft vor, da bereits eine falsche Kombination den »trunk« für immer verschließt. Haben Sie die Kombination richtig eingegeben, springt der »trunk« auf und zum Vorschein kommt ein »Scroll«. Der Zauberspruch darauf ist lang und kompliziert.

Inzwischen müßte Ihre Spielfigur so hungrig sein, daß ein weiteres Vorgehen an Nahrungsmangel scheitern könnte. Abhilfe schafft die »Orange Vial«, öffnen, trinken und dann die leere Flasche ablegen. Diese Mahlzeit hält bis zur Befreiung von Belboz vor.

Bevor man sich nun zu einem verdienten Nickerchen hinlegt, nimmt man noch die mächtige Scroll und AIMFIZt sich zu Belboz. Man landet zwar nicht bei letztgenannten, aber man hat das Haus und damit den Teil zwei des Spiels verlassen.

### Teil 3
### Wie man Jeearr (JR???!!) austreibt.

Wir befinden uns nun im Twistet Forrest, der bereits durch den Alptraum bekannt sein sollte. Auch Kollege Höllenhund kann nun in Natura bewundert werden. Fliehen Sie zum Forrest Edge.

Es folgen drei Lokationen die mit Vorsicht zu genießen sind. Meadow, River Bank und Drawbridge. In der Ersten sollte man maximal eine Eingabe tätigen und mit der Zweiten das Weite suchen.

In den beiden anderen muß nicht unbedingt etwas passieren, allerdings sollten Sie mit dem Einstürzen der Brücke oder einem Alligatorenangriff rechnen. Sie können die Räume auch gefahrlos mit dem IZYUK Spell durchqueren.

Nun das weitere Vorgehen: Am Forrest Edge lernt man IZYUK und PULVER. Man begibt sich zu MEADOW, IZYUKT sich selbst und fliegt zur River Bank. Hier muß das Wasser schleunigst gePULVERt werden. Nach der Landung wird das Dark Cave aufgesucht. »Guano« und »Scoll« werden gebraucht, das »Amber Vial« ist reiner Luxus (wenn man sie tatsächlich mal gebrauchen kann, hat man meist bereits eine andere Vial zu sich genommen und zwei Vials vertragen sich nicht).

Nun GNUSOt man FWEEP, IZYUKt sich selbst und fliegt in die Tortutre Chamber. Die hier herumliegende Indigo Vial ist zwar nicht besonders wichtig, trinken sollte man sie aber trotzdem.

Der Weg führt jetzt zum Parade Ground. Hier läßt man die Flag herunter und untersucht sie. Aqua Vial mitnehmen. Nun weiter zum Gun Emplacement. Der Vogelmist wird in die Kanone gesteckt. Dadurch werden die lästigen Yipples verjagt und man kann in aller Ruhe die »Scroll« mit dem extrem langen unkomplizierten YONK Spell aus der Kanone herausnehmen.

Für das weitere Vorgehen werden zwei Zorkmids benötigt. Den Einen kann man von dem wunderschönen Baum im Tree Room pflücken.

Natürlich muß die Schlucht auf dem Weg dorthin überflogen werden. Mit dem so organisierten »Zorkmid« verschaffen Sie

sich nun Zugang zum Vergnügungspark. Achterbahn, Geisterhaus und Wildwasserbahn sind offensichtlich nicht weiter wichtig. Um in der Arcade den Hasen zu treffen, trinken Sie am besten vorher die »Fooble Potion« aus dem »Aqua Vial«. Der Gewinn, ein »Spell«, wird geGNUSTOt. Nun zum Zocken ins Casino. Wenn Sie im Casino angekommen sind, sollten Sie vor dem Weiterspielen unbedingt den Spielstand absaven. Nun kann das Spiel, das mitunter sehr lange dauert, beginnen. Haben Sie Pech, gewinnen Sie nichts, haben Sie Glück, bleiben Sie vom Hauptgewinn verschont (in diesem Falle wird man regelrecht vom Geld erdrückt) und wenn Sie besonders viel Glück haben, gewinnen Sie genau ein Zorkmid.

Mit diesem können Sie nun die Gebühr am Troll Gate bezahlen. Dadurch haben Sie unbegrenzt Zugang zum östlichen Teil des Geschehens.

Nun beginnt der Kampf mit dem Glaslabyrinth. Man begibt sich zum Outside Glass Arch und schläft sich aus. Nun ist es an der Zeit den FWEEP Spell zu lernen, am besten gleich viermal. Wichtig ist, an diesem Punkt alles abzulegen, empfehlenswert den Spielstand abzusaven. Nun betritt man das Labyrinth.

In dieser Situation sollten Sie sich mit den Vorzügen des Fledermaus Daseins auseinandersetzen. Das Labyrinth ist übrigens ein Würfel, bestehend aus siebenundzwanzig kleineren Würfeln. Durch das den Fledermäusen eigene Orientierungssystem, können die jeweiligen Ausgänge gut geortet werden.

Fliegt man »N E S S W D E E N N U U S E«, so gelangt man zum Hollow. Hier warten Sie so lange bis Sie ihre ursprüngliche Gestalt wiederhaben. Der dort liegende Scroll muß mitgenommen werden, allerdings hat das zur Folge, daß sich das gesamte Labyrinth verändert.

Wirft man nun den Scroll in das Loch (hole), hat man genau eine Eingabe Zeit sich wieder in eine Fledermaus zu verwandeln. Just zu diesem Zeitpunkt scheint ein, in dem Schatten des Cliffs schlummernder, »Dorn« seines knurrenden Magens gewahr zu werden. Da Sie das einzig Jagbare weit und breit sind, stellt der »Dorn« sogleich hinterher. Um dem Dorn zu entkommen, sollten Sie unverzüglich wieder nach Westen in das Labyrinth fliegen.

Solange es möglich ist immer weiter nach Westen und dann eine Richtung aus der nicht gerade der Dorn angestürmt kommt. So verfahren Sie solange bis Sie in einen Raum ohne Boden gelangen. Der Dorn wird blindlings hinter Ihnen herjagen und sich in diesem Raum zu Tode stürzen.

Nun sollten Sie sich wieder zum westlichen Ausgang begeben. Übrigens haben Sie nach einem angewandten FWEEP Zauber 13 Eingaben Zeit, bis Sie sich wieder zurückverwandeln. Dem Labyrinth entkommen, sollten Sie alle abgelegten Gegenstände wieder aufnehmen und sich auch wieder mit der, in das Loch geworfenen Scroll versehen.

Nächste Station: Hall of Carvings. Der MALYON Spell wird geYONKt und der Drache geMALOYNt. Hierdurch wird der Steindrache für wenige Momente zum Leben erweckt, lange genug um aus seiner Nische hervorzutreten und den Durchgang freizugeben.

Außer dem Calendar dem Spellbook und der Orange Vial können Sie an dieser Stelle alles ablegen. Öffnen Sie nun das Vial und treten Sie durch die Nische. – Sie befinden sich nun im Coal Bin Room. Hier ist vor lauter Gas und Gestank erst einmal kein Durchkommen.

Trinken Sie das Orange Vial. Nach Genuß dieser Portion scheint für einen Moment die Welt zu wackeln und aus einem Schacht an der Decke rutscht Ihr Zwillingsbruder. Untersuchen Sie ihn, werden Sie feststellen, daß er einen Scroll bei sich hat, aber diesen nicht an Sie weitergeben will. Nachdem er Ihnen eine Zahlenkombination genannt hat, verlangt er Ihr Spellbook, welches Sie ihm auch bereitwillig aushändigen sollten.

Ihr Zwilling verschwindet daraufhin und auch Sie sollten keine Zeit mehr verlieren. Begeben Sie sich in den Dial Room. Das Dial wird auf die Kombination, die Sie von Ihrem Zwilling erhielten, eingestellt. Nun weiter zum Shaft Bottom. Legen Sie den Calendar weg und nehmen Sie das Seil. Dieses wird geFROTZT.

Nun durch die Coal Mines zum Top of Chute. Das Timber wird über das Loch gelegt und das Seil an das Timber geknotet und das andere Ende in das Loch geworfen. Nun ist es wichtig, daß Sie nichts mehr bei sich tragen.

Klettern Sie an dem Seil nach unten. In dem dortigen Raum finden Sie eine weitere Scroll. Golmac Spell-Travel temporary. Dieser Zauber bewirkt eine Zeitreise und muß sofort benutzt werden. Sogleich finden Sie sich in der Rolle des Zwillings wieder.

In der kleinen Schachtel findet sich plötzlich ein Scroll, der natürlich sogleich mitgenommen wird. Wenn Sie nun den Raum verlassen, rutschen Sie in den Coal Bin Room. Hier treffen Sie nun sich selbst zu dem Zeitpunkt als Sie den Raum das erste Mal betraten.

Nennen Sie Ihrem Doppelgänger die Kombination, daraufhin wird er Ihnen das Spell Book geben. Begeben Sie sich nun zum Lagoon Shore. Mit dem gefundenen Scroll können Sie sich hier gegen unliebsame Geister schützen.

Für das weitere Spiel ist nun ein Blick in die Zukunft nötig (VEZZA). Dabei werden Sie die verschiedensten Visionen haben, von denen allerdings nur die mit der weißen Tür und Belboz wichtig ist. Erlernen Sie nun den MEEF-Spell, legen alles ab und begeben Sie sich zur Lagune. Dort sollten Sie tauchen und die dort blühenden Blumen mit dem Meef Spell behandeln.

Daraufhin erscheint eine Schachtel die Sie mitnehmen. Jetzt begeben Sie sich wieder zu der Stelle, an der Sie das Zauberbuch abgelegt haben. In der Schachtel werden Sie eine Lampe, das berüchtigte Magic Frobozz Grue Repellent und einen Schutzanzug finden. Mit Letzterem halten Sie sich die lästigen Grues vom Leibe. Wenn Sie sich nun wieder komplett ausgerüstet haben, begeben Sie sich zu der Höhle deren Eingang von gefährlichen Schlingpflanzen bewacht wird. Diese können Sie mit dem MEEF Spell unschädlich machen.

Haben Sie den Eingang freigelegt, zünden Sie die Lampe an und begeben Sie sich in die Höhle. Die Grues werden ignoriert. Wenn Sie sich dann, wie in Ihrer Vision, in dem Raum mit den drei Türen befinden, wählen Sie die Weiße.

Im Raum dahinter finden wir endlich den armen Belboz, der ja vom bösen Jeearr besessen ist. Lernen Sie nun den geisteraustreibenden Spruch und vertreiben damit Jeearr aus Belboz Körper. Jeearr muß nun weichen und in einen anderen Körper fahren. Da der Ihrige allerdings der Einzige ist, aber gegen Geister geschützt, bleiben Jeearr nur noch die ewigen Jagdgründe. Somit ist Spellbreaker gelöst.

## They Stole a Million

Willkommen zu unserem kleinen Lehrgang: Wie begehe ich den Raub des Jahrhunderts? Dazu muß man sich natürlich erst einmal ein geeignetes Objekt aussuchen. Für Diebe, deren Barschaft noch nicht allzuviel beträgt, nehmen wir einmal Herbert Humbug's Münzladen. Zuerst kaufen wir die Blaupausen des Gebäudes und zusätzlich Informationen über Münzen. Man erfährt nun, daß im Haus die wertvolle Halb-Zloty-Münze liegt, die 15.000 $ wert ist. Jetzt besorgen wir uns einmal einen Hehler.

Hierzu eignet sich Kleingeld-Sue hervorragend, da sie am besten Münzen und Silber verschachern kann. Ein gut eingespieltes Team darf natürlich auch nicht fehlen. Aber da bei Herbert nicht allzu große Barschaft liegt, verzichten wir auf mehrere Leute und mieten uns Babyface Clive, da er Fahrer und Schlosser zugleich ist.

Wer will, kann sich noch den äußerst preiswerten Big Dave Cannon kaufen.

Nun gehen wir in die Planungsphase über. Dort macht man am besten einen lauten, aber kurzen Bruch, daß heißt, man darf ruhig Alarm auslösen. Da bei dieser Art Einbruch die Polizei allerdings 25 – 30 Minuten früher kommt, sollten sich die Teammitglieder um 1:40 im Auto einfinden. Der Bruch geht nun folgendermaßen vor sich: Clive öffnet die Haustür, geht durch die Ausstellungsräume in den hinteren Teil des Gebäudes und knackt dort die Tür. Dann betritt er den Raum dahinter und bricht die nördliche Tür auf. Nachdem er die Küche betreten hat, öffnet er das Schloß der Ablage und kehrt zum Auto zurück. In der Zeit sollte der Chef die beiden 7000$-Kästen aufgebrochen und entleert haben und findet sich ein paar Minuten später im Auto ein.

Falls Sie Cannon noch im Team haben, sollte er sich den 6.500$ und den 5.000$ Schrank schnappen. Jetzt muß er noch zum Auto laufen und Clive muß losfahren. Nun kann man sehen, wieviel man erbeutet hat. Wenn Sie etwa 65 – 70.000 $ zusammenhaben, können Sie sich an die Planung des zweiten Einbruchs machen. Das beste Team kann man folgendermaßen zusammenstellen: Man muß herausfinden, wieviel jedes einzelne Teil wert ist. Ist das geschehen, muß man folgende Stellen besetzen: Fahrer, Schlosser und bei Bedarf Tresor- oder Sprengstoffexperte.

Auch ein preiswerter Schläger darf ruhig mit von der Partie sein. Die besten Gauner sucht man sich aus dem folgenden Haufen heraus.

Zusätzlich muß man noch die Kosten für Blaupausen, Informationen zu Alarmanlagen, Panzerschränken und Beute bedenken.

**Die Creme de la Creme der Gangster**

| Name | 1. / 2. Talent | Preis | Bet. |
|---|---|---|---|
| Flanigan | Panzer/Spreng | $ 21 | 15% |
| Fredriks | Panzer/Schlosser | $ 20 | 15% |
| Roll | Spreng/Aufpasser | $ 11 | 10% |
| D'Arcy | Spreng/Fahrer | $ 17 | 10% |
| Smith | Alarm/Fahrer | $ 25 | 10% |
| Volts | Alarm/Schläger | $ 19 | 10% |
| Davies | Alarm/Panzer | $ 26 | 10% |
| Chimes | Alarm/Schlosser | $ 22 | 15% |
| Williams | Fahrer/Schlosser | $16 | 16% |
| Prost | Fahrer/Aufpasser | $ 15 | 15% |
| Clint | Aufpasser/Alarm | $ 11 | 5% |
| Cannon | Aufpasser/Schläger | $ 4 | 5 % |
| Jones | Schläger/Aufpasser | $ 5 | 5% |
| Martin | Schläger/Spreng | $ 9 | 5% |
| Carter | Schlosser/Panzer | $ 16 | 5% |
| Joe | Schlosser/Aufpasser | $ 9 | 9% |
| Clive | Fahrer/Schlosser | $ 18 | 10% |
| Branigan | Panzer/Schläger | $ 18 | 10% |

Noch etwas zur Erläuterung: Panzer bedeutet Panzerknacker, Spreng ist Sprengstoffspezialist & Alarm ist Alarmexperte. Man kann, wenn man Fahrer und Schlosser sucht, diese auch durch eine Person ersetzen. Man muß die Kosten der beiden Personen zusammenrechnen, und wenn das Ergebnis mehr beträgt als der Schlosser und Fahrer in einer Person, sollte man ihn wählen. Es sollten allerdings genügsame Mitarbeiter sein, 15% der Beute abzugeben sind sehr schwer zu verschmerzen. Hier noch ein paar interne Favoriten:

*Fahrer:* Detonator D'Arcy, Adam Prost, Babyface Clive. *Schlosser:* Skelett Joe, Babyface Clive. *Alarm:* Charlie Volts, Smith. *Panzer:* Boxer Branigan, Schlüssel Carter, Detonator D'Arcy, Sülze Roll. *Schläger:* Mad Man Martin, Big Dave Cannon, Jones.

In späteren Gebäuden kommen immer raffiniertere Alarmsysteme zur Geltung: Abstandsalarm, der durch Lichtschranken ausgelöst wird, Druckalarm, der sobald man einen so gesicherten Raum betritt ausgelöst wird, Kameras, die sobald etwas Fremdes den Raum betritt, Alarm schlagen & Wachmänner. Diese zu umgehen, gestaltet sich recht schwierig, da Sie mindestens zu zweit auftreten. Falls ein lauter Bruch nicht mehr funktioniert, sollten Sie unter »Sicherheit« die Information über die Anlage abrufen, die Ihnen am meisten Ärger macht. So, mit diesen Hinweisen sollte es Ihnen möglich sein, sich bis zum Narburak Museum vorzuarbeiten. Ich wünsche Ihnen einen guten Bruch!

## Adventure-Potpurri

An dieser Stelle wollen wir einige Tips zu den unterschiedlichsten Adventures an den Mann bringen. Doch fangen wir am besten gleich an, mit:

**Suspect**

Zuerst nimmt man das zerbrochene Glas. Kurz nachdem Monica den Ballraum verlassen hat, folgt man ihr in das Büro. Dort findet man sie bereits erwürgt auf. Zuerst nimmt man das Seil an sich, schaut unter den Körper, nimmt die Patrone, untersucht den Papierkorb und nimmt die Maske & die Karte. Außerdem sollte man den Umschlag an sich nehmen. Jetzt geht man wieder in den Ballraum und wartet vor dem Kamin, bis Colonel Marston erscheint, und sich die Hände am Feuer wärmt. Jetzt die Feuerstelle untersuchen und das Papier nehmen. Anschließend den Detektiv suchen und ihm das Glas, den Umschlag, die Karte und das Haar geben. Nun werden Sie nicht um 12 Uhr verhaftet, sondern können ihre Ermittlungen weiterführen.

**Infidel**

Erst einmal aus der Hängematte steigen und das Zelt verlassen. Jetzt die Schaufel und die Spitzhacke suchen. Den Rucksack sollte man auch mitnehmen. Wieder das Zelt betreten, das Schnappschloß mit der Spitzhacke öffnen. Das Schloß entfernen und die Kiste öffnen. Die Karte auseinanderfalten und den Würfel an sich nehmen. Alles bis auf die Feldflasche steckt man in den Rucksack und füllt die Feldflasche am Nil. Nun kann man sich in die Wüste aufmachen und dort die Pyramide suchen.

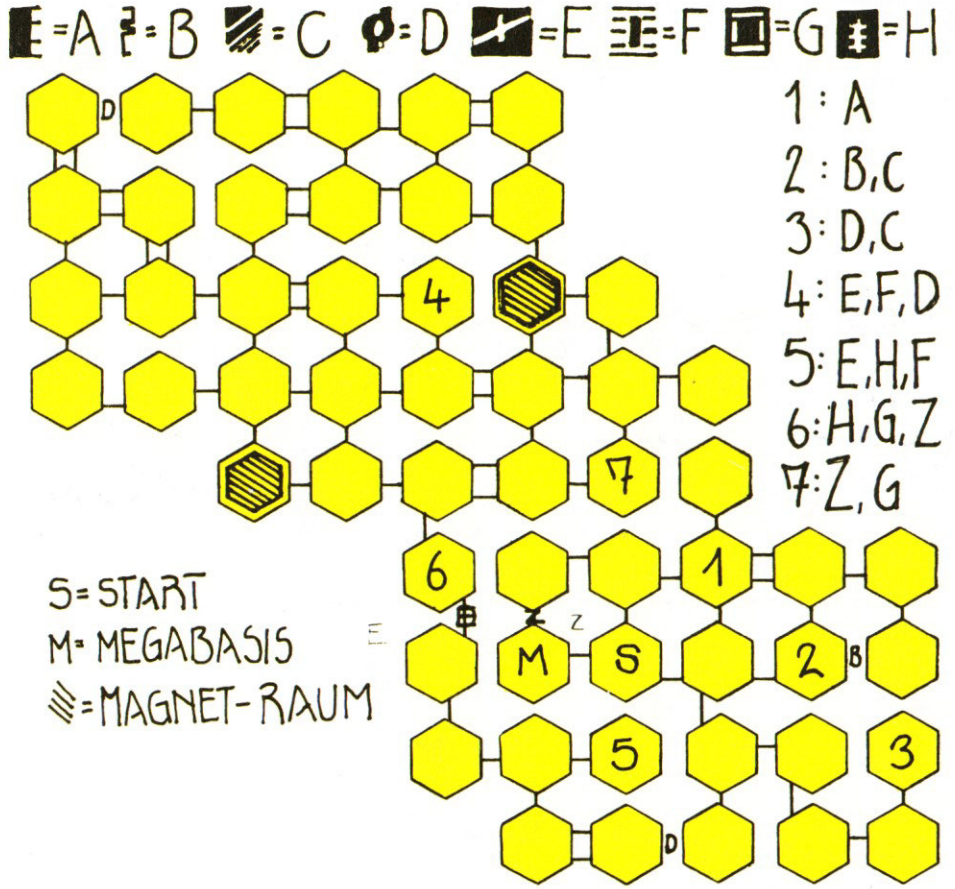
**Red Moon**

Zuerst zum Vulkan gehen und die Handschuhe nehmen. In dem alten Haus alles an sich nehmen, die Lampe anstellen und das Schloß auf dem unterirdischen Weg betreten. Sie werden unterwegs eine kleine Tür und einen Pilz finden. Ein Tip:Nach Genuß des Pilzes werden Sie ein Stück kleiner... Man kann auch zum See gehen, den Busch untersuchen. Nun den Hebel in das Loch stecken und denselben drehen. Der See wird auslaufen und der Weg ins Schloß ist geebnet. Im Schloss findet man eine Schmiede. Das dort befindliche Schwert läßt sich nur unter Zuhilfenahme der Handschuhe nehmen. So, ich hoffe, allen Abenteurern sind mit diesen Tips geholfen. Und denken Sie daran: Sicherheitsabstand von einem Ork: 190 Km.

**M. G. T.**

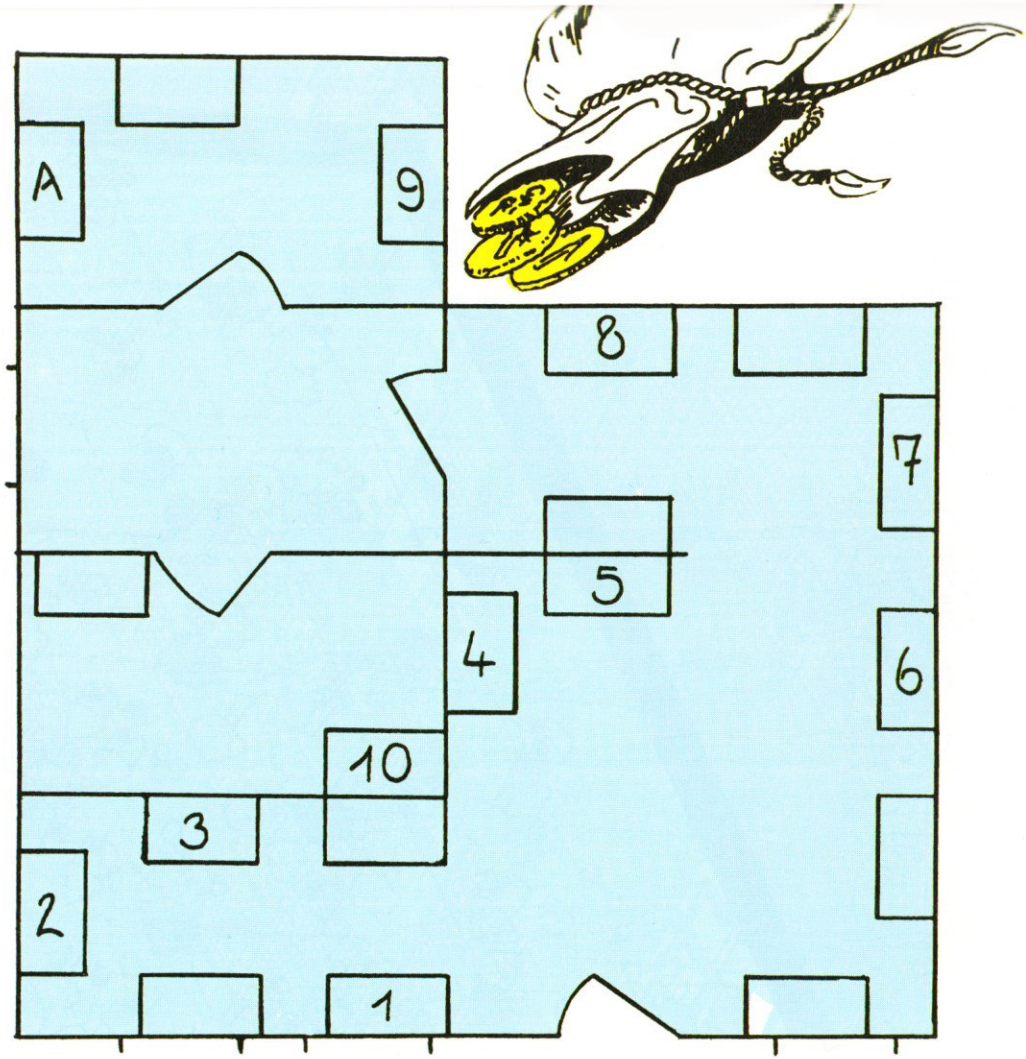
Nachricht an alle professionellen MGT-Piloten: Eine Karte des Einsatzgebietes und Hinweise zur Zerstörung der Megabasis. Zuerst einmal etwas Grundsätzliches:Die Schalter sind nicht immer zum Öffnen von Türen, einige haben auch weniger gute Auswirkungen auf das MGT. Der Schalter C verdreht die Steuerung für eine Weile, während F die gesamte Munition zerstört. G schließlich verringert die Energie des MGTs auf die Hälfte. Von diesen Schaltern sollte man lieber die Finger – pardon: die Laser lassen. Es gibt noch Räume, die einiger Erklärung notwendig sind. Der Raum, der leer aussieht, ist unsichtbares Labyrinth. Man muß einige verschlungene Wege fahren, bis man zum anderen Ausgang kommt. Außerdem gibt es einen Raum in dem drei Lasersperren wie Billiardkugeln ziellos im Raum hin- und herrasen. Hier muß man den richtigen Augenblick abwarten und sich an der linken Wand entlangtasten. In den Magnet-Räumen sollte man keineswegs den Boden berühren, da sonst das einzige MGT hinüber ist. In Räumen, in denen Eisblöcke den einzigen Weg versperren, sollte man einmal alle Eisblöcke anschießen. Oft hebt sich einer und man kann mit dem MGT im richtigen Moment drunterdurchfahren. Die Blöcke können das MGT auch zu höher gelegenen Türen transportieren. Jetzt schwing' Dich aber in dein MGT, starte durch und beschleunige auf Höchstgeschwindigkeit in Richtung Megabasis.

MESSAGE END..... (HS/BB/JH)



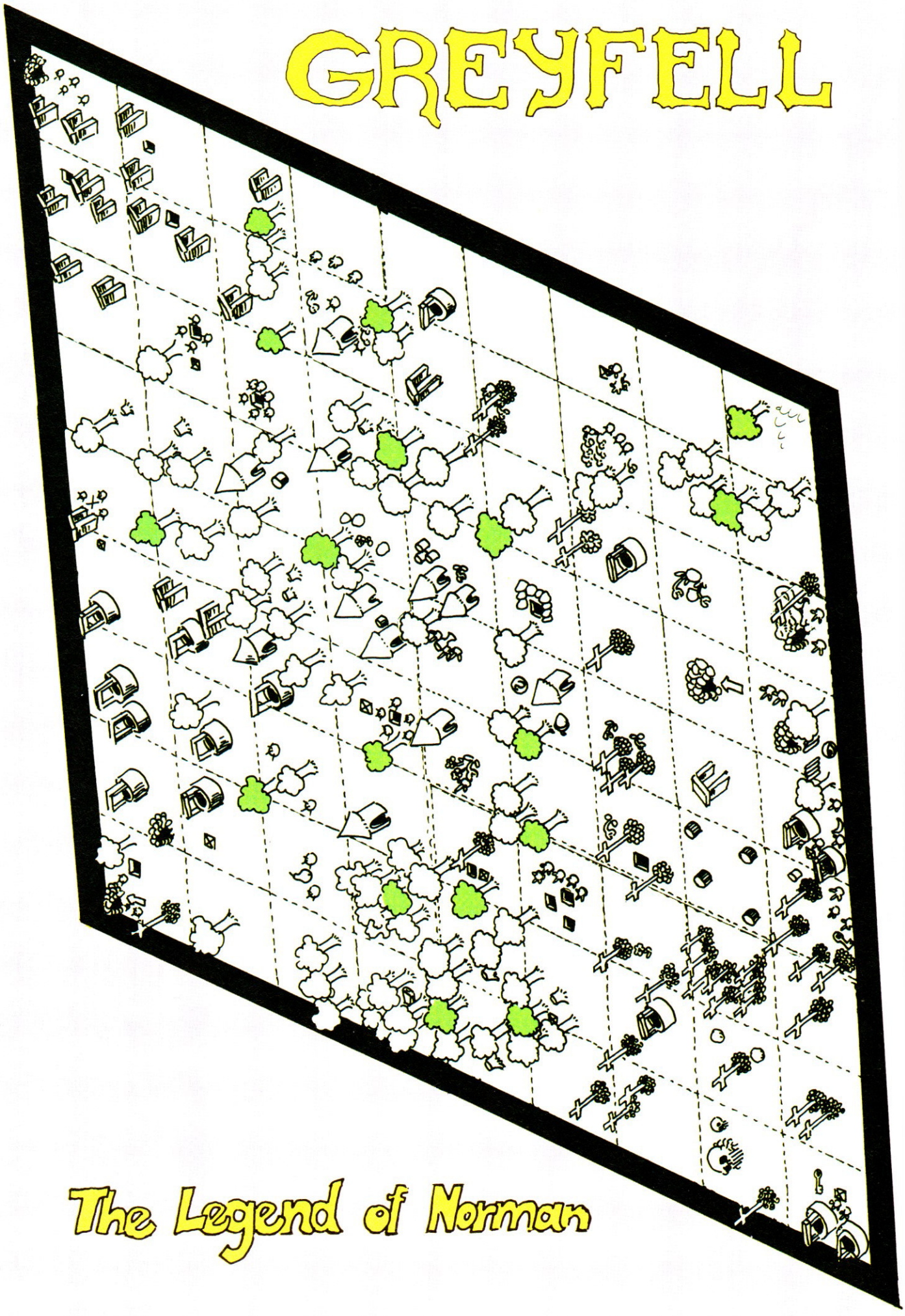




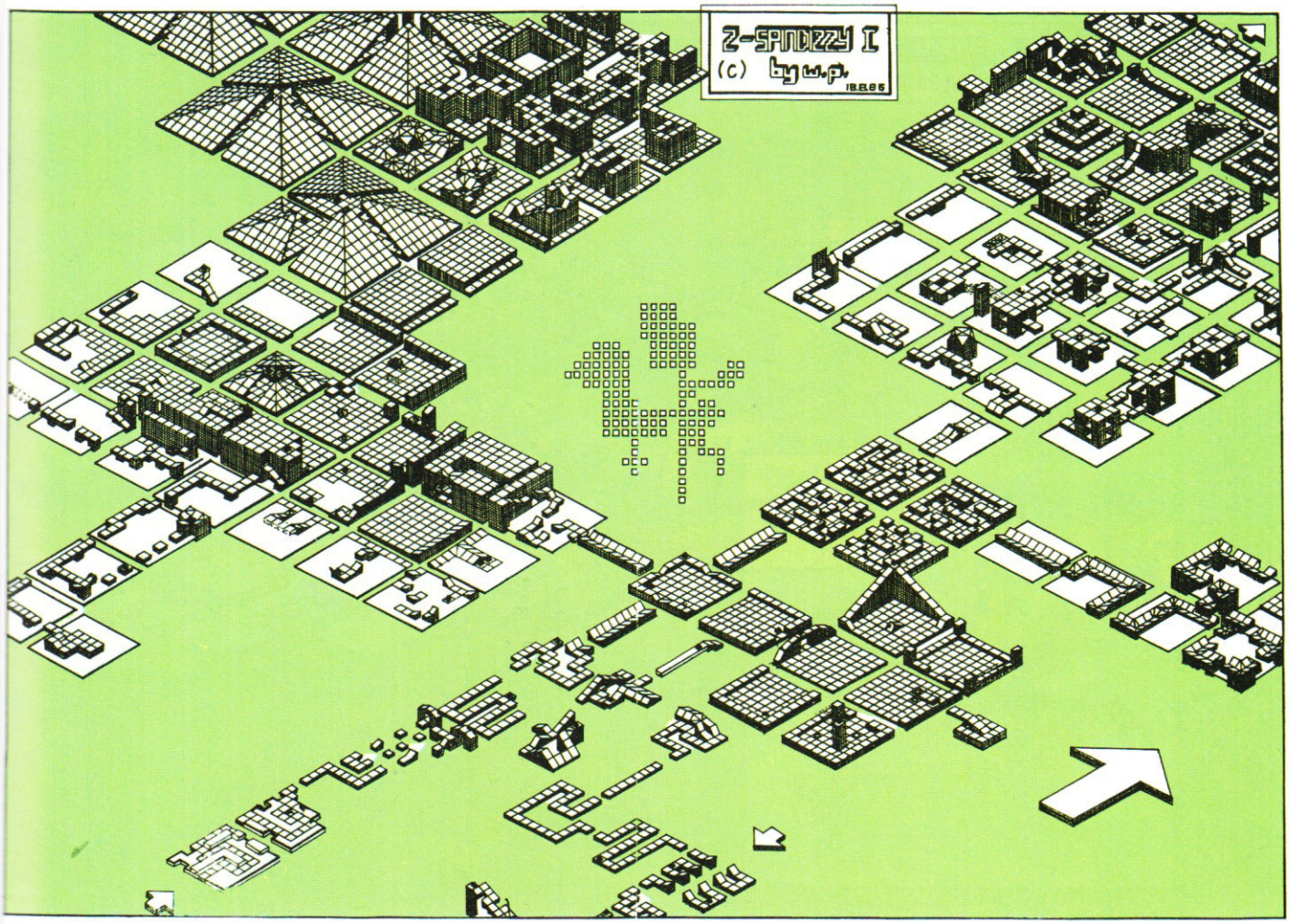


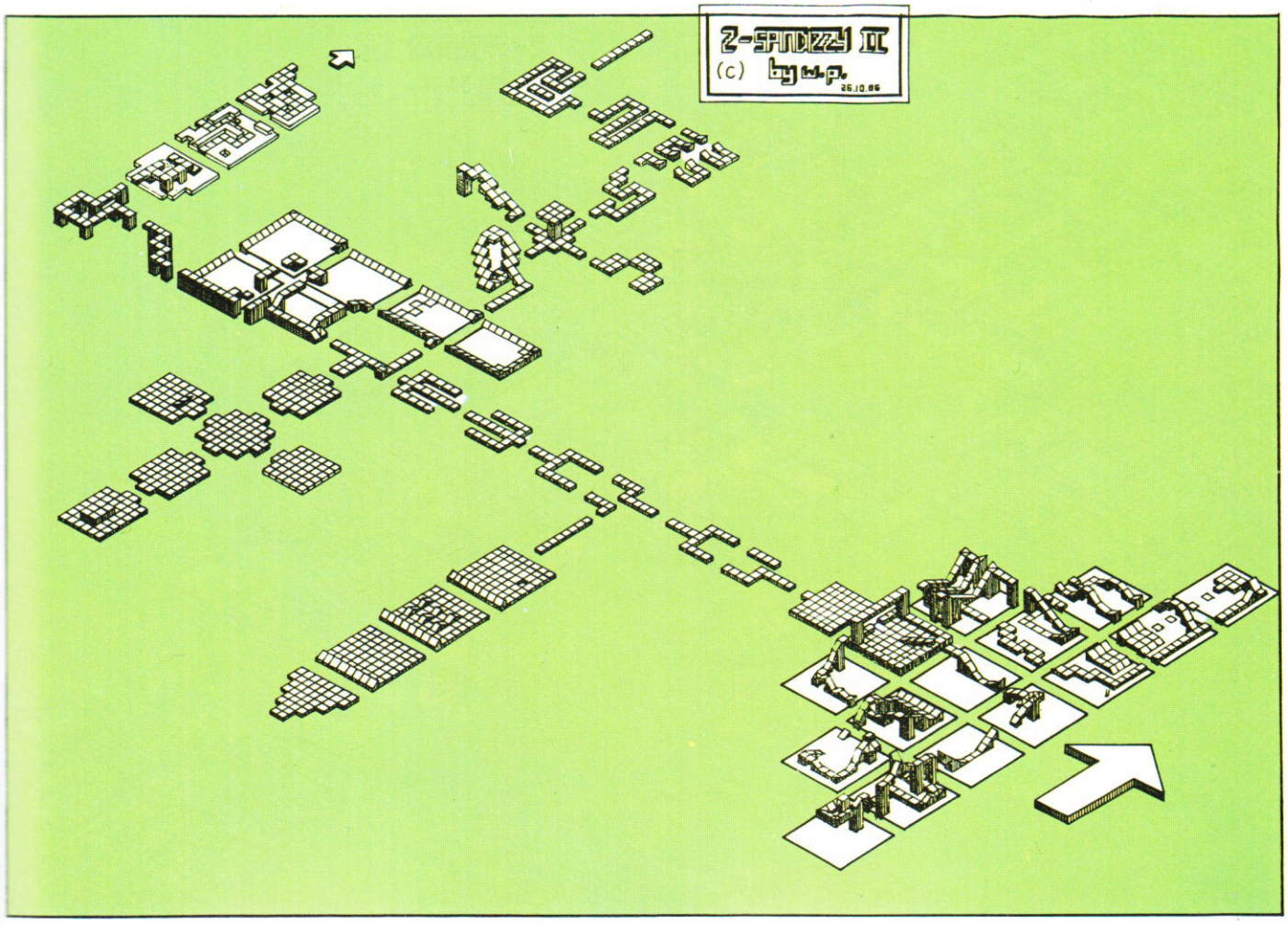

# THEY $TOLE A MILLION

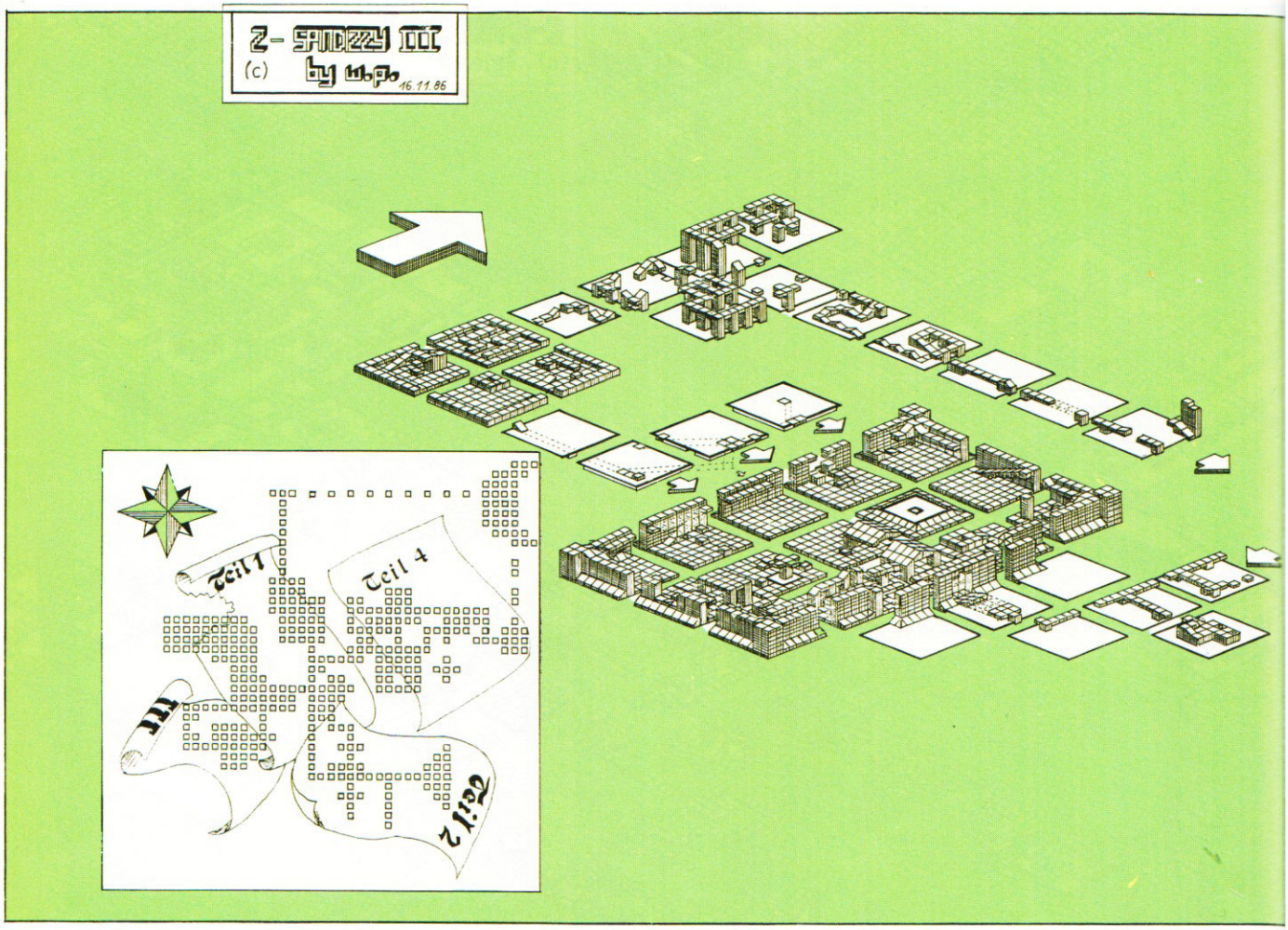
# GREYFELL

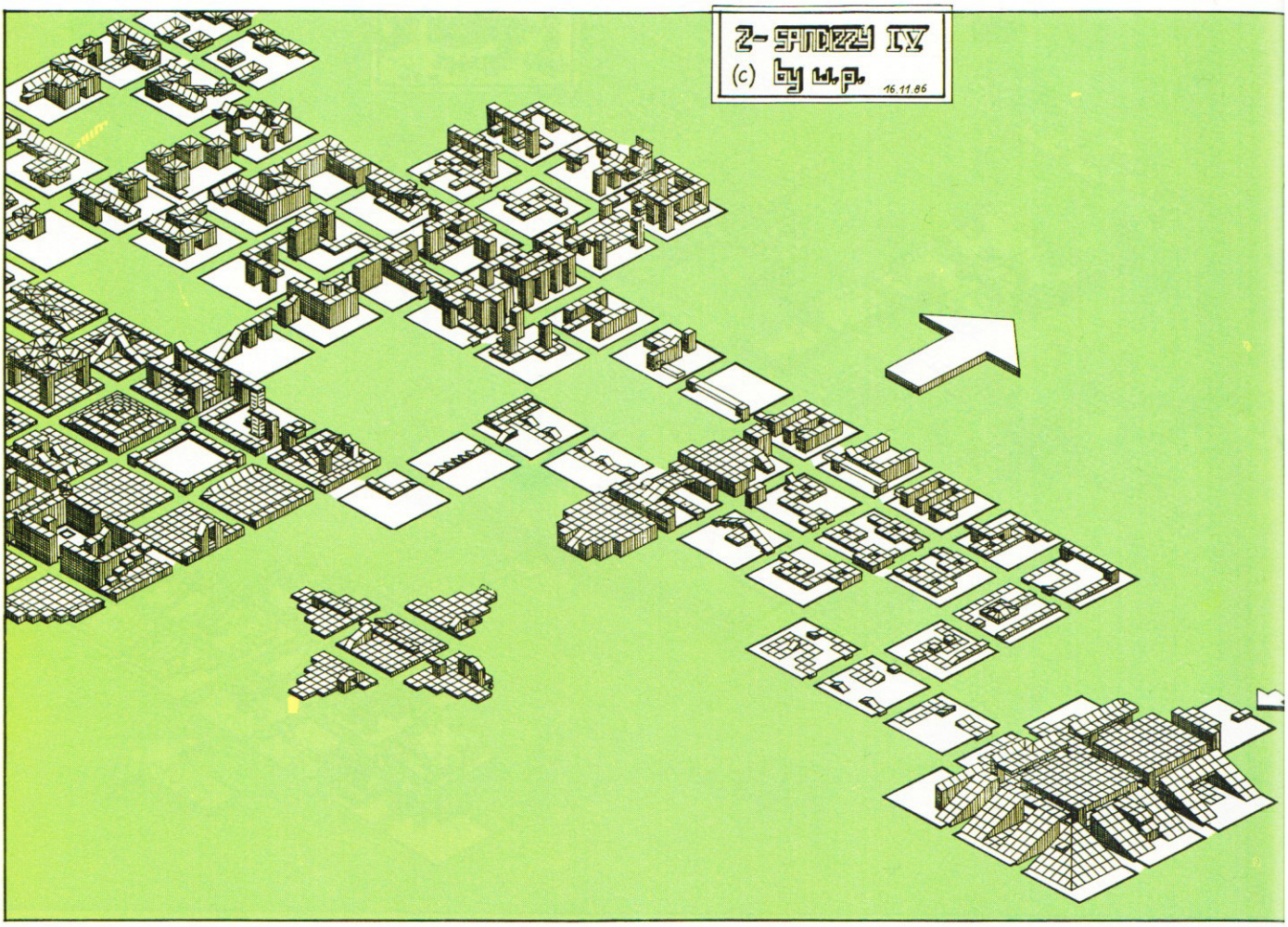


## The Legend of Norman


Z-SPINDIZZY I
(C) by W.P.

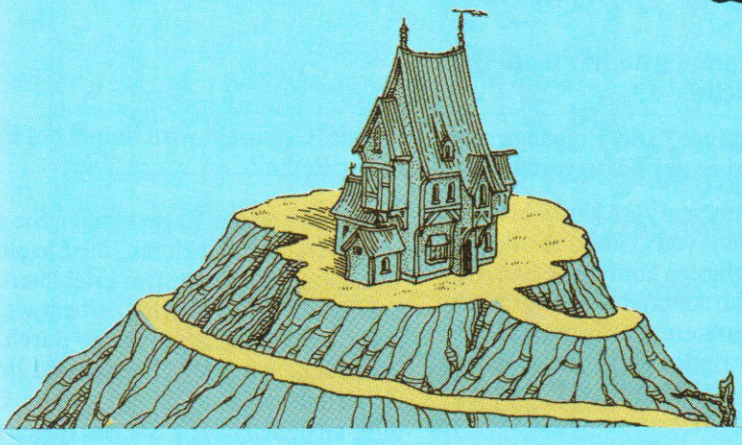

Z-SPINDIZZY II
(C) by W.P.


Z-SPIZZY III
(c) by W.P. 16.11.86
Teil 1
Teil 4
Teil 3
Teil 2


Z-SPIZZY IV
(c) by W.P. 16.11.86

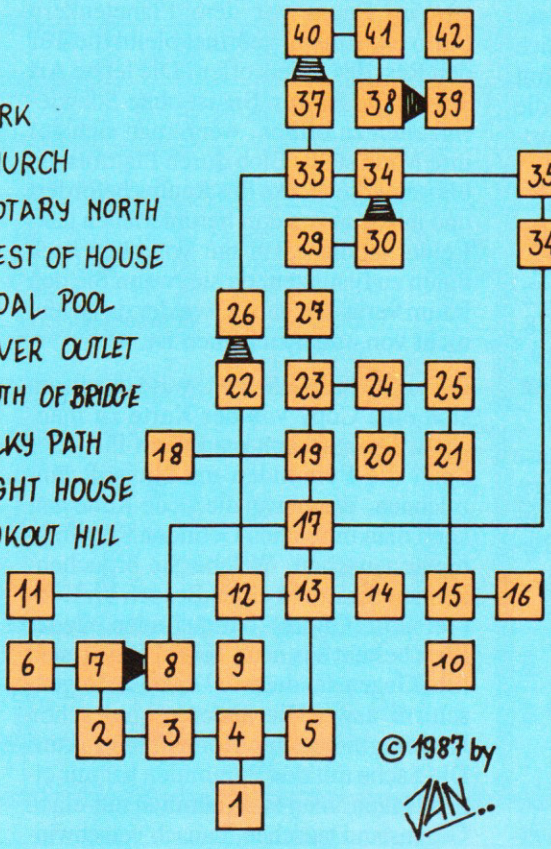
# Wishbringer



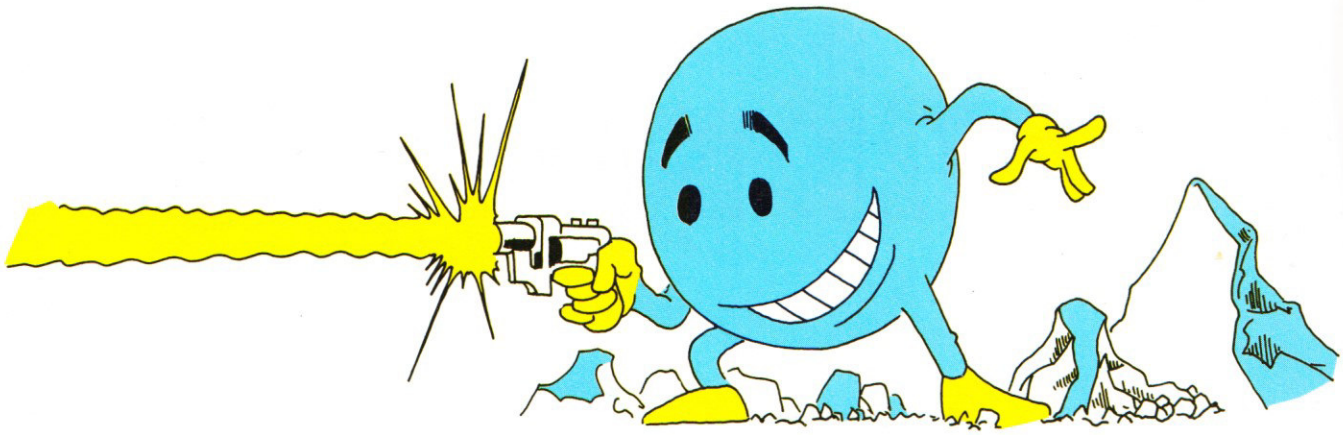
1 POST OFFICE
2 CREEPY CORNER
3 OUTSIDE CEMENTERY
4 HILLTOP
5 OUTSIDE COTTAGE
6 TWILIGHT GLEN
7 SPOOKY COPSE
8 OPEN GRAVE
9 POLICE
10 VIDEO ARCADE
11 LAKE END
12 ROTARY WEST
13 ROTARY SOUTH
14 ROTARY EAST
15 PLEASURE HARF
16 HARF END

17 PARK
18 CHURCH
19 ROTARY NORTH
20 WEST OF HOUSE
21 TIDAL POOL
22 RIVER OUTLET
23 SOUTH OF BRIDGE
24 ROCKY PATH
25 LIGHT HOUSE
26 LOOKOUT HILL




© 1987 by JAN

27 BRIDGE
28 CELL
29 NORTH OF BRIDGE
30 CLIFF BOTTOM
31/32 UNDERGROUND HOLE
35/36 UNDERGROUND
37/38/40/41 CLIFF
42 INSIDE MAGICK SHOPPE
39 OUTSIDE MAGICK SHOPPE
43 GRUE'S NEST



# Starquake

**Es gibt Spiele, die fängt man an zu spielen und auf einmal wundert man sich, warum es schon so spät ist. Starquake ist ein solches.**

Um es gleich vorweg zu nehmen, ich habe Tage an diesem Programm verbracht und das nicht nur mit Anfertigen der Karte, doch bis jetzt ist es mir noch nicht gelungen, den Kern vollständig zu erneuern. Das liegt ganz einfach daran, daß alle Gegenstände, sowohl jene zum Aufstocken von Energie, Plattformen und Schuß, als auch die Kernfragmente, bei jedem Spiel neu verteilt werden. Rein nach Murphy schaffe ich es nachdem ich diesen Artikel abgeliefert habe beim ersten Versuch.

Der endgültige Erfolg setzt sich aus drei Faktoren zusammen: Know-how, einer Karte und einer gehörigen Portion Glück. Bis auf letzteres kann ich Ihnen behilflich sein. In der Karte sind alle Teleports mit einer Zahl gekennzeichnet. Hier nun die dazugehörigen Namen mit den Standortkoordinaten:

| | | | |
|---|---|---|---|
| 1) RALIQ | P2 | 2) VOREX | I3 |
| 3) TALIS | C5 | 4) ASCIO | G10 |
| 5) DULON | C11 | 6) QUORE | F14 |
| 7) ANGLE | B19 | 8) ELIXA | H22 |
| 9) KRYZL | M24 | 10) UPAZZ | B28 |
| 11) OPTIK | N29 | 12) INDOL | J29 |
| 13) SNODY | G30 | 14) AMBOR | K32 |
| 15) ZODIA | D32 | | |

Ich glaube, eine Legende ist für diese Karte überflüssig, da die Formen weitgehend aus dem Spiel übernommen wurden. Die Buchstaben haben folgende Bedeutung:

R – Raumschiff
F – Falltür
O – Olly (Krake)
S – Skelettonaut
L – Anti-Gravitätslifte
SD – Security Door
D – Einfache Tür

In dem Labyrinth des Planeten gibt es einige Geheimgänge. Die Eingänge und Ausgänge sind in der Karte verbunden.

Soweit zur Karte, nun zum Know-how. Es gibt drei Arten von Türen in diesem Spiel. Am häufigsten kommt die Security Door vor. Man kann sie nur mit einer Code-Karte passieren, auf der ein Dreieck ist. Man findet diese meistens im Startraum (I1) oder aber bei der Teleportstation Vorex.

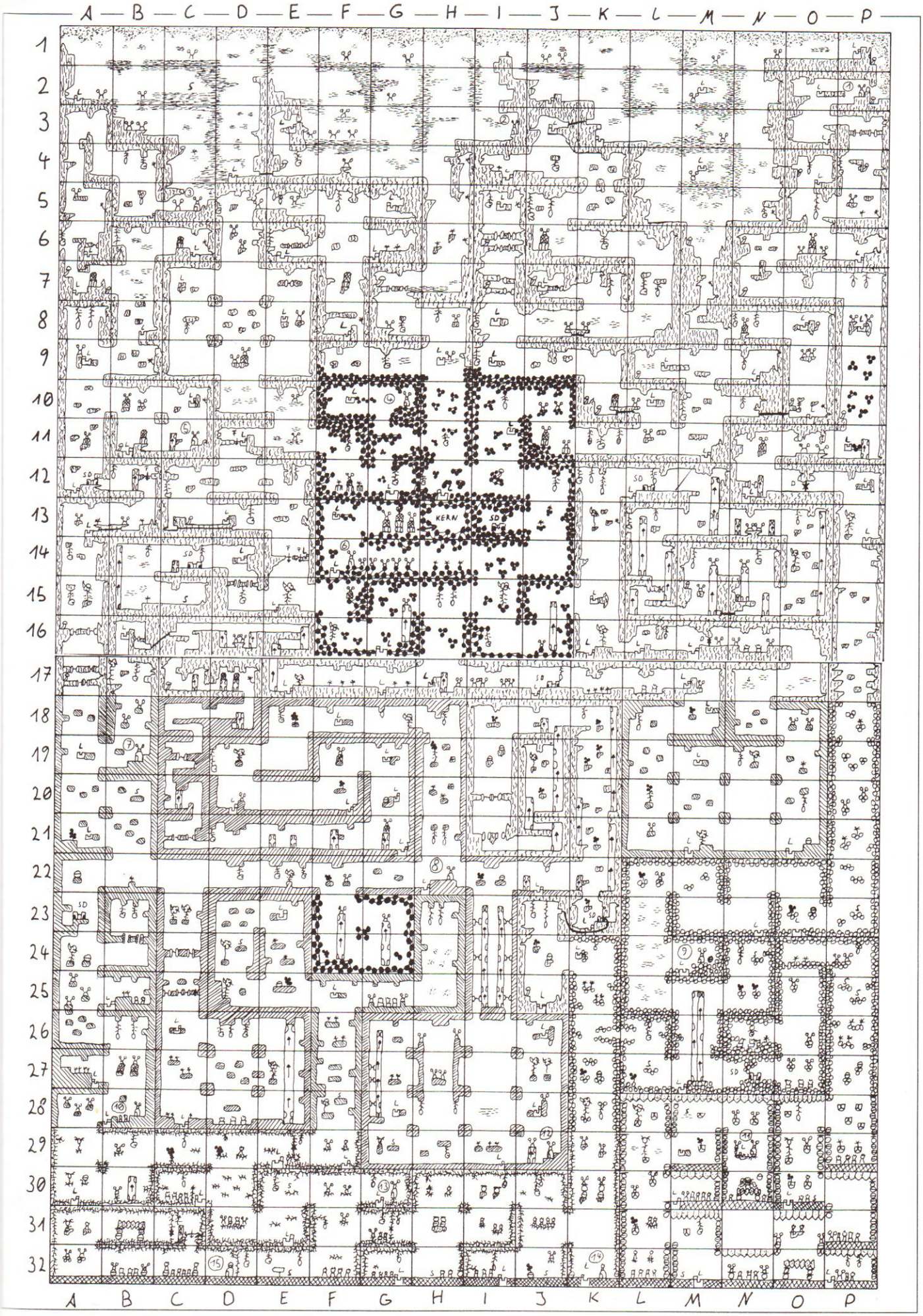
Am einfachsten ist es, solange neu anzufangen, bis man sie in einem der beiden Räume gefunden hat. Für die einfache Tür brauchen Sie einen Schlüssel, der öfters im Raum vor dem Planetenkern (G13) ist. Einmal geöffnet bleibt die Tür den Rest des Spiels offen. Die letzte Art Tür ist die Falltür. Sie ist ohne Schwierigkeiten zu öffnen, wenn man sich auf ihre Mitte stellt, Blob durch Plattformen bis unter die Decke des Raums befördert und ihn dann von dort herunterfallen läßt. Leider ist diese Tür nur von oben nach unten zu benutzen, da sie, wenn Sie den Raum verlassen haben, wieder da ist und nicht von unten zu öffnen ist.

Bevor Sie nun anfangen, wäre es ratsam sich eine Copy von der Karte zu machen. Darauf kann man dann für jedes Spiel die Pyramiden mit Bleistift einzeichnen. Wenn man die Code-Karte hat, kann man in solchen Gebilden Kernfragmente tauschen. Welche Sie brauchen, können Sie durch einen Besuch im Kern (Teleport Quore) herausfinden. Jeder Kern besteht aus neun Teilen. Davon sind fünf Gegenstände, wie Diskette, Regenschirm usw.. Die anderen sind schon Kernfragmente in Monekülarstruktur. Die Sache mit den Pyramiden hat nur einen Haken, man kann nämlich nur einen Gegenstand tauschen, danach verschwinden sie. Getauscht wird immer der Erste in der Reihe.

Auf der Copy der Karte können Sie sich auch Bereiche markieren, die Sie einem Teleporter zuordnen. Das erleichtert die Suche erheblich. Mit der Zeit entwickelt jeder seine festgelegte Route durch die verschiedenen Teile. Dabei wird Ihnen sicherlich auch auffallen, daß es Orte gibt, an denen häufig Sachen liegen und andere, die man von der Suche von vorne herein ausklammern kann.

So, nun zu den »Resourcen-Auffrischern«. Die Wirkung der einzelnen Gegenstände ist so offensichtlich, daß ich sie wohl nicht erklären brauche. Man sollte allerdings sparsam mit ihnen umgehen und sie z.B. nicht gleich aufnehmen, wenn man weiß, daß man dies noch auf dem Rückweg kann. Achten Sie immer ein bißchen auf die Anzeigen der Resourcen. Falls eine auf Null sinken sollte, lassen Sie sich am besten in einen noch nicht besuchten Teil des Planeten teleportieren. Die meisten Anti-Gravitätslifte haben an den Rändern Einsteigehilfen. Ohne diese ist es nicht immer einfach sich den Lift zu holen. Wenn Sie allerdings mit dem Joystick in Schrägposition über ihn hinweg laufen, gibt es keine Probleme.

In manchen Vertiefungen wandern Geräte hin und her, die wie die Stromabnehmer einer Straßenbahn aussehen. Man kann sie abschießen, sollte allerdings aufpassen, denn wenn man in der Vertiefung den Löffel abgibt, kann es vorkommen, daß gleich der ganze Besteckkasten verloren ist, da die Viecher wieder auftauchen. So nun aber ran an die Praxis. Lassen Sie sich bloß nicht frustrieren, ich bin überzeugt, es ist zu schaffen.


A B C D E F G H I J K L M N O P
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
KERN
SD

# The Sentinel

**Seit ich die Regeln kapiert habe, gehört The Sentinel zu meinen Lieblingsspielen. Vielleicht erleichtert Ihnen dieser Artikel den Einstieg in die Welt der 10000 Ebenen.**

Auch nach zweimaligem Durchlesen der Anleitung hatte ich bei diesem Spiel meine Schwierigkeiten. Kein Wunder bei einer solchen Story. Theoretisch war ja alles klar, aber wie soll man an Höhe gewinnen? Die Anleitung erwähnt nur so nebenbei, daß man auf Felsblöcke etwas draufstellen kann. Daß das auch Roboter einschließt und so ziemlich die einzige Art ist, um auf einen grünen Ast zu kommen, soll man anscheinend selbst rausfinden.

Jetzt ist es einfach die ersten 10 Ebenen zu bewältigen, doch sobald man es mit zwei, drei oder gar vier Sentinels zu tun bekommt, sieht man ohne eine gewisse Portion räumlichem Vorstellungsvermögen und einer Strategie ziemlich alt aus. Einen schwierigen Level könnte man beispielsweise in folgenden Schritten angehen:

1. Ausmachen seiner Position in der Ebene und relativ zu den Wächtern.
Mit der Übersicht der Ebene, wie sie vor Betreten eines Levels gezeigt wird kann man eigentlich erst etwas anfangen, wenn man bereits in ihm war. Legen Sie erst einmal alle Energie in einem Roboter auf möglichst vielen Blöcken an. Von diesem aus können Sie meistens Ihre Lage bestimmen, oder zumindest so viele markante Punkte ausfindig machen, daß Sie solches mit Hilfe der Übersichtskarte können. Ob Sie in dieser alle Sentinels gefunden haben, was mit einem Grünmonitor nicht gerade einfach ist, läßt sich durch Zählen der Geräusche überprüfen, die die Wächter bei jedem Drehen machen. Wichtig ist auch sich zu merken, aus welcher Ausgangsposition und in welcher Richtung sich die Sentinels drehen, doch leider kann das nicht immer bestimmt werden.

2. Planung des Angriffs.
Wenn Sie wissen wo in der Ebene Sie anfangen, können Sie daran gehen, Anhöhen zu suchen, die von dort aus gut erreichbar sind. Wählen Sie diejenige aus, die gerade abgetastet wird, da man bis zu jener dann noch Zeit hat zwischendurch ein paar Bäume zu absorbieren. Haben Sie es mit mehreren Sentinels zu tun, müssen Sie entsprechend kleinere Schritte planen, also nicht gleich auf eine Lage aus sein, von der aus man einen Wächter schlagen kann. Suchen Sie ersteinmal Lücken zwischen den Abtasträumen. Aus diesen ergibt sich dann meistens automatisch eine Route nach oben. Wichtig ist, immer ungefähr zu wissen, wo die Wächter ihre Augen haben, um sich immer rechtzeitig hinter ihren Abtastsektor setzen zu können. In manchen Leveln muß man warten, bis die Wächter über den Startpunkt weggezogen sind. Diese sollten Ihren Standpunkt natürlich nicht gerade einsehen können.

3. Ausführung und Verfeinerung des Plans.
So, jetzt wissen Sie wo Sie sind, wo Sie hin wollen, aber noch nicht wie. Fangen Sie einfach mal an. Wenn es irgendwann nicht mehr weiter geht, wissen Sie auf jeden Fall schon mal, wo Bäume zum absorbieren stehen, wo man sich beeilen muß und die Route ändern sollte. Bei dieser Methode muß man zwar öfters von vorne anfangen, aber da man, zumindest rein wissensmäßig gesehen, jedesmal ein Stück weiterkommt, erreicht man das Ziel mit etwas Übung doch meist recht schnell. Da Sie ja möglichst viele Level überspringen wollen, sollten Sie den Sentinel auf dem Turm erst ganz zum Schluß und von der höchsten Position aus absorbieren. Dem einen Wächter kann man gut aus dem Weg gehen, so daß man gemütlich alle Bäume mitnehmen kann. Sollten Sie doch einmal Energie abgenommen kriegen, bleibt sie ja doch in der Ebene.
Nun noch einiges zur Spielpraxis. Man sollte sich nicht nur Gedanken machen, wie man am besten nach oben kommt, sondern auch, ob man von dem neuen Standpunkt aus den alten Robby und die Blöcke darunter absorbieren kann. Da meistens mehrere Felder zur Auswahl stehen, ärgert man sich später nur, wenn man wegen einer Energie-Einheit festsitzt. Sollten Sie in einem Level mit zwei oder mehr Wächtern einmal erwischt worden sein, ist es besser, sich unter Energieverlust selber in Sicherheit zu bringen, als das Risiko auf sich zu nehmen, von einem Abtastalarm in den anderen »hyperraum gefahren« zu werden. Wenn Sie auf vielen Blöcken standen, können Sie so wenigstens einige Energie-Einheiten retten. Auch da gibt es einen Trick.

Daß der Sentinel Energie abzieht, kann man nicht verhindern. Da aber die helle Seite eines Blocks als erweiterte Quadratoberfläche interpretiert wird, kann man durch schnelles Drücken der Taste zum Absorbieren meistens nicht nur das, was auf dem Block steht rüberziehen, sondern auch den Block selbst. So geht von vier Einheiten nur eine verloren.

Es folgen nun die Codes und die Anzahl der Sentinels für die ersten 50 Ebenen, damit Sie ein bißchen üben können.

| Ebene | Code | Sentinels |
|---|---|---|
| 1 | 92416816 | 1 |
| 2 | 77651325 | 1 |
| 3 | 39380666 | 1 |
| 4 | 68888059 | 1 |
| 5 | 66799483 | 1 |
| 6 | 45600956 | 1 |
| 7 | 88070956 | 1 |
| 8 | 97558411 | 1 |
| 9 | 78959942 | 1 |
| 10 | 97567465 | 2 |
| 11 | 34128668 | 2 |
| 12 | 12837047 | 2 |
| 13 | 82497760 | 2 |
| 14 | 89619176 | 2 |
| 15 | 78844565 | 2 |
| 16 | 69076196 | 2 |
| 17 | 18167694 | 2 |
| 18 | 34259849 | 2 |
| 19 | 74546878 | 2 |
| 20 | 78042177 | 3 |
| 21 | 92261899 | 3 |
| 22 | 41108601 | 3 |
| 23 | 69780499 | 2 |
| 24 | 84620577 | 2 |
| 25 | 49636597 | 3 |
| 26 | 06268381 | 3 |
| 27 | 93449087 | 3 |
| 28 | 85414694 | 2 |
| 29 | 55711979 | 3 |
| 30 | 41062296 | 3 |
| 31 | 94185468 | 2 |
| 32 | 57640765 | 4 |
| 33 | 51837557 | 1 |
| 34 | 03549980 | 1 |
| 35 | 46855644 | 4 |
| 36 | 68261871 | 2 |
| 37 | 34454988 | 3 |
| 38 | 69096154 | 2 |
| 39 | 89515235 | 3 |
| 40 | 19390739 | 4 |
| 41 | 68195946 | 3 |
| 42 | 51957584 | 2 |
| 43 | 74666797 | 3 |
| 44 | 45888884 | 3 |
| 45 | 47577247 | 2 |
| 46 | 66526784 | 4 |
| 47 | 49877068 | 1 |
| 48 | 93458839 | 4 |
| 49 | 58764753 | 2 |
| 50 | 37411586 | 2 |

# Starglider

**Mit vereinten Kräften gegen die Ergons.**
**Hier einige Tips...**

Zuerst wollen wir uns mal den Starglider One vornehmen, denn auf die Dauer nervt es schon, regelmäßig von oben attakiert zu werden. Fliegen Sie dazu gleich nach dem Start in das Silo, das sich am linken Bildschirmrand befindet. Wenn man beim Eintritt leicht nach unten lenkt und nicht zu schnell fliegt, sollte es da keine Probleme geben. Dort erhalten Sie eine zweite Rakete, ohne die der Vogel nicht erlegt werden kann. Wenn Klein Hermann sich dann mal wieder zeigt, klemmen Sie sich dicht hinter das Flaggschiff und visieren das Heck an. Wenn es ruhig in der Mitte des Fadenkreuzes liegt, wird es Zeit, mit besten Grüßen eine Rakete auf den Weg zu schicken. Hat diese ihr Ziel getroffen, beginnt der Starglider One zu blinken. Wiederholen Sie das Manöver mit der zweiten Rakete. Danach haben sie einen Quälgeist weniger und 5000 Punkte mehr.

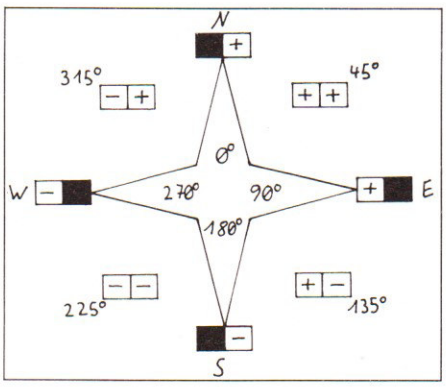
Aufmerksame Leser der Novelle, die übrigens wirklich lesenswert ist, werden nun fragen: So einfach? Schon alles fertig? Nein, ganz und gar nicht. Erst wenn Sie 10000 Punkte auf dem Konto haben sind Sie fertig und das auch nur mit dem ersten Level, denn dann geht alles von vorne los, mit dem Unterschied, daß nun zur Zerstörung der Gegner mehr Treffer benötigt werden.

Was beim Starglider anscheinend die größten Probleme hervorruft, ist das Aufladen der Energiezellen mit Hilfe der Starkstromleitungen. Wenn Sie eine solche Anordnung von drei Energietürmen gefunden haben, ist es ratsam nicht nur die Steuereinheit (Dach des einzelnen Turms) zu zerstören, sondern auch alle mit Raketen ausgerüsteten Gegner. Diese machen es nämlich fast unmöglich genau zwischen den Zwillingstürmen einzufädeln. Wenn die Gegend um die Leitungen sicher ist, manövrieren Sie das AGAV vor die Zwillingstürme und dann langsam und gefühlvoll, fast schon mit Bodenkontakt, durch diese hindurch. Wenn alles geklappt hat, sollte die Meldung »ENERGY CELLS REFUELLING« erscheinen und die Energieanzeige langsam auf 100% wandern. Wichtig ist nur, daß Sie auf der Bahn bleiben, auf der die Lichtblitze hin und her wandern. Wenn Sie wollen, können Sie sogar stehen bleiben. Die Hinweise des Playguids und der Novelle die darauf schließen lassen, daß man mit hoher Geschwindigkeit zwischen den Zwillingstürmen durch, auf den Dritten fliegen muß, dienen anscheinend nur zur Irreführung des Spielers. Schnelles Fliegen bringt auch im Kampf nichts. Fliegende Objekte verfolgt man am besten in deren eigenen Geschwindigkeit, Bodengerät sehr langsam oder stehend. Dabei solte immer ein Auge den Radar auf nahende Raketen absuchen. Die Silos sind übrigens in jedem Spiel an derselben Stelle. Ihre Koordinaten sind:

1) 48/53 (Start)
2) 92/61
3) 59/15
4) 11/85

Manchmal erhält man in den Silos die Koordinaten von nützlichem Gerät. Da das AGAV leider kein Instrument hat, das einen Zusammenhang zwischen den Himmelsrichtungen, die wiederum in Grad angegeben sind, und den Koordinaten herstellt, muß der Spieler sich selber helfen. Mit der untenstehenden Übersicht ist das Anfliegen bestimmter Koordinaten schon einfacher. Wenn Sie jetzt z.B. von 23/54 nach 97/09 fliegen möchten, bremsen Sie, drücken Pause und berechnen die Gradzahl folgendermaßen: links-plus, rechts-minus d.h. 135 Grad. Da links die Differenz größer ist, noch ein bißchen weniger, womit wir bei ungefähr 125 Grad wären. Im Stand in diese Richtung gedreht kommt der Flieger meist so genau an, daß man die Feinarbeit im Sichtkontakt mit dem Ziel erledigen kann. Die Geräte zum einsammeln, als da wären ein Rear-View-Scanning System, Raketen und Zusatzenergiekapseln, unterscheiden sich von den Ergons durch Blinken o.ä. und sind durch niedriges Überfliegen leicht aufzunehmen. Man sollte nur aufpassen, daß man nicht das ein oder andere Teil durch unkontrolliertes Rumballern vernichtet, denn so auffällig sind sie nun auch wieder nicht.

Nun noch zur Zerstörung der Walker und Stamper. Beide sind durch eine Rakete auf die Beine leicht zu zerlegen. Komisch, die Crew in der Story konnte die Viecher nicht mit Raketen kleinkriegen, dafür allerdings mit einem Laser. Mir geht es umgekehrt. Ich konnte noch keine Stelle finden an der eine der beiden Kampfmaschinen auf den Laser reagierte. Die kamen höchstens auf mich zu, was mitunter recht praktisch ist, denn dann kann man ihnen ohne großes Lenken eine Rakete verabreichen. Diese schießen Sie am besten ab, wenn Sie gerade eine Rakete des Ungetüms in die ewigen Jagdgründe geschickt haben, da sonst nur selbige getroffen wird. Die normalen Starglider kann man durch einige Treffer mit dem Laser beseitigen, genauso wie sonst alles andere auch. Die einzige Ausnahme machen da Gebilde, die wie Brücken aussehen. Sie sind anscheinend unzerstörbar. Ich glaube, sie stehen in Zusammenhang mit den Stromleitungen, da, wenn man gerade durch sie durchfliegt meistens eine solche auftaucht. Bliebe nur noch zu sagen, daß ich die Steuerung mit beweglichem Cursor für eleganter halte und Zentrierung schlichtweg unpraktisch finde. Das ist allerdings Geschmackssache.



# Die Axt im Haus erspart den Zimmermann

**Falsch geraten: Dieser Bericht ist keine Abhandlung über Sprichwörter im deutschen Sprachschatz, sondern er befaßt sich möglichst umfassend mit der Problematik der Disketten-Werkzeuge. Wir verstehen unter diesem Begriff nicht etwa Schraubendreher, Pinzette oder Lötkolben, sondern Software, die den Disketten-Benutzern ihre Arbeit erleichtern soll.**

## Was man vor dem Kauf beachten sollte

Ohne gutes Werkzeug ist auch der beste Handwerker nur die Hälfte wert. Deshalb sind gründliche Überlegungen bezüglich der Mindestanforderungen dringend erforderlich. Eine »eierlegende Wollmilchsau«, ein Programmpaket für alle Bedürfnisse gibt es nicht. Es ließe sich bei der Vielfalt der Wünsche auch nicht entwickeln und verkaufen. Deshalb muß eine Checkliste mit Fragen her:

- Welche Art von Programmen benötige ich? (Monitor, Formatierer, usw.)
- Welches Betriebssystem benutze ich am häufigsten? (AMSDOS oder CP/M)
- Wurden die Programme schon einmal in einer Zeitschrift getestet ? (Wer kauft schon gern die Katze im Sack?)
- Sind die Programme mit ausreichenden Anleitungen versehen?
- Haben sie eine anwenderfreundliche Benutzeroberfläche?
- Stimmt das Preis-/Leistungsverhältnis?
- usw., usw.

Diese Liste läßt sich bei Bedarf noch weiter ausbauen, sie soll an dieser Stelle auch nur als Anregung betrachtet werden. Sie werden nun sicherlich fragen: Was verstehen Sie eigentlich unter Disketten-Werkzeugen? Nun, hier folgt eine kleine Aufstellung der wichtigsten Utilities

- Diskettenmonitore, auch Diskettenscanner genannt.
- Formatierprogramme, auch zum Formatieren und Einstellen von Fremdformaten.
- Kopierprogramme von Kassette zu Diskette und Disk zu Disk.
- Verwaltungsprogramme, falls Sie mittlerweile mehr als eine Diskette, gefüllt mit Software, besitzen sollten.

## Bordwerkzeug

Im allgemeinen ist das Bordwerkzeug eines Automobils recht spartanisch, ähnlich sieht es auch mit DiskettenUtilities auf den Schneider Systemdisketten aus. Während Programme unter AMSDOS gänzlich fehlen. Bei den unter CP/M 2.2 oder CP/M Plus lauffähigen Programmen sieht es anzahlmäßig ein wenig besser aus, aber bezüglich Anwenderfreundlichkeit sind sie nicht gerade berauschend.

- DISCKIT2 / DISCKIT3 formatiert Disketten in allen vier Schneiderformaten, leider etwas langsam.
- PIP.COM kopiert Files von Disk zu Disk.
- DIR.COM zeigt das Inhaltverzeichnis der Diskette.
- SHOW.COM zeigt die noch freie Diskettenkapazität.
- RENAME.COM gibt Files neue Namen.
- ERASE.COM löscht Files auf der Diskette.

Vergeblich sucht man einen Diskettenmonitor, neben dem Formatierprogramm das wichtigste Werkzeug überhaupt. Neugierig geworden, fragen Sie nun: Alles schön und gut, aber wie müssen Disketten-Utilities nun eigentlich beschaffen sein? Welche Möglichkeiten sollten sie dem Anwender bieten? Wir werden versuchen, im folgenden Abschnitt diese Fragen an Hand von Beispielen zu beantworten.

## QUALITÄTSSOFTWARE – Forderungen und Beispiele

**Diskettenmonitor/ -scanner**
Eines der wichtigsten Werkzeuge für jeden Diskettenbenutzer, wenn nicht sogar das wichtigste, ist ein Diskettenmonitor, oder auch Diskettenscanner genannt. Ein guter Monitor besteht aus folgenden Teilwerkzeugen

- Directory-Editor
- Sektor-Editor
- Disk-Mapper
- Disk-Header

Mit Hilfe dieser Utilities dringen Sie bis in den entferntesten Winkel Ihrer Diskette vor und können Änderungen sowohl im Inhaltsverzeichnis, als auch in jedem anderen Teil der Diskette vornehmen. Bei der Arbeit mit einem Diskettenmonitor ist in jedem Fall höchste Vorsicht geboten. Nehmen Sie Änderungen nur an Softwarekopien vor, nie an Originalen. Denn schon eine kleine Fehlbedienung kann den Absturz Ihrer Diskette hervorrufen. Im schlimmsten Fall sind Teile des Inhaltsverzeichnisses oder der Programme zerstört und für alle Zeiten unwiederbringlich verloren. Ebenso wichtig ist es, sich vor jeder geplanten Änderung an Diskettensoftware den Inhalt des Directorys und der zu ändernden Programmteile ausdrucken zu lassen. So können Sie später immer noch den alten Zustand wiederherstellen, falls die Änderungen nicht nach Wunsch ausgefallen sind. Doch betrachten wir die einzelnen Werkzeugteile ein wenig genauer.

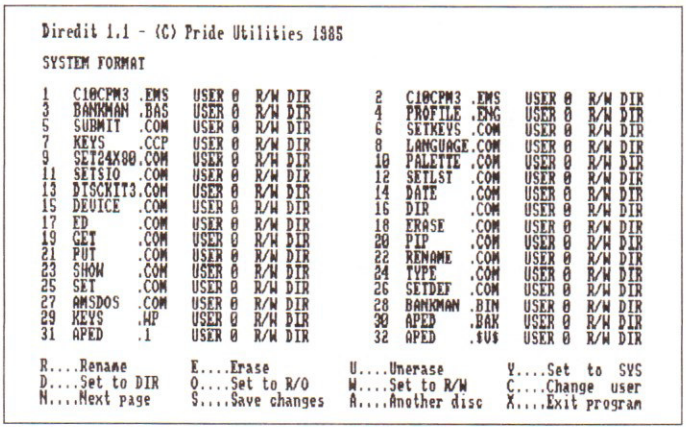



*Bild 1: Dir-Editor*

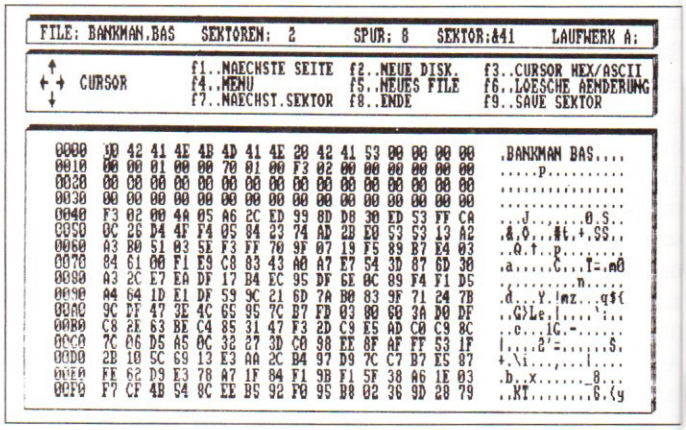



*Bild 2: Full-Screen, Disc-Editor*

## Directory-Editor

Der Directory-Editor listet den Disketteninhalt mit Filenamen, der User-Nummer, sowie dem jeweiligen File-Status auf. Aus dem File-Status können Sie erkennen, ob ein File sichtbar oder unsichtbar, schreibgeschützt oder nicht schreibgeschützt ist. In der Regel lassen sich aus einem Menü mittels definierter Tasten die folgenden Optionen auswählen:

- Filenamen ändern
- File löschen
- Gelöschtes File reaktivieren
- File unsichtbar machen (SYS)
- File wieder sichtbar machen (DIR)
- Schreibschutz aufheben (R/W)
- Schreibschutz setzen (R/O)
- User-Bereich ändern
- Geändertes Directory sichern

Ein Beispiel für eine gute Lösung zeigt die Hardcopy in Bild 1.

## File- / Sektor-Editor

File- bzw. Sektor-Editoren sind weitestgehend identisch aufgebaut. Nur der Einstieg, oder besser gesagt, die Suchkriterien sind unterschiedlich. Der File-Editor erwartet von Ihnen die Angabe eines Files, welches er auf der Diskette ausfindig machen soll. Findet er es, zeigt er auf dem Monitor den ersten Sektor dieses Files an. Beim Sektor-Editor müssen Sie dagegen die Spur und den Sektor angeben, den sie sich ansehen oder editieren wollen. In beiden Fällen erscheinen auf dem Bildschirm, in mehreren Fenstern folgende Informationen:

Status:
mit Angabe des Laufwerkes, der Spur, des Sektors, des Filenamens und der Länge des Files in Sektoren.
Sektorinhalt:
Nummerierung der Bytes, Inhalt des Sektors in Hex-Code und in ASCII-Zeichen.

Bedienung des Editors mittels definierter Tasten:

- Cursorsteuerung
- Sektorauswahl
- Fileauswahl
- Umschaltung des Cursors zwischen HEX und ASCII
- Änderungen rückgängig machen
- Änderungen sichern usw.

Einen Full-Screen-Editor mit Cursorsteuerung und den oben genannten Möglichkeiten zeigt die Hardcopy in Bild 2. Eine Änderung des Sektorinhaltes kann sowohl im HEX-, als auch im ASCII-Teil erfolgen, der jeweils andere Teil wird ebenfalls automatisch geändert.

## Disk-Mapper

Ein Disk-Mapper dient zur Unterstützung eines Sektor-Editors. Er zeigt an, auf welchen Spuren und in welchen Sektoren sich die Teile der Files auf der Diskette befinden. Hierzu ein Beispiel:

| NAME | TRACK / SEKTOR |
|---|---|
| MYPROG.BAS | 0,C4 0,C5 0,C6 1,C5 1,C6 1,C7 1,C8 |
| HELP.DAT | 1,C9 2,C0 |
| usw. | |

Sehr gute Dienste leistet der Disk-Mapper beim Suchen nach bestimmten Programmteilen, z.B. um ein englisches Programm mit einer deutschen Benutzeroberfläche zu versehen. Das mühsame Umrechnen der Blöcke aus dem Directory kann entfallen.

## Disk-Header

Ein Disk-HEADER listet folgende Daten eines Programms auf:

- Name
- Startadresse
- Länge in Byte
- Type (Basic, MCode, ASCII-File)
- Execute Adresse (bei Programmen, die nachgeladen werden)

Auch hierzu ein kurzes Beispiel:

| Name | Start | Size | Type | Exec |
|---|---|---|---|---|
| HALLO.BAS | 0170 | 1000 | BASIC | |
| HALLO.BIN | A123 | 8FFF | MC | 8200 |
| HALLO.DAT | 3000 | 05FF | ASCII | |

Dies waren in Kürze einige Beispiele guter Diskettenmonitore. Wir werden später in einer kleinen Übersicht einige empfehlenswerte Programmpakete nennen.

## Formatierprogramme

Da Disketten in der Regel unformatiert verkauft werden, benötigt jeder User ein Formatierprogramm. Zwar wird auf der Schneider-Systemdiskette für die Betriebssysteme CP/M 2.2 und CP/M + Disckit2, bzw. Disckit3 mitgeliefert. Für den Anfang reichen diese Programme aus, aber wer möchte nicht auch unter AMSDOS und dann nach Möglichkeit in der halben Zeit seine Disketten formatieren. Ein brauchbares Formatierprogramm muß je nach Einsatz alle Formate, die der CPC oder Joyce kennt, beherrschen. Gut darf sich ein Programm nennen, das außerdem die Spuren 40–42 formatiert und zur Verfügung stellt. Dies ergibt dann eine Speicherkapazität von bis zu 208k pro Diskettenseite. Mit einigen speziellen Formatierwerkzeugen lassen sich aber auch Fremdformate einstellen und formatieren. So kann ein Datenaustausch auch zwischen Benutzern unterschiedlicher Computersysteme stattfinden. Im Zusammenspiel mit einem 5.25-Zoll Laufwerk mit 80 Spuren und 2 Köpfen erreicht man dann eine Speicherkapazität von rund 800k, und dies ohne irgend einen Hardwareeingriff. Bild 3 zeigt das Menue eines Formatierprogramms der gehobenen Klasse.

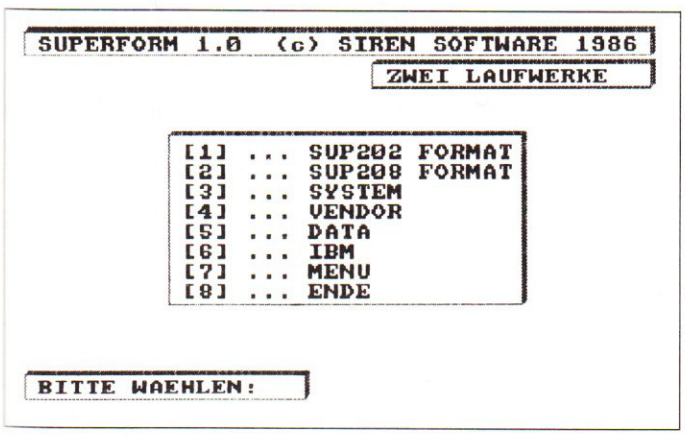



*Bild3: Formatierungsprogramm*

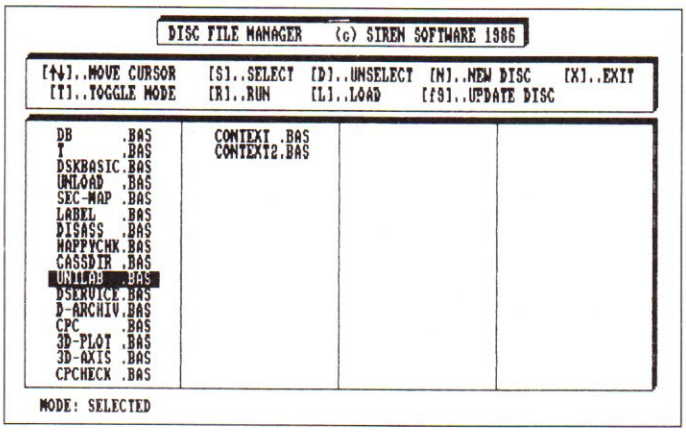



*Bild4: Menue-Maker*

## Kopier-Werkzeuge

Oft ist es notwendig, daß sich der Anwender Arbeitskopien seiner Originale ziehen muß, z. B. bei Anpassung an sein Computersystem oder seinen Drucker. Verwendet er seine Originaldiskette auch als Arbeitsdiskette, so läuft er Gefahr, sie irgendwann durch Fehlbedienung zu zerstören. Disketten-Kopierprogramme bieten folgende Möglichkeiten:

- Übertragung von ausgewählten Files oder Filegruppen bei Einsatz von sogenannten »wildcards«.
  Beispiele:
  MYPROG.BAS (nur dieses File)
  *.BAS (alle Basic-Files)
  *.*(alle Files auf der Diskette)
- Backup einer Diskette
  Eine 1:1 Kopie der Quelldiskette einschließlich des Formates wird auf der Zieldiskette erstellt. Einige Programme übertragen auch unformatierte Spuren und gelöschte Files.

Wer von der Kassette als Speichermedium auf Diskette umsteigen will, wem das Laden der Kassettensoftware zu lange dauert oder wem die 3-Zoll Disketten als Archiv-Speichermedium zu teuer sind und doch lieber Kassetten bevorzugt, der benötigt Software für den Transfer von Kassette auf Diskette und umgekehrt. Auch diese Programme können verschiedene Möglichkeiten beinhalten:

- Automatische oder manuelle Übertragung
- Übertragung einzelner Files
- Übertragung der kompletten Kassette oder Diskette
- Auswahl der Übertragungsgeschwindigkeit
- Aufhebung oder Umgehung eines Kopierschutzes
- automatische Anpassung des File-Namens bei der Übertragung von Kassette auf Diskette
- Möglichkeit für den Anwender, den File-Namen selbst zu ändern usw.

Das Thema »Kopieren von geschützter Software« wollen wir nicht behandeln. Es würde den Rahmen dieses Artikels sprengen.

| FILENAME | DISKNAME | US | BLO | TYP | ST | LG | EN | BEMERKUNG |
|---|---|---|---|---|---|---|---|---|
| DATAFLEX.BAS | bas-1 | A | 00 | 015 BASIC | 0170 | 3918 | 0000 | |
| DIRDOK .BAS | bas-1 | A | 00 | 010 BASIC | 0170 | 276E | 0000 | |
| DISCODE .BIN | bas-1 | A | 00 | 008 BINAER | 8000 | 1D00 | 0000 | |
| DISKDOK .BAS | bas-1 | A | 00 | 007 BASIC | 0170 | 18FA | 0000 | |
| DISKMAT .BAS | bas-1 | A | 00 | 018 BASIC | 0170 | 442B | 0000 | |
| DISKPATC.BAS | bas-1 | A | 00 | 031 BASIC | 0170 | 7AED | 0000 | |
| HEXMON .BAS | bas-1 | A | 00 | 014 BASIC | 0170 | 355A | 0000 | |
| KALENDER.BAS | bas-1 | A | 00 | 008 BASIC | 0170 | 1D5E | 0000 | |
| MENU .BIN | bas-1 | A | 00 | 004 BASIC | 60C3 | 0C70 | 0170 | |
| NAME .INF | bas-1 | A | 00 | 001 ASCII | 0000 | 0000 | 0000 | |
| NOTEBOOK.BAS | bas-1 | A | 00 | 006 BASIC | 0170 | 14F6 | 0000 | |
| NURSE .BAS | bas-1 | A | 00 | 011 BASIC | 0170 | 28BA | 0000 | |
| SYNAID .BIN | bas-1 | A | 00 | 001 BINAER | A000 | 0090 | 0000 | |
| SYNOPSIS.BAS | bas-1 | A | 00 | 014 BASIC | 0170 | 3608 | 0000 | |
| TAPEMON .BAS | bas-1 | A | 00 | 006 BASIC | 0170 | 13DD | 0000 | |
| UDG .BAS | bas-1 | A | 00 | 012 BASIC | 0170 | 2E64 | 0000 | |
| VARDUMP .BAS | bas-1 | A | 00 | 009 BASIC | 0170 | 21E6 | 0000 | |
| 3D-AXIS .BAS | bas-1 | B | 00 | 027 BASIC | 0170 | 6AC7 | 0000 | |
| 3D-PLOT .BAS | bas-1 | B | 00 | 013 BASIC | 0170 | 31A0 | 0000 | |
| CASSDIR .BAS | bas-1 | B | 00 | 003 BASIC | 0170 | 0B74 | 0000 | |
| CONTEXT .BAS | bas-1 | B | 00 | 009 BASIC | 0170 | 22D3 | 0000 | |
| CONTEXT .BIN | bas-1 | B | 00 | 004 BINAER | 9300 | 0D00 | 0000 | |
| CONTEXT .HLP | bas-1 | B | 00 | 003 ASCII | 0000 | 0000 | 0000 | |
| CONTEXT2.BAS | bas-1 | B | 00 | 010 BASIC | 0170 | 23E5 | 0000 | |
| CPC .BAS | bas-1 | B | 00 | 010 BASIC | 0170 | 26EA | 0000 | |
| CPCHECK .BAS | bas-1 | B | 00 | 002 BASIC | 0170 | 047A | 0000 | |
| D-ARCHIV.BAS | bas-1 | B | 00 | 008 BASIC | 0170 | 1CDE | 0000 | |
| DB .BAS | bas-1 | B | 00 | 001 BASIC | 0170 | 0034 | 0000 | |
| DB .BIN | bas-1 | B | 00 | 002 BINAER | 8000 | 0732 | 0000 | |
| DISASS .BAS | bas-1 | B | 00 | 015 BASIC | 0170 | 3805 | 0000 | |
| DISK-RSX.BIN | bas-1 | B | 00 | 001 BINAER | A000 | 0120 | 0000 | |
| DSERVICE.BAS | bas-1 | B | 00 | 013 BASIC | 0170 | 3329 | 0000 | |
| DSKBASIC.BAS | bas-1 | B | 00 | 001 BASIC | 0170 | 0034 | 0000 | |
| HAPPYCHK.BAS | bas-1 | B | 00 | 002 BASIC | 0170 | 0649 | 0000 | |
| LABEL .BAS | bas-1 | B | 00 | 003 BASIC | 0170 | 0B3C | 0000 | |
| MENU .BIN | bas-1 | B | 00 | 004 BASIC | 60C3 | 0C70 | 0170 | |
| SEC-MAP .BAS | bas-1 | B | 00 | 002 BASIC | 0170 | 0656 | 0000 | |
| T .BAS | bas-1 | B | 00 | 011 BASIC | 0170 | 2944 | 0000 | |
| TEXTPROG.BIN | bas-1 | B | 00 | 014 BINAER | 3E00 | 33B1 | 0000 | |
| UNILAB .BAS | bas-1 | B | 00 | 016 BASIC | 0170 | 3CB4 | 0000 | |
| UNLOAD .BAS | bas-1 | B | 00 | 002 BASIC | 0170 | 0673 | 0000 | |

*Bild 5: Ausdruck einer Directory-Datei*

## Diskettenverwaltung

Unter diesen Oberbegriff fallen alle Programme – zur Generierung eines DiskettenMenues. Die Daten werden dem Directory der Diskette entnommen.

- zur Archivierung und Bearbeitung von Disketten-Inhaltsverzeichnissen mit Editier-, Sortier-, Such-, List- und Druckroutinen

Ein Menuegenerator erweist sich als nützliches Instrument bei der Erstellung von:

- Arbeitsdisketten mit Dienstprogrammen
- Disketten mit Nachladeprogrammen

Während der Installierungsphase kann der Anwender Programme oder Textdateien sichtbar oder unsichtbar machen. Die unsichtbaren Dateien oder Programme erscheinen später nicht mehr im Inhaltsverzeichnis. Das Bild 4 zeigt ein Beispiel eines MENUE-MAKERS. Ist das Menü installiert, kann der Anwender mit Hilfe der Cursor-Tasten ein beliebiges Programm aus dem Menue anwählen und durch Druck auf eine festgelegte Taste, z.B. <RETURN> wird das Programm geladen und gestartet. Das Menue kann jederzeit geändert und erneuert werden.

Im Laufe der Zeit wächst bei jedem User die Zahl der Disketten, vollgestopft mit einer Vielzahl von Programmen und Dateien. Irgendwann ist der Zeitpunkt erreicht, daß man den Überblick verliert. Ein gutes Archivprogramm muß her. Die folgende Aufstellung zeigt einige Möglichkeiten die ein solches Programm bieten kann:

- Automatische Aufnahmen der Disketten-Inhaltsverzeichnisse in eine Datei
- Löschen und Updaten von Directorys in der Datei
- Sortieren nach verschiedenen Gesichtspunkten
  * alphabetisch
  * nach Diskettennummern
  * nach Filetypen (ASCII, BASIC, MCODE, usw.)
  * und anderen
- Suche nach bestimmten Programmen
- Einfügen von Kommentaren
- Ausgabe auf Monitor und Drucker

Im Bild 5 finden Sie den Ausdruck einer Archivdatei alphabetisch sortiert).

## Werkzeuge (fast) zum Nulltarif

PUBLIC DOMAIN–Software ist hiermit gemeint. Unter den Betriebssystemen

CP/M und CP/M Plus werden einige gute Disketten-Werkzeuge für wenig Geld angeboten. Betrachten wir kurz die Vor- und Nachteile von Public Domain Software:

**Vorteile:**

- geringer Anschaffungspreis
- in fast allen Fällen wird eine gute Anleitung auf der Diskette mitgeliefert
- neben den ablauffähigen Files enthält die Diskette oft den Source-Code in Assembler, BASIC oder PASCAL. Der Anwender kann sein Programm entsprechend anpassen.

**Nachteile:**

- oft sind die Anleitungen in englischer Sprache
- man kauft die Katze im Sack, da ausführliche Beschreibungen vorab nicht von den Vertreibern herausgegeben werden
- nicht alle CP/M-Programme laufen ohne Anpassungen auf den CPCs oder auf dem Joyce

Wer gern bastelt oder experimentiert, oder sich in den Programmiersprachen und der Maschinenebene seines Computers gut auskennt, für den ist die Public Domain-Software eine wahre Fundgrube.

Zum Abschluß diese Abschnittes einige Beispiele brauchbarer Public Domain-Software:

| | |
|---|---|
| DU | Disketten-Monitor. Es gibt ihn in verschiedenen Versionen. |
| WASH-10 | Komfortabler Diskettenmanager |
| COPYFST3 | Sektorweises Kopieren |
| DIRLST/1 | schreibt Disketten-Directory in eine Datei. |
| BACKUP | (erklärt sich selbst) |
| SD | noch ein SUPER DIRECTORY |
| DUTIL | Diskettenmonitor |
| MFT50 | Kopierprogramm |
| FIND40 | sucht Zeichenketten in Dateien |
| SWEEP | Diskettenmanager |
| CRCBUILD | erstellt Catalog-Dateien |
| SUPERZAP | Diskettenmonitor |
| UNERA+ | stellt gelöschte CP/M+ Dateien wieder her |
| DP | Diskpatch-Programm |
| SCAN | sucht defekte Sektoren |

Folgende Programme laufen nur unter Turbo-Pascal:

| | |
|---|---|
| CHG | wechselt Diskettenformat |
| DISKDEF | Fremdformat-Bearbeitung |
| DSKDMP | Disketten-Dump |

## Spezialwerkzeuge – was bietet der Softwaremarkt

Der letzte Abschnitt befaßt sich mit mit dem Angebot von Diskettenwerkzeugen auf dem deutschen Softwaremarkt. Unsere Übersicht stellt nur einen kleinen Ausschnitt dar, denn täglich kommt neue Software hinzu. Sie werden feststellen, daß in keinem Programmpaket alle Möglichkeiten geboten werden.

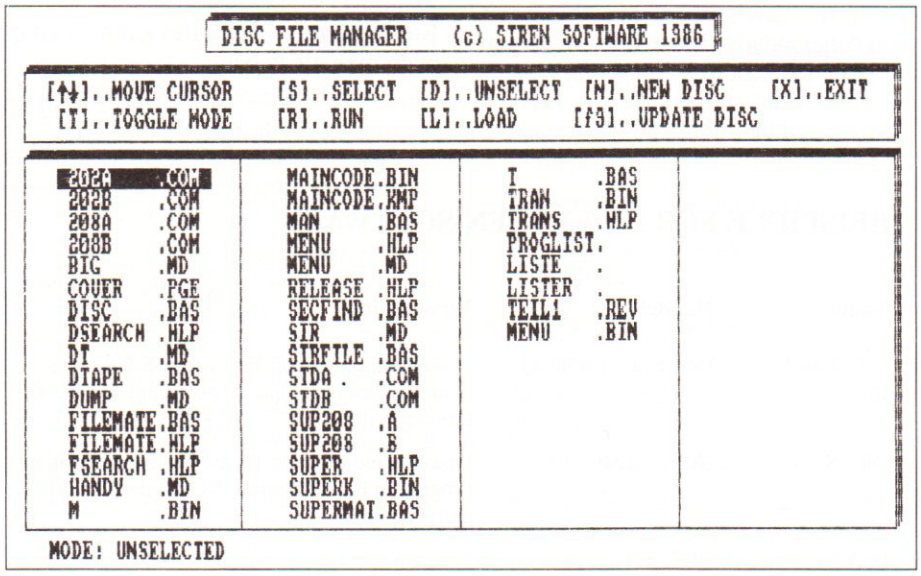
Ein Programm, was alles kann, wird es auch niemals geben.

(H.W. Fromme)

### BEISPIELE FÜR DISKETTEN-SOFTWARE

| Name | Hersteller | Verwendung |
|---|---|---|
| RH-DMON | (Integral Hydraulik) | Diskettenmonitor mit Fullscreen-Editor, Formatierprogramm, Diskettenreorganisation und Druckroutine. |
| Disc-Scanner | (RSE Software) | Diskettenmonitor mit Druckroutine, Formatierprogramm für 43 Spuren, Fileheader, Disk-Kopierprogramm |
| Disk-Sorter | (RSE Software) | Dateiverwaltung zur Aufnahme und Editierung von Diskettendirectories, Sortierroutine für alphabetisches Sortieren, Druckroutine |
| F.I.D.O. | (Pride Utilities) | Dateiverwaltung für Diskettendirectories mit Druckroutine, Menuegenerator Besonderheit: Passwortschutz |
| DISKPARA | (Strauß Elektronik) | Formatierprogramm, erstellt beliebige Fremdformate und stellt Zweitlaufwerk auf Fremdformate ein. Benötigt CP/M 2.2 oder CP/M Plus. |
| MASTERCOPY | (Peter Mandrella) | Menuegesteuertes Kopierprogramm für Diskettensoftware, unterstützt zwei Laufwerke, ist selbst gegen kopieren geschützt. |
| OAX | | Programm zum Übertragen geschützter (Speedlock) Kassettensoftware. |
| CLONE | | Programm zur Übertragung geschützter Diskettensoftware. |
| Copy Master | (RSE Software) | Transferprogramm für Diskette und Kassette, Backup-Routine auch auf Kassette, Druckroutine, Programmverschiebung möglich. |
| HANDY MAN | (SIREN) | Sehr schnelles Formatierprogramm bis 208k, unterstützt zwei Laufwerke unter AMSDOS und CP/M. Beinhaltet weiterhin Menügenerator, Diskettenmonitor mit DISCSEARCH und FILESEARCH. Sehr gute Benutzeroberfläche. |
| MASTERDISC | (SIREN) | Zwölf Diskettenutilities, File-Copy, Disketten-Backup, Directory-Editor, Disketten-Monitor und Editor, schnelles Formatierprogramm, gelöschte Files können wieder sichtbar gemacht werden. Gute Benutzeroberfläche. |
| Discovery Plus | | Kopiert geschützte Software von Kassette auf Diskette. Vier Kopierprogramme für z.B. Speedlock, Headerlos, usw. User Service: Discovery Plus wird laufend überarbeitet. Für Problemfälle werden Anleitungen herausgegeben. |
| LOCKSMITH | | Kopier-Utilities für Kassetten und Disketten-Software. Besonderheit: Locksmith kann in eigene Benutzeroberflächen eingebunden werden. |

# DREI AUF EINEN STREICH




## – HANDY MAN
## – MASTERDISC
## – DISCOVERY PLUS

**Drei Programmpakete von SIREN für alle CPCs liegen diesmal auf dem Untersuchungstisch. Lassen Sie sich überraschen:**

Hersteller: SIREN
Vertrieb: PR8-SOFT Margetshöchheim
Steuerung: Tastatur
Monitor: Farbe/Grün
Datenträger: Diskette 3-Zoll
Programme: Basic / MC
Betriebssystem: AMSDOS
Preis pro Paket: 59,90 DM
CPC 464, 664, 6128
mit mindestens einem Laufwerk

### HANDY MAN
### – Sieben Disketten-Utilities mit deutscher Benutzeroberfläche

Nach dem Start der Diskette erscheint auf dem Monitor das Hauptmenü mit folgenden Unterprogrammen:

- SUPERFORM
- TRANS FILE
- DISC RELEASE
- DISC SEARCH
- FILE SEARCH
- FILE MATE
- MENÜ MAKER
- HANDBUCH
- ENDE

Nach Auswahl einer der ersten 7 Möglichkeiten kann ein zum jeweiligen Programm gehörender Hilfstext in deutscher Sprache über den Monitor ausgegeben werden. Mittels der 8. Möglichkeit lassen sich alle Hilfstexte über den Drucker als Handbuch ausgeben. Ein toller Service, oder?

Doch befassen wir uns nun genauer mit den einzelnen Utilities: Neben den Standardformaten DATA, SYSTEM, VENDOR und IBM bietet SUPERFORM ein Spezialformat an, welches die Diskettenkapazität auf 202 bzw. 208 KByte erweitert. Nutzbar sind die so formatierten Disketten sowohl unter AMSDOS als auch unter CP/M. Die jeweilige Software zum Initialisieren und Rücksetzen wird für den Gebrauch unter AMSDOS mit auf die Diskette kopiert. Bei Verwendung unter CP/M muß dies manuell erfolgen.

Die Formatiergeschwindigkeit ist beachtenswert. Für eine Diskettenseite benötigt SUPER FORM nur ca. 18 Sekunden, DISCKIT3 auf der Systemdiskette dagegen stolze 34 Sekunden. Formatierfehler haben wir trotz vieler Versuche nicht feststellen können.

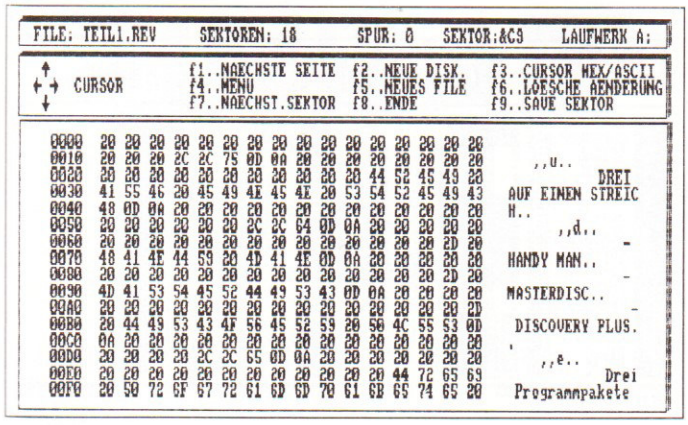
Wer trotz SUPER FORM immer noch nicht genügend Diskettenkapazität zur Verfügung hat oder wer sein Archiv mit Kassetten aufbauen möchte, dem kann mit DISK RELEASE geholfen werden. Es bietet dem Anwender die Möglichkeit, seine auf Diskette gespeicherten Programme oder Texte auf Kassette zu kopieren. Dies funktioniert jedoch nur mit Disketten in den AMSDOS-Formaten und SUPERMAT. Dafür wird aber das jeweilige Format mit auf Kassette gespeichert und beim Rückladen von Kassette auf Diskette diese gleich wieder richtig formatiert. Abschließend sei noch erwähnt, daß beim Rückladen sowohl ganze Kassetteninhalte, als auch einzelne Files überspielt werden können.

Aber auch für den Transfer von Programmen und Texten auf und von Disketten ist gesorgt. Mit Hilfe von TRANS FILE lassen sich entweder ganze Disketten oder auch einzelne Files überspielen. Einige Besonderheiten möchten wir Ihnen nicht vorenthalten:

- Vor einem Transfer können Sie die USER-Nr. wechseln.
- Unsichtbare Files müssen vor dem Transfer sichtbar gemacht werden.
- R/O Files werden zu R/W Files, geschützte Programme bleiben geschützt.

Mit den Programmen FILE SEARCH und DISC SEARCH stellt HANDY MAN zwei fast identische, aber ausgezeichnete Diskettenmonitoren zur Verfügung. Wollen Sie ein File auf der Diskette inspizieren oder editieren und Sie kennen den Filenamen, dann ist FILE SEARCH das richtige Werkzeug. Suchen Sie jedoch einen bestimmten String oder eine HEX-Zeichenfolge, dann benötigen Sie DISC SEARCH. In beiden Fällen erhalten Sie nach erfolgreicher Suche auf dem Monitor auf drei Fenstern:

- den jeweiligen Diskettenstatus mit Laufwerk-Nr., Spur- und Sektornr. Bei FILE SEARCH wird zusätzlich der Filename und die Anzahl der Sektoren des Files angegeben.
- Die Funktionstastenbelegung zur Bedienung
- bei FILE SEARCH den Inhalt des 1. Sektors des gesuchten Files, bei DISC SEARCH den Sektor, in dem der gesuchte String gefunden wurde.







Mit Hilfe der Funktionstasten kann der Diskетteninhalt weiter untersucht, geändert und wieder abgespeichert werden. (Bild HANDY-2.PIC)

FILE MATE ist ein Ausgabeprogramm für reine ASCII-Files auf Monitor oder über Drucker.

Das letzte Utility der Sammlung, MENÜ MAKER, ermöglicht es Ihnen, Ihre Disketten mit einem eigenen Menü zu versehen. Sie können Unterprogramme, die nachgeladen werden ,unsichtbar machen, sowie angewählte Programme laden oder starten. Weiterhin ist es jederzeit möglich, das Menü zu ändern und es erneut auf der Diskette abzuspeichern.
(Bild HANDY-1.PIC)

Fassen wir zusammen:
HANDY MAN von SIREN mit der deutschen Benutzeroberfläche von PR8-SOFT ist ein ausgezeichnetes Diskettenwerkzeug für den CPC-Anwender und seinen Preis wert. Hoffentlich macht das gute Beispiel mit eingebauten Hilfstexten und der Möglichkeit, sich sein Handbuch selbst zu drucken, auf dem deutschen Markt Schule.

| | |
|---|---|
| Nutzeffekt: | 1 |
| Bedienerfreundlichkeit: | 1 |
| Dokumentation: | 1 |
| Arbeitsgeschwindigkeit: | 1 |
| Preis/Leistung: | 2 |

## MASTERDISC – Ein Programmpaket mit 12 interessanten Disketten-Utilities

Auch MASTERDISC, der ältere Bruder von HANDY MAN, ist schon mit einer deutschen Benutzerführung, mit deutschen Hilfstexten und der Option HANDBUCH ausgestattet. Der erste Eindruck nach dem Start des Programms ist vielversprechend, denn im Hauptmenü hat der Anwender die Qual der Wahl zwischen den im nachfolgenden Bild aufgeführten Disketten-Hilfsprogrammen. (Bild MASTER-1.PIC)

Der DIRECTORY EDITOR bietet folgende Möglichkeiten:

- Files retten
- Files auf SYS oder DIR setzen
- Files löschen
- Files umbenennen
- USER-Nr. ändern
- Löschschutz setzen (R/O) oder entfernen (R=W)

Auf dem Monitor wird jeweils eine Hälfte des Directory-Inhaltes (32 Eintragungen) angezeigt. Zusätzlich erscheint in einem seperaten Fenster die Tastenbelegung für die Bedienung dieses Programmteils.
(Bild MASTER-2.PIC)

SPEEDFORM nennt sich ein Formatier-Utility, mit dem Sie innerhalb von 18 Sekunden pro Diskettenseite in allen 4 Schneider Formaten (System, Data, Vendor und IBM) Ihre Disks formatieren können. Sie haben auch Zugriff auf ein eventuell angeschlossenes zweites Laufwerk, 3-Zoll oder 5.25-Zoll.

Der SECTOR EDITOR ist eine notwendige und gute Ergänzung des DIRECTORY EDITORS. Mit diesem ausgezeichneten Diskettenmonitor läßt sich jeder Sektor jeder Spur von Laufwerk A: oder B: laden, editieren und wieder speichern. Die Benutzerführung erfolgt wie bei dem DIRECTORY EDITOR.
(Bild MASTER-3.PIC)

Mit DEPROTECTOR lassen sich Basic-Programme, die mit dem 'P'-Listschutz versehen worden waren, wieder listen. Der Listschutz wird entfernt.

Die wichtigsten Header-Daten von Programmen auf Diskette oder Kassette erhält man mit Hilfe der Utilities DISC HEADER, bzw. TAPE HEADER.

Es sind bei beiden Programmen:

- Programmname
- Startadresse
- Programmlänge
- Typ (Basic, MCode, ASCII)
- Execute Adresse

Zusätzlich gibt TAPE HEADER noch die Information, ob es sich um ein geschütztes Programm handelt.

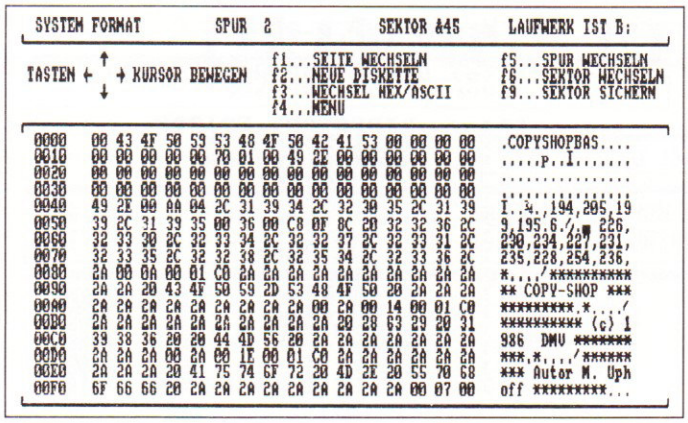
Für den Transfer von Files von und auf Disketten empfiehlt sich TRANS DISC. Es unterstützt zwei Laufwerke, ist menügesteuert und vergleichbar mit PIP.COM auf der Systemdiskette. TRANS TAPE benötigt man zum Überspielen von Daten und Programmen von Diskette auf Kassette. Auch hier können beide Laufwerke als Quelle benutzt werden. Das Programm fordert den Anwender zu allen notwendigen Eingaben auf.

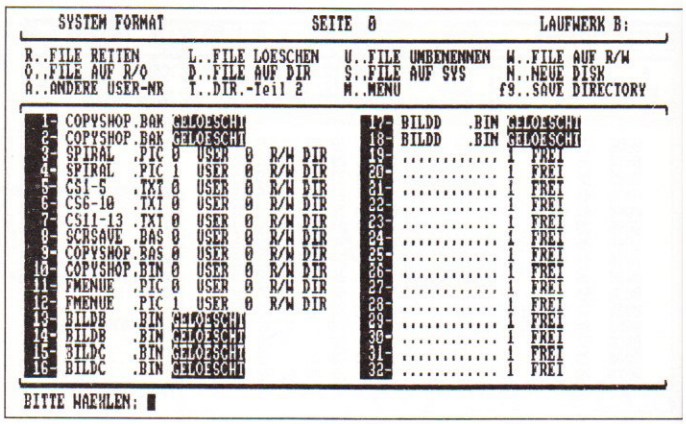
Sehr mühsam ist das Aufsuchen von einzelnen Programmteilen auf der Diskette. DISC MAP kann hier helfen. Es zeigt an, auf welchen Spuren und in welchen Sektoren die einzelnen Files abgespeichert worden sind. Ein Beispiel:

PICTURE.BAS 0,C6 0,C7 1,C6 1,C7
1,C8 1,C9 usw.

Auch an die Ausgabe von Textfiles und Programmen wurde gedacht. So lassen sich ASCII-Files über Monitor und Drucker mit Hilfe von TYPE FILE ausgeben. Steuerrcodes können jedoch zu Schwierigkeiten führen, wie Sonderzeichen aus Textsystemen, wie WORDSTAR, TASWORD usw.

Einen Totalabsturz vermeidet man, indem man die Druckoption wirklich nur bei einem angeschlossenen Drucker anwählt. Das Ausgabe-Utility für Programme nennt sich DUMP FILE und gibt den Fileinhalt in HEX und ASCII aus.




Falls Ihnen die Diskettenzugriffszeit zu lang ist, können Sie mit Hilfe von ZIP DISC um bis zu 20% verkürzen.

Das letze Utility von MASTER DISC nennt sich IMAGE. Es gibt Ihnen die Möglichkeit Backup-Kopien von Ihren Disketten anzufertigen. Auch hier werden beide Laufwerke unterstützt, eine Formatierung erfolgt automatisch.

Fazit: Auch MASTER DISC ist eine gute Sammlung von Disc-Utilities für den Einsteiger und den fortgeschrittenen Nutzer eines CPC.

Positiv viel uns, wie schon bei HANDY MAN, die deutsche Benutzerführung und das integrierte Handbuch auf.

| | |
|---|---|
| Nutzeffekt: | 2 |
| Bedienerfreundlichkeit: | 1 |
| Dokumentation: | 2 |
| Arbeitsgeschwindigkeit: | 2 |
| Preis/Leistung: | 3 |

## DISCOVERY PLUS Programm-Sammlung für das Kopieren von Kassette auf Diskette

Bei dem dritten Programmpaket aus dem Hause SIREN handelt es sich um Software zum Übertragen von Kassettenprogrammen von Kassette auf Diskette. Auf der 3-Zoll Diskette befinden sich vier Programme, mit denen sich lt. Angabe von PR8-Soft, dem deutschen Vertrieb, ein Großteil der geschützten Kassettensoftware auf Diskette übertragen läßt.

Ein Programm dieser Sammlung ist das altbekannte DISCOVERY. Für alle, die es noch nicht kennen, hier eine kurze Beschreibung:

Discovery entfernt automatisch den Listschutz von Basic-Programmen. Dadurch lassen sich leicht Änderungen durch den Anwender durchführen.
Man hat die Wahl zwischen einen Automatik- und einem Manuel-Modus. Im letzteren kann sowohl der Programmname vor dem Abspeichern auf Diskette geändert, als auch ein Relocator zum Verschieben des Programms im Speicher hinzugefügt werden.
Der Relocator wird dann notwendig, wenn das zu kopierende Programm den Speicherbereich der Diskettenverwaltung überschreiben würde. Während des Kopiervorgangs erhält der Anwender auf seinem Monitor die Header Daten der kopierten Blöcke. Erfahrungsgemäß leistet DISCOVERY gute Hilfe bei älterer Software mit einem nicht allzu komplizierten Kopierschutz.
SPLOCK TRANS wird vom Entwickler für den Transfer von Programmen empfohlen die mit dem »SPEEDLOCK«-Verfahren geschützt sind.
Vor dem Starten des Rekorders muß eine formatierte Diskette mit noch min. 52K Speicherkapazität in das Laufwerk A: eingelegt werden. Nach dem Start läuft das Kopierprogramm automatisch ab. Hat man Erfolg, d.h., die Kopierroutinen wurden nicht durch das geladene Kassettenprogramm überschrieben, erfolgt nach dem Laden sofort das Speichern auf Diskette.
Wie der Anwender ein Überschreiben der Kopiersoftware umgehen kann, ist uns leider nicht bekannt.
Die beiden übrigen Programme TD.BAS und TD1.BAS für normale, headerlose und »anders geschützte« Software müssen vor dem ersten Start initialisiert werden. Leider fehlt uns hier ebenfalls die Information, um welchen Schutz es sich handelt.

Mit TD.BAS oder TD1.BAS kopierte Programme können nur von diesen aus wieder geladen und gestartet werde. Alle notwendigen Anleitungen erhält der Anwender während des Programmablaufs.

Eine ausführliche Erläuterung der Programme befindet sich als Textfile auf der Diskette und kann auch über den Drucker ausgegeben werden.

Bei Kopierversuchen mit einigen neuen Spielen konnten wir mit keinem der vier Programme einen Erfolg verbuchen. Der Schutz der Kassettensoftware ist mittlerweile so ausgefeilt worden, daß die Entwickler von Transferprogrammen nicht nachkommen. Nach unseren Informationen plant SIREN zur AMSTRAD USER SHOW die Veröffentlichung einer neuen Version von DISCOVERY PLUS. Ob und zu welchen Konditionen die Anwender der alten Programmsammlung diese neue Version beziehen können, ist noch nicht bekannt. Einige Hilfen stellt PR8-Soft dem DISCOVERY PLUS Anwender heute schon zur Verfügung.

Der Diskette beigelegt ist eine Liste mit Tranferlösungen von ca. 100 Spielprogrammen. Ferner bietet PR8-Soft unter dem Namen DISCUS bis heute vier Sammlungen von Tips und Tricks für den Transfer von ca. 285 Spielen an, gegen Entgeld.

Um Ärger und Frust zu vermeiden, sollte sich jeder potentielle Käufer vorher erkundigen, ob es schon eine Transferlösung für sein Lieblingsprogramm gibt.

| | |
|---|---|
| Nutzeffekt: | 4 |
| Bedienerfreundlichkeit: | 3 |
| Dokumentation: | 3 |
| Arbeitsgeschwindigkeit: | 4 |
| Preis/Leistung: | 4 |

# UNILAB V 2.0

**Ein komfortables Editier- und Druckprogramm für Endlos-Etiketten für alle CPCs, angepaßt für DMP 2000**

Durch den modularen Aufbau von Unilab ist es sehr einfach, das Programm für andere Drucker als den DMP 2000 anzupassen. Eine Kurzbeschreibung ist in das Programm eingebaut und kann über das Menü aufgerufen werden.
Es folgt eine Aufbau-Übersicht:

- Zeile 170 – 1810
  Diverse Unterprogramme und Dimensionierungen
- Zeile 210 – 320
  Dimensionierung der Variablen und der Fenster
- Zeile 340 – 440
  Festlegung des deutschen Zeichensatzes
- Zeile 450 – 760
  Festlegung der Anzahl und der Parameter der Etiketten. Beides kann erweitert oder verändert werden
- Zeile 780 – 920
  Generierung des Titelbildes
- Zeile 940 – 1000
  Aufbau der Titelleiste und des Eingabefeldes am unteren Ende des Bildschirms
- Zeile 1020 – 1060
  Einträge in Eingabefeld
- Zeile 1130 – 1310
  Globale Druckparameter anzupassen an den jeweiligen Drucker
- Zeile 1330 – 1480
  Zeilenabhängige Druckparameter anzupassen an den jeweiligen Drucker
- Zeile 1500 – 1640
  Simulation der Breitschrift auf dem Bildschirm
- Zeile 1660 – 1800
  Strings für Schriftartenmenü kann ebenfalls bei Bedarf bei gleichzeitiger Anpassung der Druckparameter geändert werden
- Zeile 1820 – 2480
  Einzeiliger Texteditor
- Zeile 2490 – 3040
  Eingabe der Schriftarten, global und zeilenabhängig
- Zeile 3050 – 3240
  Druckroutine
- Zeile 3250 – 3540
  Kurzbeschreibung des Programms
- Zeile 3550 – 3610
  Programmende und Rückkehr zu Basic
- Zeile 3620 – 3790
  Hauptprogramm

Folgende Druckmöglichkeiten sind im Programm eingebaut:

- Etikettengrößen
  71,5 x 69,6 mm Label für 3“ Disk
  73,5 x 10,3 mm
  88,9 x 35,7 mm Label für 5 1/4“
  88,9 x 48,4 mm
- Globale Druckparameter
  (gelten für das ganze Label)
  - Standard-Druck
  - Standard-Fettdruck
  - NLQ-Druck
  - Kursiv-Druck
  - Kursiv-Fettdruck
- Zeilenabhängige Druckparameter
  - Hochgestellte Schrift
  - Unterstrichen
  - Breitschrift
  - Breitschrift und Unterstrichen
  - Normale Schrift
- Zeilenzentrierung
  bei Bedarf werden alle Zeilen zentriert

Da das Programm menügesteuert ist, erübrigt sich eine weitere Beschreibung. Viel Erfolg beim Druck Ihrer Labels.

(Hans-Werner Fromme)

**für 464-664-6128**

```
10  ' ##################################### [2279]
#####################################
20  ' ###                                   [61]
                                  ###
30  ' ###                            U N I [389]
 L A B                            ###
40  ' ###                                 [61]
                                  ###
50  ' ###        Ein komfortables Druckpro [3792]
gramm f|r Endlos-Etiketten        ###
60  ' ###                                 [61]
                                  ###
70  ' ###          Entwickelt auf SCHNEIDE [3290]
R CPC 6128 und DMP-2000           ###
80  ' ###                                 [61]
                                  ###
90  ' ###                      (c) Dezember [1423]
 1986  by                        ###
100 ' ###                                 [61]
                                  ###
110 ' ###       Hans-Werner Fromme         [1147]
                                  ###
120 ' ###                                 [61]
                                  ###
```

```
130 ' ##################################### [2279]
#####################################
140 '                                     [117]
150     GOTO 3620                         [2568]
          ' Sprung zum Hauptprogramm
160 '                                     [117]
170 ' ********************************** [2556]
**********************************
180 ' ***                    DIVERSE  UNTER [2790]
PROGRAMME                          ***
190 ' ********************************** [2556]
**********************************
200 '                                     [117]
210 '--- Dimensionierungen ---            [2078]
220 '                                     [117]
230 DIM zeilpar1$(1),zeilpar2$(1),hflag(25 [2132]
),zflag(25)
240 DIM ustr$(1),zwr$(1),gpar1$(20),gpar2$ [2299]
(20)
250 DIM label$(1),breit$(1),zeile$(1),glop [3975]
ar$(5),zeipar$(5),size(1)
260 glopar=5:zeipar=5                     [1109]
270 WINDOW #1,57,79,2,2: PAPER #1,0: PEN # [1746]
1,1
280 WINDOW #2,2,79,23,24: PAPER #2,1:PEN # [2620]
2,0
290 WINDOW #3,42,79,5,20: PAPER #3,1:PEN # [1554]
3,0
```

```
300 WINDOW #5,1,80,4,21: PAPER #5,0                    [1549]
310 etikettenanzahl=4                                  [675]
320 RETURN                                             [555]
330 '                                                  [117]
340 '--- DEUTSCHER ZEICHENSATZ ---                     [2459]
350 '                                                  [117]
360 SYMBOL AFTER 90                                    [1214]
370 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66              [2074]
        '-- AE --
380 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C              [2558]
        '-- OE --
390 SYMBOL 93,&66,&0,&66,&66,&66,&66,&3C               [2893]
        '-- UE --
400 SYMBOL 123,&6C,&0,&78,&C,&7C,&CC,&76               [2418]
        '-- ae --
410 SYMBOL 124,&0,&66,&0,&3C,&66,&66,&3C               [1863]
        '-- oe --
420 SYMBOL 125,&0,&66,&0,&66,&66,&66,&3E               [1618]
        '-- ue --
430 SYMBOL 126,&3C,&66,&66,&7C,&66,&66,&7C             [2375]
,&60    '-- sz --
440 RETURN                                             [555]
450 '                                                  [117]
460 '--- ETIKETTENWERTE ---                            [2110]
470 '                                                  [117]
480 z1$="ETIKETTEN - AUSWAHL"                          [1153]
490 Z2$="--------------------"                         [1591]
500         z3$="Nr:     Gr|~e        Zeile           [2813]
n   Spalten"
510 etiwert$(1)=" 1   71,5 x 69,6      16   [2051]
    28"
520 etiwert$(2)=" 2   73,7 x 10,3       2   [2111]
    29"
530 etiwert$(3)=" 3   88,9 x 35,7       8   [1600]
    35"
540 etiwert$(4)=" 4   88,9 X 48,4      11   [1921]
    35"
550 RETURN                                             [555]
560 '                                                  [117]
570 ' Etikett1                                         [1191]
580 zeilen=16:spalten=28:RETURN                        [2074]
590 '                                                  [117]
600 ' Etikett2                                         [1186]
610 zeilen= 2:spalten=29:RETURN                        [1679]
620 '                                                  [117]
630 ' Etikett3                                         [1189]
640 zeilen= 8:spalten=35:RETURN                        [2603]
650 ' Etikett4                                         [1200]
660 '                                                  [117]
670 zeilen=11:spalten=35:RETURN                        [2017]
680 '                                                  [117]
690 ' Zeichnen der Rahmen                              [2266]
700 '                                                  [117]
710 ORIGIN 4,340                                       [295]
720 DRAWR (spalten*8+8),0,(farbe)                      [2574]
730 DRAWR 0,-(zeilen*16+8),(farbe)                     [1360]
740 DRAWR -(spalten*8+8),0,(farbe)                     [1981]
750 DRAWR 0,(zeilen*16+8),(farbe)                      [1884]
760 RETURN                                             [555]
770 '                                                  [117]
780 '--- Titelbild generieren ---                      [2352]
790 '                                                  [117]
800 MODE 2:OUT &BC00,6:OUT &BD00,0                     [2310]
810 ORIGIN 75,125:DRAW 395,0,1:DRAW 395,13             [2746]
0:DRAW 0,130:DRAW 0,0
820 FOR i=0 TO 80 STEP 16                              [790]
830 ORIGIN 75+i,125+i                                  [709]
840 MOVE 395-16,0:DRAW 395,0:DRAW 395,130:             [3703]
DRAW 0,130:DRAWR 0,-16
850 NEXT i                                             [375]
860 LOCATE 30,11:PRINT "U N I -  L A B"                [2340]
870 LOCATE 15,13:PRINT "Ein komfortables E             [5300]
tiketten-Druckprogramm"
880 LOCATE 18,14:PRINT "f|r Schneider CPC6             [3876]
128 und DMP-2000"
890 LOCATE 17,16:PRINT "Copyright 1986 by              [5117]
Hans-Werner Fromme"
900 OUT &BC00,6:OUT &BD00,25                           [901]
910 FOR i= 1 TO 3000:NEXT                              [1003]
920 RETURN                                             [555]
930 '                                                  [117]
940 '--- Titelleiste und Eingabefeld aufba             [3381]
uen ---
950 '                                                  [117]
960 CLS:ORIGIN 0,390:DRAWR 637,0:DRAWR 0,-             [3691]
29:DRAWR -637,0:DRAWR 0,29
970 ORIGIN 2,392:DRAWR 637,0:DRAWR 0,-29:M             [3177]
OVER -265,-2:DRAWR 0,29
980 LOCATE 2,2:PRINT"UNILAB V1.0     ";CHR             [6542]
$(164);" 1986 by Hans-Werner Fromme"
990 LOCATE 49,2:PRINT "Modus: ":RETURN                 [3590]
1000 ORIGIN 0,52:DRAWR 637,0:DRAWR 0,-42:D             [2773]
RAWR -637,0:DRAWR 0,42:RETURN
1010 '                                                 [117]
1020 '--- Window-Eintr{ge                              [2053]
1030 '                                                 [117]
1040 CLS#1:LOCATE #1,58,1:PRINT #1,modus$              [1862]
1050 CLS #2:LOCATE #2,2,1:PRINT #2,zeile23             [2295]
$
1060 LOCATE #2,2,2:PRINT #2,zeile24$:RETUR             [2892]
N
1070 '                                                 [117]
1080 '--- Ja/Nein-Abfrage ---                          [1734]
1090 '                                                 [117]
1100 ein$="":WHILE ein$="" OR INSTR("JN",e             [3478]
in$)=0:ein$=UPPER$(INKEY$):WEND
1110 RETURN                                            [555]
1120 '                                                 [117]
1130 '--- Globale Druckparameter ---                   [2475]
1140 '                                                 [117]
1150 deutsch$=CHR$(27)+"R"+CHR$(2)                     [4239]
          ' Deutscher Zeichensatz
1160 '                                                 [117]
1170 vorschub$=CHR$(27)+"2":RETURN                     [2139]
          ' Vorschub 1/6"
1180 '                                                 [117]
1190 gpar$(1)="":gpar$(2)="":RETURN                    [537]
1200 gpar$(1)=CHR$(27)+"E"                             [3311]
          ' Standard-Fettdruck ein
1210 gpar$(2)=CHR$(27)+"F"                             [3087]
          ' Standard-Fettdruck aus
1220 RETURN                                            [555]
1230 gpar$(1)=CHR$(27)+"x"+CHR$(1)                     [1805]
          ' NLQ ein
1240 gpar$(2)=CHR$(27)+"x"+CHR$(0)                     [1148]
          ' NLQ aus
1250 RETURN                                            [555]
1260 gpar$(1)=CHR$(27)+"4"                             [1709]
          ' Kursiv ein
1270 gpar$(2)=CHR$(27)+"5"                             [914]
          ' Kursiv aus
1280 RETURN                                            [555]
1290 gpar$(1)=CHR$(27)+"4"+CHR$(27)+"E"                [3596]
          ' Kursiv-Fettduck ein
1300 gpar$(2)=CHR$(27)+"F"+CHR$(27)+"5"                [3172]
          ' Kursiv-Fettduck aus
1310 RETURN                                            [555]
1320 '                                                 [117]
1330 --- Zeilenabh{ngige Druckparameter --             [1649]
-
1340 '                                                 [117]
1350 gpar$(1)="":gpar$(2)="":RETURN                    [537]
1360 zpar$(1)="":zpar$(2)="":hflag=0:zwr$(             [5451]
i,j)="":ustr$(i,j)=" ":GOSUB 1520:RETURN
1370 zpar$(1)=CHR$(27)+"S"+CHR$(0)                     [3902]
          ' Hochgestellt ein
1380 zpar$(2)=CHR$(27)+"T"                             [2448]
          ' Hochgestellt aus
1390 hflag=1:zwr$(i,j)="":ustr$(i,j)=" ":G             [3141]
OSUB 1520:RETURN
1400 zpar$(1)=CHR$(27)+"-"+CHR$(1)                     [3292]
          ' Unterstreichen ein
1410 zpar$(2)=CHR$(27)+"-"+CHR$(0)                     [3273]
          ' Unterstreichen aus
1420 hflag=0:zwr$(i,j)="":ustr$(i,j)="_":G             [2327]
OSUB 1520:RETURN
1430 zpar$(1)=CHR$(27)+"W"+CHR$(1)                     [2881]
          ' Breitschrift ein
1440 zpar$(2)=CHR$(27)+"W"+CHR$(0)                     [1465]
          ' Breitschrift aus
1450 hflag=0:zwr$(i,j)=" ":ustr$(i,j)=" ":             [2273]
GOSUB 1520:RETURN
1460 zpar$(1)=CHR$(27)+"W"+CHR$(1)+CHR$(27             [3612]
)+"-"+CHR$(1)   ' BS unterstr. ein
1470 zpar$(2)=CHR$(27)+"-"+CHR$(0)+CHR$(27             [4586]
)+"W"+CHR$(0)   ' BS unterstr. aus
1480 hflag=0:zwr$(i,j)=" ":ustr$(i,j)="_":             [3333]
GOSUB 1520:RETURN
1490 '                                                 [117]
1500 '--- Breitschrift f|r Bildschirm simu             [3917]
lieren ---
1510 '                                                 [117]
1520 IF zwr$(i,j)=" " THEN IF size(i,j)>(s             [15061]
palten\2) THEN zeile23$="Text f|r Breitsch
```

```
rift zu lang !":zeile24$="Bitte anderen Ze
ilenparameter waehlen !":GOSUB 1050:PRINT
CHR$(7):FOR x=1 TO 1000:NEXT:j=j-1:RETURN
1530 LOCATE #4,1,(j):PRINT #4,STRING$((spa [2914]
lten)," ");
1540 LOCATE #4,1,(j)                       [763]
1550 IF zflag(i)=1 THEN PRINT #4,SPACE$((s [6388]
palten-(size(i,j)+size(i,j)*LEN(zwr$(i,j))
))\2);
1560 FOR x=1 TO size(i,j)                  [1153]
1570     PRINT #4,MID$((label$(i,j)),x,1); [1851]
zwr$(i,j);
1580 NEXT x                                [356]
1590 LOCATE #4,1,(j)                       [763]
1600 IF zflag(i)=1 THEN PRINT #4,SPACE$((s [6388]
palten-(size(i,j)+size(i,j)*LEN(zwr$(i,j))
))\2);
1610 PRINT #4,CHR$(22);CHR$(1);            [1541]
1620     PRINT #4,STRING$((size(i,j)+size( [4367]
i,j)*LEN(zwr$(i,j))),(ustr$(i,j)));
1630 PRINT #4,CHR$(22);CHR$(0);            [1545]
1640 RETURN                                [555]
1650 '                                     [117]
1660 '--- Strings f}r Schriftartenmenue -- [2500]
-
1670 '                                     [117]
1680 z11$="Globale Druckparameter":z2$=STR [3301]
ING$(22,"-")
1690 glopar$(1)="Standard"                 [1236]
1700 glopar$(2)="Kursiv"                   [1204]
1710 glopar$(3)="NLQ"                      [619]
1720 glopar$(4)="Standard + Fettdruck"     [3002]
1730 glopar$(5)="Kursiv + Fettdruck":RETUR [3530]
N
1740 z1$="Zeilenabh{ngige Druckparameter": [4755]
z2$=STRING$(30,"-")
1750 zeipar$(1)="Hochgestellt"             [1890]
1760 zeipar$(2)="Unterstrichen"            [2872]
1770 zeipar$(3)="Breitschrift"             [2470]
1780 zeipar$(4)="Breitschrift unterstriche [3702]
n"
1790 zeipar$(5)="Keine zus{tzlichen Parame [3636]
ter"
1800 RETURN                                [555]
1810 '                                     [117]
1820 '------------------------------------ [3447]
----------------------------------
1830 '---                          E D I T [848]
O R                              ---
1840 '------------------------------------ [3447]
----------------------------------
1850 '                                     [117]
1860 modus$=" E D I T O R "                 [1578]
1870 zeile23$=""                           [734]
1880 zeile24$=""                           [529]
1890 GOSUB 1040                            [875]
1900 ORIGIN 322,340                        [393]
1910 DRAWR 315,0:DRAWR 0,-266:DRAWR -315,0 [2333]
:DRAWR 0,266
1920 GOSUB 480:CLS #3                      [1087]
1930 LOCATE #3,9,2: PRINT #3,z1$           [1094]
1940 LOCATE #3,9,3: PRINT #3,z2$           [927]
1950 LOCATE #3,2,6: PRINT #3,z3$           [1605]
1960 FOR x=1 TO etikettenanzahl            [2069]
1970     LOCATE #3,2,7+x:PRINT #3,etiwert$ [1728]
(x)
1980 NEXT x                                [356]
1990 zeile23$="Bitte w{hlen Sie: ":GOSUB 1 [3548]
050:PRINT CHR$(7);
2000 ein$="":WHILE ein$="" OR INSTR("1234" [3402]
,ein$)=0:ein$=INKEY$:WEND
2010 CLS #2                                [372]
2020 ON VAL(ein$) GOSUB 580,610,640,670    [1439]
2030 farbe=1:GOSUB 710                     [841]
2040 CLS#3:LOCATE #3,2,2: PRINT #3,"Sie ha [2909]
ben gew{hlt:"
2050       LOCATE #3,2,3: PRINT #3,"------ [2615]
----------"
2060       LOCATE #3,2,5: PRINT #3,"Etiket [2723]
t Nr: ";ein$
2070       LOCATE #3,2,7: PRINT #3,"Zeilen [3057]
zahl          :  ";zeilen
2080       LOCATE #3,2,9: PRINT #3,"Spalte [4676]
n (Normalschrift): ";spalten
2090       LOCATE #3,2,10: PRINT #3,"Spalt [4767]
en (Breitschrift) : ";spalten\2
2100 zeile23$="Etikettenwahl korrekt ?  (J [5114]
/N)":GOSUB 1050:PRINT CHR$(7);
2110 GOSUB 1090                            [905]
2120 IF ein$="N" THEN farbe=0:GOSUB 710:CL [2468]
S #3:GOTO 1930
2130 WINDOW #4,2,2+(spalten-1),5,5+(zeilen [3997]
-1):PAPER #4,0:PEN #4,1:CLS #4
2140 FOR x=1 TO zeilen                     [1123]
2150     LOCATE spalten+2,x+4: PRINT" "    [1799]
2160     LOCATE spalten+2,x+4: PRINT x     [1786]
2170 NEXT x:ORIGIN 4+(spalten*8+8),340:DRA [3806]
WR 0,-(zeilen*16+8),farbe
2180 zeile23$="Wieviel Etiketten dieser Gr [6330]
|~e wollen Sie editieren ?"
2190 GOSUB 1050:PRINT CHR$(7);             [1798]
2200 LOCATE #2,2,2:INPUT #2,"Anzahl";anzah [2125]
l
2210 IF anzahl <1 THEN 2180                [721]
2220 ERASE label$,breit$,zeile$,size,zeilp [3585]
ar1$,zeilpar2$,zwr$,ustr$
2230 DIM label$(anzahl,zeilen),breit$(anza [6249]
hl,zeilen),zeile$(anzahl,zeilen)
2240 DIM size(anzahl,zeilen),zeilpar1$(anz [5616]
ahl,zeilen),zeilpar2$(anzahl,zeilen)
2250 DIM zwr$(anzahl,zeilen),ustr$(anzahl, [3368]
zeilen)
2260 FOR i=1 TO anzahl                     [1073]
2270     FOR j=1 TO zeilen                 [1689]
2280         CLS #2:LOCATE #2,2,1:PRINT #2 [4037]
,"Etikett: ";i;" Zeile: ";j
2290         LOCATE #2,2,2:PRINT #2,"Einga [4751]
be:  ";STRING$((spalten),".")
2300         LOCATE #2,10,2:INPUT #2,"";la [3237]
bel$(i,j)
2310         size(i,j)=LEN(label$(i,j))    [1601]
2320         IF size(i,j)>spalten THEN PRI [2535]
NT CHR$(7):GOTO 2280
2330         LOCATE #4,1,(j):PRINT #4,labe [2716]
l$(i,j);
2340     NEXT j                            [370]
2350     CLS #2:LOCATE#2,2,1:PRINT #2,"Eti [5607]
kett ";i;" korrekt ?  (J/N)";CHR$(7)
2360     GOSUB 1090                        [905]
2370     IF ein$="J" THEN 2460             [522]
2380     CLS #2:LOCATE #2,2,1:INPUT #2,"We [6361]
lche Zeile ist nicht korrekt";j
2390     CLS #2:LOCATE #2,2,1:PRINT#2,"Eti [4037]
kett: ";i;" Zeile: ";j
2400     LOCATE #2,2,2:PRINT #2,"Eingabe:  [4751]
 ";STRING$((spalten),".")
2410     LOCATE #2,10,2:INPUT #2,"";label$ [3237]
(i,j)
2420     IF LEN(label$(i,j)) >spalten THEN [3109]
 2390
2430     LOCATE #4,1,(j):PRINT #4,STRING$( [2914]
(spalten)," ");
2440     LOCATE #4,1,(j):PRINT #4,label$(i [2716]
,j);
2450     GOTO 2350                         [309]
2460     CLS #4:NEXT i                     [861]
2470 CLS #3:CLS #4:RETURN                  [1147]
2480 '                                     [117]
2490 '------------------------------------ [3447]
----------------------------------
2500 '---                      S C H R I F T [2219]
A R T                            ---
2510 '------------------------------------ [3447]
----------------------------------
2520 '                                     [117]
2530 modus$=" S C H R I F T A R T "         [1907]
2540 zeile23$=""                           [734]
2550 zeile24$=""                           [529]
2560 GOSUB 1040:GOSUB 1670:GOSUB 1740      [1798]
2570 FOR j=1 TO zeilen                     [1689]
2580     LOCATE #4,1,(j):PRINT #4,label$(i [2716]
,j);
2590 NEXT j                                [370]
2600 CLS #3:LOCATE #3,2,2: PRINT #3,z11$   [1947]
2610 LOCATE #3,2,3:PRINT #3,z2$            [2356]
2620 FOR x=1 TO glopar                     [1453]
2630     LOCATE #3,2,4+x:PRINT #3,x;glopar [1546]
$(x)
2640 NEXT x                                [356]
2650 zeile24$="Bitte w{hlen Sie: ":GOSUB 1 [3505]
050:PRINT CHR$(7);
2660 LOCATE #2,2,1:PRINT #2,"Etikett : ";i [2592]
2670 ein$="":WHILE ein$="" OR INSTR("12345 [3640]
",ein$)=0:ein$=INKEY$:WEND
2680 CLS #2                                [372]
```

```
2690 count=VAL(ein$)                                    [822]
2700 ON VAL(ein$) GOSUB 1190,1260,1230,120              [1299]
0,1290
2710 gpar1$(i)=gpar$(1):gpar2$(i)=gpar$(2)              [2595]
2720 LOCATE #3,2,12:PRINT #3,"Sie haben ge              [2836]
w{hlt:"
2725 zeile24$=""                                        [529]
2730 LOCATE #3,2,13:PRINT #3,STRING$(18,"-              [1884]
")
2740 LOCATE #3,2,15:PRINT #3,glopar$(count              [2685]
);CHR$(7);
2750 zeile23$="Parameterwahl korrekt ? (J/              [5763]
N)":GOSUB 1050:GOSUB 1100
2760 IF ein$="N" THEN GOTO 2600                         [479]
2770 CLS #3:LOCATE #3,2,2: PRINT #3,z1$                 [1896]
2780 LOCATE #3,2,3:PRINT #3,z2$                         [2356]
2790 FOR x=1 TO zeipar                                  [1239]
2800     LOCATE #3,2,4+x:PRINT #3,x;zeipar              [1489]
$(x)
2810 NEXT x                                             [356]
2820 GOSUB 1740                                         [863]
2830 FOR j=1 TO zeilen                                  [1689]
2840     CLS #2:LOCATE #2,2,1:PRINT #2,"Et              [4037]
ikett: ";i;" Zeile: ";j
2850     LOCATE #2,2,2:PRINT #2,"Parameter              [3310]
-Nr: ";CHR$(7)
2860     ein$="":WHILE ein$="" OR INSTR("1              [3640]
2345",ein$)=0:ein$=INKEY$:WEND
2870     hflag=0                                        [233]
2880     ON VAL (ein$) GOSUB 1370,1400,143              [1641]
0,1460,1360
2890     zeilpar1$(i,j)=zpar$(1):zeilpar2$              [2731]
(i,j)=zpar$(2):hflag(j)=hflag
2900     NEXT j                                         [370]
2910 CLS #3:LOCATE #3,2,1:PRINT #3,"Zus{tz              [3584]
lich zum nebenstehenden "
2920       LOCATE #3,2,2:PRINT #3,"Schrif               [3628]
tbild werden folgende Zeilen"
2930       LOCATE #3,2,3:PRINT #3,"<HOCHG               [3420]
ESTELLT> gedruckt:"
2940 y=0:FOR x=1 TO zeilen                              [1603]
2950        IF (hflag(x)=1) THEN LOCATE                 [2360]
#3,5,4+y:PRINT #3,x: y=y+1
2960     NEXT x                                         [356]
2970 ZEILE24$=""                                        [529]
2980 zeile23$="Parameterwahl korrekt ? (J/              [6403]
N)":GOSUB 1050:PRINT CHR$(7)
2990 GOSUB 1100                                         [863]
3000 IF ein$="N" THEN 2770                              [383]
3010 zeile23$="Sollen die Zeilen zentriert              [5976]
 werden ? (J/N)":GOSUB 1050
3020 PRINT CHR$(7):GOSUB 1090                           [2143]
3030 IF ein$="N" THEN zflag(i)=0:CLS #4:RE              [1903]
TURN
3040 zflag(i)=1:FOR j=1 TO zeilen :GOSUB 1              [4737]
520:NEXT j:FOR x=1 TO 2000:NEXT:CLS #4:RET
URN
3050 '------------------------------------             [3447]
---------------------------------
3060 '---                          D R U C              [817]
K                                  ---
3070 '------------------------------------             [3447]
---------------------------------
3080 '                                                  [117]
3090 modus$=" D R U C K "                               [1278]
3100 zeile23$="Bitte das richtige Etikette              [8534]
nformat in den Drucker einlegen."
3110 zeile24$="Wenn Drucker bereit, bitte               [4880]
mit <SPACE> quittieren "
3120 GOSUB 1040                                         [875]
3130 ein$="":WHILE ein$<>CHR$(32):ein$=INK              [2041]
EY$:WEND
3140 GOSUB 1150                                         [821]
3150 FOR i=1 TO anzahl                                  [1073]
3160 PRINT #8,deutsch$;vorschub$;gpar1$(i)              [1623]
;
3170 FOR j=1 TO zeilen                                  [1689]
3180 PRINT #8,SPACE$(3);                                [1564]
3190 IF zflag(i)=1 THEN PRINT #8,SPACE$((s              [7310]
palten-(size(i,j)+size(i,j)*LEN(zwr$(i,j))
))\2);
3200     PRINT #8,zeilpar1$(i,j);label$(i,              [3116]
j);zeilpar2$(i,j):NEXT j
3210 PRINT#8, gpar2$(i)                                 [1369]
3220 NEXT i                                             [375]
3240 RETURN                                             [555]
3250 '------------------------------------             [3447]
---------------------------------
3260 '---                  K U R Z B E S C H            [1089]
R E I B U N G                  ---
3270 '------------------------------------             [3447]
---------------------------------
3280 '                                                  [117]
3290 modus$="BESCHREIBUNG"                              [1347]
3300 zeile23$=""                                        [734]
3310 zeile24$=""                                        [529]
3320 GOSUB 1040                                         [875]
3325 RESTORE 3410                                       [733]
3330 FOR i=1 TO 14                                      [311]
3340 READ z$                                            [331]
3350 LOCATE #5,1,1+i                                    [340]
3360 PRINT #5,z$                                        [536]
3370 NEXT i                                             [375]
3380 zeile23$="Weiter mit beliebiger Taste              [2571]
":GOSUB 1050
3390 ein$="":WHILE ein$="":ein$=INKEY$:WEN              [2253]
D
3400 RETURN                                             [555]
3410 DATA"                          K U R Z B           [2340]
E S C H R E I B U N G"
3420 DATA"UNILAB V2.0 ist ein menuegesteue              [7877]
rtes Editier- und Druckprogramm f}r Endlos
eti-"
3430 DATA"ketten. Vier Etikettenformate si              [7605]
nd bereits im Programm gespeichert. Durch
den "
3440 DATA"modularen Aufbau k}nnen die Form              [6875]
ate einfach ge{andert oder ausgetauscht we
rden."
3450 DATA"Das gleiche gilt f}r die Drucker              [4251]
anpassung."
3460 DATA""                                             [208]
3470 DATA"Das Programm l{uft wie folgt ab:              [3167]
3480 DATA"Nach Wahl der taste <E> befindet              [7820]
 sich der Computer im Editiermodus. Sie ge
ben "
3490 DATA"den gew}nschten Text der Etikett              [6078]
en zeilenweise ein. Ist alles korrekt, geh
t"
3500 DATA"es weiter mit der Auswahl der Sc              [7836]
hriftarten, global und zeilenweise, f}r je
des"
3510 DATA "Etikett. Nach dem Einlegen der               [4709]
Etiketten in den Drucker ( Pfeil der linke
n"
3520 DATA"Stachelwalze auf 9'' ), werden a              [5453]
lle Etiketten gedruckt."
3530 DATA""                                             [208]
3540 DATA"Viel Spa~ mit   UNILAB V2.0"                  [2810]
3550 '------------------------------------             [3447]
---------------------------------
3560 '---                  P R O G R A M M              [805]
E N D E                        ---
3570 '------------------------------------             [3447]
---------------------------------
3580 '                                                  [117]
3590 CLS                                                [91]
3600 RETURN                                             [555]
3610 '                                                  [117]
3620 '------------------------------------             [3553]
---------------------------------
3630 '---                   H A U P T P R O            [1997]
 G R A M M                     ---
3640 '------------------------------------             [3553]
---------------------------------
3660 '                                                  [117]
3670 GOSUB 360:GOSUB 780                                [1677]

3680 GOSUB 230:GOSUB 960                                [1323]
3690 GOSUB 1000                                         [851]
3700 CLS #5                                             [385]
3710 modus$=" H A U P T M E N U E "                     [1975]
3720 zeile23$="  <E> = Editor              <K           [5447]
> = Kurzbeschreibung          <Q> = Quit"
3725 zeile24$=""                                        [529]
3730 GOSUB 1040                                         [875]
3740 ein$="":WHILE ein$="" OR INSTR("EKQ",              [4682]
ein$)=0:ein$=UPPER$(INKEY$):WEND
3750 IF UPPER$(ein$)="E" THEN GOSUB 1820:F              [5866]
OR i=1 TO anzahl:zflag(i)=0:GOSUB 2490:NEX
T i:GOSUB 3050:GOTO 3700
3760 IF UPPER$(ein$)="K" THEN GOSUB 3250:               [2934]
GOTO 3700
3770 IF UPPER$(ein$)="Q" THEN GOSUB 3550                [1753]
3780 END                                                [110]
3790 SAVE"unilab2                                       [468]
```

# Erzeugung eines READY Signals für 5 1/4"-Floppy-Laufwerke am Schneider-CPC.

**Betreibt man ein oder zwei 5 1/4"-Laufwerke am CPC und legt das READY-Signal einfach auf Masse, so stellt man beim Betreiben des Computers unter CP/M fest, daß das System nicht mehr in der Lage ist einen Diskettenwechsel festzustellen. Nimmt man eine Diskette nach der Aufforderung »Remove disc, preß any key« heraus und drückt eine Taste, so erscheint diese Aufforderung immer wieder.**

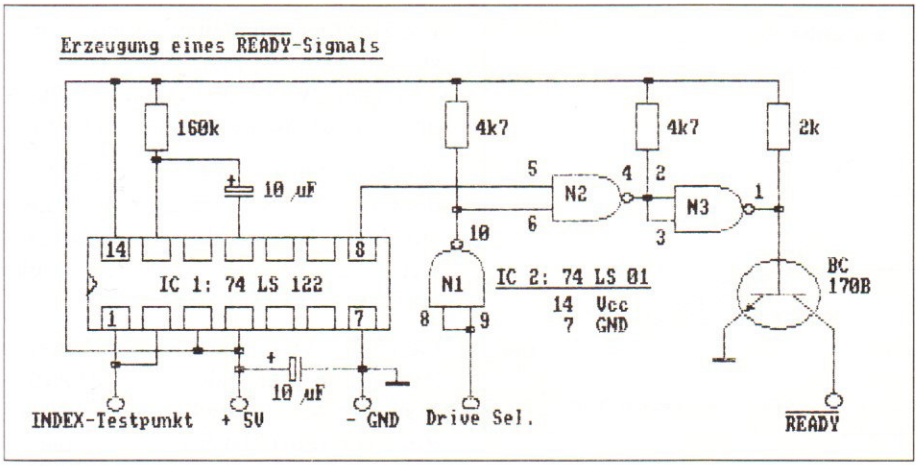
Die Ursache ist, daß für das READY-Signal folgende Bedingungen abgefragt werden:

1. dreht der Motor des Disketten-Laufwerks,
2. ist eine Diskette eingelegt,
3. ist das Laufwerk gewählt.

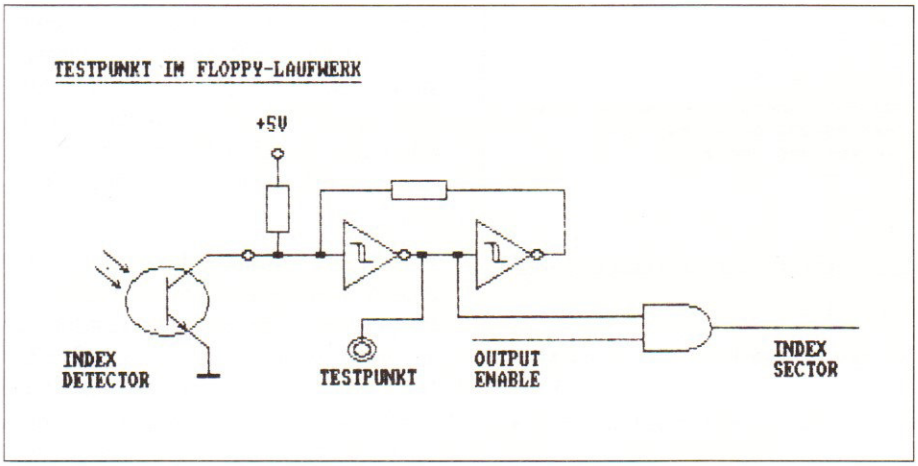
Legt man das READY-Signal auf Masse, sind ständig alle drei Bedingungen erfüllt. Abhilfe schafft Schaltung Nr.: 1.

**Belegung 74 LS 122:**

1 A1
2 A2
3 B1
4 B2
5 CLR
6 not Q
7 GND
8 Q
9 R int
10 NC
11 C ext
12 NC
13 R ext / C ext
14 Vcc




*Schaltung 1*




*Schaltung 2*

**Stückliste:**

1 IC 74 LS 122
1 IC 74 LS 01
1 Transistor BC170B oder BC108B
1 Widerstand 160 kOhm 1/8 W
1 Widerstand 2 kOhm 1/8 W
2 Widerstände 4,7 kOhm 1/8 W
2 Elkos 10 uF 10 V

Auf der Platine des Laufwerks wird der Testpunkt für die Index-Lichtschranke abgegriffen. Wenn eine Diskette eingelegt ist und der Motor dreht, wird am IC 1 der Ausgang Q (8) HIGH, da das 74LS122 immer wieder nachgetriggert wird. Dadurch sind die Bedingungen 1 und 2 erfüllt. Der 160 kOhm Widerstand mit dem 10 uF Kondensator bestimmen die Zeitkonstante. Hinter den DIP-Schaltern auf der Laufwerks-Platine wird das Drive Select Signal abgegriffen und über das NAND-Gate 1 invertiert. Sind nun alle drei Bedingungen erfüllt, so sind die Eingänge von Gate 2 logisch 1. Dadurch ist der Ausgang logisch 0 und der Ausgang von Gate 3 wird logisch 1.

Der Ausgangstransistor leitet und schaltet das READY-Signal gegen Masse. Entfernt man nun die Diskette aus dem Laufwerk, sind die Bedingungen 1 und 2 nicht mehr erfüllt, der Ausgang Q von IC 1 bleibt ständig LOW und der Transistor bleibt gesperrt.

Die Schaltung läßt sich leicht auf eine Experimentierplatine aufbauen.

Testpunkte für die Index-Lichtschranke:

Siemens FDD 100-5 TP5
Micropolis M/N 1115-VI TP8, Pos.2
Tandy TRS 80 Mini Disc TP7

(Siehe Schaltung Nr.: 2)

Zur Funktion des B-Laufwerkes ist noch folgendes anzumerken: Wenn nach dem Einschalten oder RESET des Computers nur eine Diskette in das B-Laufwerk eingelegt und auf B umgeschaltet wird, passiert es manchmal, daß folgende Fehlermeldung erscheint: »Disc missing , Retry, Ignore or Cancel«. In diesem Falle ist »R« für Retry einzugeben und der Computer läuft normal.

Man sollte nur Disketten doppelter Dichte und guter Qualität verwendenden. Bei weißer Ware oder einfacher Dichte habe ich sehr oft Schreib- und Lesefehler gehabt, die in Track 0 oder 37 bis 39 auftraten. Diese sehr billigen Disketten laufen auf dem Apple oder C64, da diese Computer normalerweise nur von Track 1 bis 35 bzw. 36 formatieren.

(Klaus Munsteiner)

# DCF-77 Zeitsignal

**Die physikalisch-technische Bundesanstalt in Braunschweig gibt über den Sender Mainflingen (bei Frankfurt) das amtliche Zeitzeichen in der Bundesrepublik. Diese codierten Zeichen können über geeignete Empfänger dekodiert werden, wobei eine sehr genaue Zeitanzeige zur Verfügung steht.**

```
Einlaufende Daten der Version ACS-77 V3.1  (1.3.1987)
-------------------------------------------------------->
01. 02. 03. 04. 05. 06. 07. 08. 09. 10. 11. 12. 13.  Stelle

 9   0   7   1   6   1   1   0   3   0   7   8   1   Beispiel von mög-
 :   :   :   :   :   :   :   :   :   :   :   :   :   lichen Daten
 :   :   :   :   :   :   :   :   :   :   :   :   :
 :   :   :   :   :   :   :   :   :   :   :   :   :.. Tag      1 =Sonntag
 :   :   :   :   :   :   :   :   :   :   :...:..... Jahr     78=1987
 :   :   :   :   :   :   :   :   :...:............. Monat    30=März
 :   :   :   :   :   :   :...:...:.................. Datum    10=01.
 :   :   :   :   :...:...:.......................... Stunde   61=16 Uhr
 :   :   :...:...:.................................. Minute   71=17 Min.
 :...:...:.......................................... Sekunde  90=09 Sek.

Sobald die Jahreszahl = 87 erkannt wird (im Programm = 78 da die  Ei -
ner als erstes einlaufen) werden alle nachfolgenden Daten in die Zeit-
variablen eingelesen und zwar wie im gezeigten Beispiel ersichtlich:
1.Sekunden  Einer  2.Sekunden  Zehner 3.Minuten Einer 4.Minuten Zehner
5.Stunden  Einer 6.Stunden Zehner 7.Datum Einer 8.Datum Zehner 9.Monat
Einer 10.Monat Zehner 11.Jahr Einer 12.Jahr Zehner 13.Wochentag.

13.Stelle Wochentage:

   1 = Montag    2 = Dienstag    3 = Mittwoch    4 = Donnerstag
   5 = Freitag   6 = Samstag     7 = Sonntag
```

*Abb.1*

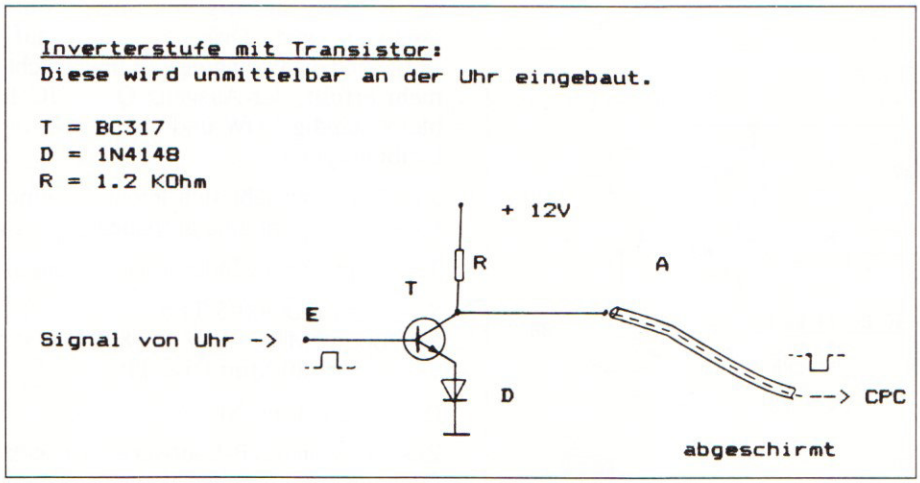



*Abb. 2*

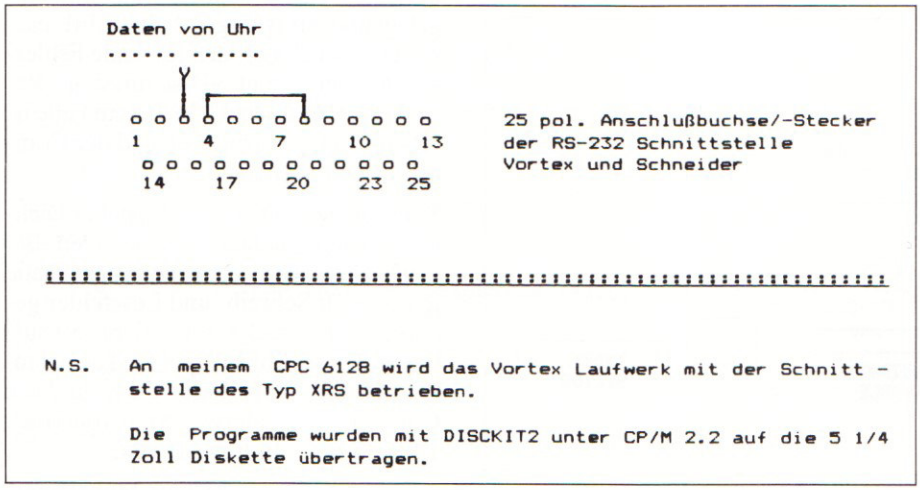



*Abb. 3*

Wer sich über die technischen Einzelheiten genauer informieren will, besorgt sich entsprechende Publikationen. So wurde mir vor einigen Jahren der Auszug aus PTB-Bericht PTB-Me-23 (März 1979) von der Phy.-Techn. Bundesanstalt zur Verfügung gestellt, der für technisch Interessierte ein Leckerbissen ist. Im übrigen wird auf die Bestimmung der Deutschen Bundespost für den Empfang von DCF77, der gebührenfrei ist, hingewiesen.

So bietet eine namhafte Electronik-Versandfirma schon mehrere Jahre unter der Bezeichnung »Renkforce ACS-77« Atomic-Clock-System eine Uhr an, die als Einbaumodul (Einbau in eigenes Gehäuse) verhältnismäßig preiswert ist. Als Alternative zu einer Hardware-Uhr ist dieser Baustein nicht zu verachten.

Die Uhr dient an meiner Amateurfunkstation seit Jahren zur Auswertung der Zeitsignale wie Steuerung von Tochteruhren nebst Auswertung der UT-Time.

An die eingebaute V24 Schnittstelle kann der CPC über einen Inverter mit Transistor BC 317 angeschlossen werden, soweit dieser über eine serielle Schnittstelle verfügt.

Geliefert wird eine Standardversion ACS-77 V3.1, die leider bei der Übertragung der ASCII-Zeichen am Ende der Zeitdaten kein CR/LF anhängt, welches in Verbindung mit Computern vorteilhaft wäre. Sonderversionen sind zwar erhältlich, aber es wird nicht darauf hingewiesen, so daß in der Regel die Standardversion ausgeliefert wurde. »Sollte die Ausführung mit CR/LF vorhanden sein, benutzt man dies natürlich zur Erkennung der Zeiteinlesung«, anstatt der Jahreszahl.

Das Programmbeispiel 1 und 2 ist für eine »Vortex RS-232C Schnittstelle«. Im Programm 1 werden die Sekunden mit angezeigt, wobei sich das Programm 2 in Ihre eigenen Programme einbinden läßt. Funkamateure nutzen zur Auswertung die UT-Time, um die Zeit in die Logauswertung einzubeziehen.

Damit sich die Besitzer der Schneider RS-232 Schnittstelle helfen können, wurde im »Programmbeispiel 3« die Schnittstelleneinstellung dargestellt.

Einlaufende Daten der Version ACS-77 V3.1 (1.3.1987, s. Abb. 1)

Sobald die Jahreszahl = 87 erkannt wird, (im Programm = 78, da die Einer als erstes einlaufen), werden alle nachfolgenden Daten in die Zeitvariablen eingelesen und zwar wie im gezeigten Beispiel ersichtlich:

1. Sekunden Einer
2. Sekunden Zehner
3. Minuten Einer
4. Minuten Zehner
5. Stunden Einer
6. Stunden Zehner
7. Datum Einer
8. Datum Zehner
9. Monat Einer
10. Monat Zehner
11. Jahr Einer
12. Jahr Zehner
13. Wochentag.

13. Stelle Wochentage:

1 = Montag
2 = Dienstag
3 = Mittwoch
4 = Donnerstag
5 = Freitag
6 = Samstag
7 = Sonntag

Inverterstufe mit Transistor:
Diese wird unmittelbar an der Uhr eingebaut.

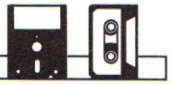
T = BC317
D = 1N4148
R = 1.2 KOhm

(s. Abb. 2)
Daten von Uhr
(s. Abb. 3)

3 = DatenIn
4 – 8 Brücke

(Hans J. Ziegler/SH)

## für 464-664-6128

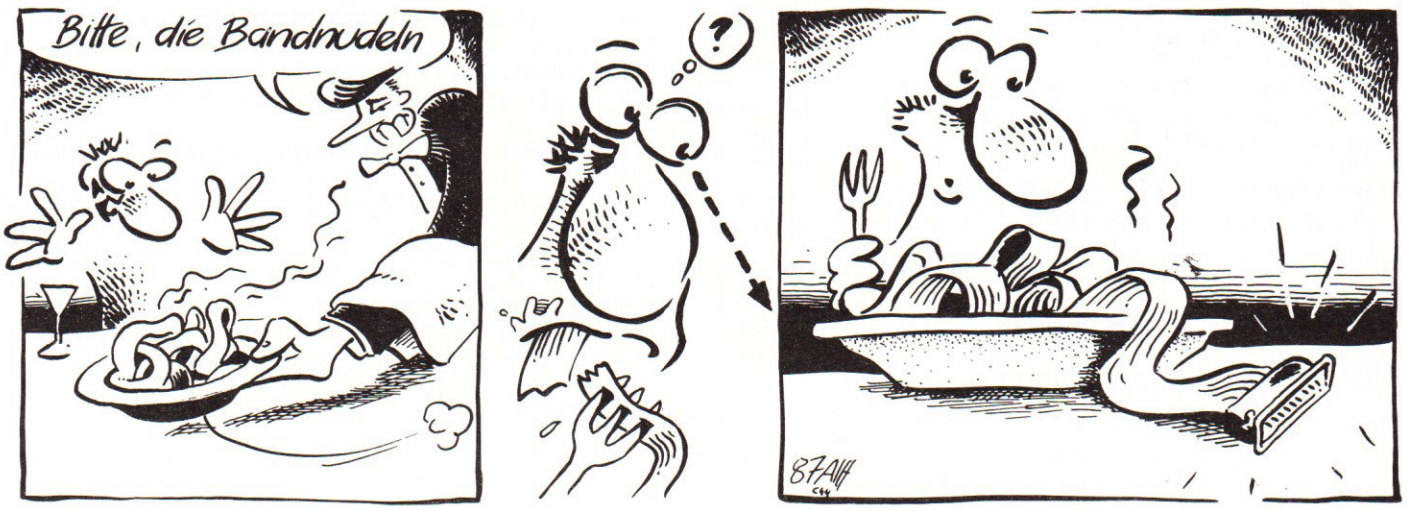


```
10 '************************************ [1668]
*
20 '****         (c) HJZ DK9PM         *** [946]
*
30 '************************************ [1668]
*
40 MODE 2:DIM z$(13)                     [1634]
50 'Jahr einlesen                        [611]
60 OPENIN "dcf77.dat":INPUT#9,jahr$:CLOSEI [2078]
N
70 IF jahr$="" THEN n=21 :GOTO 180       [2698]
80 WINDOW 1,80,2,25:WINDOW#7,1,80,1,1    [2987]
90 'Serielle Schnittstelle Vortex        [2188]
100 |SETSIO,300,300,0,8,0,1              [2105]
110 |INITSIO                             [852]
120 |SETTIMEOUT,1000                     [2056]
130 DEFINT s,z:s=0:z=0                   [1227]
140 |INCHAR,@s,@z                        [1335]
150 z$=CHR$(z)                           [600]
160 zeit$=zeit$+z$                       [980]
170 IF INSTR(zeit$,jahr$) THEN 200       [817]
180 IF n>20 THEN n=0:CLS:PRINT:INPUT"Jahr  [7433]
Zehner";jahr1$:INPUT"Jahr Einer ";jahr2$:j
ahr$=jahr2$+jahr1$:OPENOUT"dcf77.dat":PRIN
T#9,jahr$:CLOSEOUT:GOTO 80
190 n=n+1:GOTO 130                       [365]
200 n=0:n1=0                             [231]
210 DEFINT s,z:s=0:z=0                   [1227]
220 |INCHAR,@s,@z                        [1335]
230 z$(n1)=CHR$(z)                       [1486]
240 IF datum=1 AND n1=6 THEN 390         [566]
250 IF n1>12 THEN 270                    [1162]
260 n1=n1+1:GOTO 210                     [538]
270 IF datum=1 THEN 390                  [533]
280 PRINT#7,CHR$(24)STRING$(80,32)" ";   [1941]
290 IF z$(0)="1" THEN PRINT#7,"Montag    " [2596]
;
300 IF z$(0)="2" THEN PRINT#7,"Dienstag  " [2698]
;
310 IF z$(0)="3" THEN PRINT#7,"Mittwoch  " [2662]
;
320 IF z$(0)="4" THEN PRINT#7,"Donnerstag" [3319]
;
330 IF z$(0)="5" THEN PRINT#7,"Freitag   " [2639]
;
340 IF z$(0)="6" THEN PRINT#7,"Samstag   " [2806]
;
350 IF z$(0)="7" THEN PRINT#7,"Sonntag   " [3726]
;
360 PRINT#7,",den ";                     [1188]
370 LOCATE#7,17,1:PRINT#7,z$(8)+z$(7)+"."+ [3426]
z$(10)+z$(9)+"."+z$(12)+z$(11)
380 LOCATE#7,40,1:PRINT#7,"  (c)  HJZ  "CH [2786]
R$(24)
390 LOCATE#7,68,1:PRINT#7,CHR$(24)"[ "z$(6 [5114]
)+z$(5)+":"+z$(4)+z$(3)+":"+z$(2)+z$(1)" ]
"CHR$(24)
400 zeit$=" "                            [650]
410 datum=1                              [581]
420 GOTO 130                             [371]
```

```
10 '************************************ [1668]
*
20 '****         (c) HJZ DK9PM         *** [946]
*
30 '************************************ [1668]
*
40 MODE 2:DIM z$(13)                     [1634]
50 'Jahr einlesen                        [611]
60 OPENIN "dcf77.dat":INPUT#9,jahr1$:INPUT [4912]
#9,jahr2$:INPUT#9,som$:jahr$=jahr2$+jahr1$
:CLOSEIN
70 CLS:PRINT"Jahr ";jahr1$+jahr2$;:IF som$ [6155]
="j" OR som$="J" THEN PRINT"    MESZ ":IF
som$="n" OR som$="N" THEN PRINT"MEZ "
80 PRINT"Aenderung vornehmen (j/n)"      [2925]
90 a$=INKEY$:IF a$="" THEN 90            [1090]
100 IF a$="j" OR a$="J" THEN n=40:GOTO 500 [1629]
50
110 WINDOW 1,80,2,25:WINDOW#7,1,80,1,1   [2987]
120 'Serielle Schnittstelle Vortex       [2188]
130 |SETSIO,300,300,0,8,0,1              [2105]
140 |INITSIO                             [852]
150 |SETTIMEOUT,1000                     [2056]
160 GOSUB 50000                          [950]
170 EVERY 1000 GOSUB 50000               [1462]
180 LOCATE 1,2:PRINT"Zwischen Zeile 180 bi [6435]
s 40999 koennen Sie Ihr Programm einbinden
 !!!!"
190 GOTO 180                             [401]
```

```
50000 DEFINT s,z:s=0:z=0                                    [1227]
50010 ¦INCHAR,@s,@z                                         [1335]
50020 z$=CHR$(z)                                            [600]
50030 zeit$=zeit$+z$                                        [980]
50040 IF INSTR(zeit$,jahr$) THEN 50070                      [1159]
50050 IF n>38 THEN n=0:CLS:PRINT:INPUT"Jah                  [13844]
r Zehner";jahr1$:INPUT"Jahr Einer ";jahr2$
:INPUT"Sommerzeit (j/n)";som$:OPENOUT"dcf7
7.dat":PRINT#9,jahr1$:PRINT#9,jahr2$:PRINT
#9,som$:CLOSEOUT:ERASE z$:GOTO 10
50060 n=n+1:GOTO 50000                                      [668]
50070 n=0:n1=0                                              [231]
50080 DEFINT s,z:s=0:z=0                                    [1227]
50090 ¦INCHAR,@s,@z                                         [1335]
50100 z$(n1)=CHR$(z)                                        [1486]
50110 IF datum=1 AND n1=6 THEN 50310                        [640]
50120 IF n1>12 THEN 50140                                   [1066]
50130 n1=n1+1:GOTO 50080                                    [736]
50140 IF datum=1 THEN 50310                                 [679]
50150 IF som$="j" OR som$="J" THEN som=1                    [2873]
50160 IF som$="" OR som$="n" OR som$="N" T                  [2160]
HEN som=0
50170 CLS                                                   [91]
50180 PRINT#7,CHR$(24)STRING$(80,32)" ";                    [1941]
50190 IF z$(0)="1" THEN PRINT#7,"Montag                     [2596]
 ";
50200 IF z$(0)="2" THEN PRINT#7,"Dienstag                   [2698]
 ";
50210 IF z$(0)="3" THEN PRINT#7,"Mittwoch                   [2662]
 ";
50220 IF z$(0)="4" THEN PRINT#7,"Donnersta                  [3319]
g";
50230 IF z$(0)="5" THEN PRINT#7,"Freitag                    [2639]
 ";
50240 IF z$(0)="6" THEN PRINT#7,"Samstag                    [2806]
 ";
50250 IF z$(0)="7" THEN PRINT#7,"Sonntag                    [3726]
 ";
50260 PRINT#7,",den ";                                      [1188]
50270 LOCATE#7,17,1:PRINT#7,z$(8)+z$(7)+".                  [3426]
"+z$(10)+z$(9)+"."+z$(12)+z$(11)
50280 LOCATE#7,46,1:PRINT#7,"(c) HJZ dk9pm                  [3728]
 "CHR$(24)
50290 IF som=1 THEN LOCATE#7,30,1:PRINT#7,                  [3371]
CHR$(24)"MESZ"CHR$(24)
50300 IF som=0 THEN LOCATE#7,30,1:PRINT#7,                  [3172]
CHR$(24)"MEZ"CHR$(24)
50310 LOCATE#7,35,1:PRINT#7,CHR$(24)"[ "z$                  [3671]
(6)+z$(5)+":"+z$(4)+z$(3)" ]"CHR$(24)
50320 z1$=z$(6)+z$(5):z1=VAL(z1$)                           [2032]
50330 IF z1>0 AND som=0 THEN z1=z1-1:GOTO                   [1102]
50380
50340 IF z1>1 AND som=1 THEN z1=z1-2:GOTO                   [3502]
50380
50350 IF z1=0 AND som=0 THEN z1=23:GOTO 50                  [2844]
380
50360 IF z1=0 AND som=1 THEN z1=22:GOTO 50                  [2615]
380
50370 IF z1=1 AND som=1 THEN z1=23                          [2184]
50380 z2$=STR$(z1)                                          [1034]
50390 LOCATE#7,68,1:PRINT#7,CHR$(24)"UT ["                  [4492]
z2$+":"+z$(4)+z$(3)" ]"CHR$(24)
50400 zeit$=" "                                             [650]
50410 datum=1                                               [581]
50420 RETURN                                                [555]
```

```
10 MODE 2:DIM z$(13)                                        [1634]
20 'Jahr einlesen                                           [611]
30 OPENIN "dcf77.dat":INPUT#9,jahr$:CLOSEI                  [2078]
N
40 WINDOW 1,80,2,25:WINDOW#7,1,80,1,1                       [2987]
41 PRINT                                                    [361]
42 PRINT"Beachten Sie,dass die erste einla                  [5890]
ufende Zahl die ›› 13. Stelle ‹‹ ist!!!"
50 'Serielle Schnittstelle RS232 Schneider                  [2718]
60 OUT &F8E8,6                                              [296]
70 OUT &F8E0,3                                              [204]
80 OUT &F8E1,0                                              [187]
90 OUT &F8E8,1                                              [301]
100 OUT &F8E0,26                                            [301]
110 OUT &F8E8,2                                             [300]
120 OUT &F8E0,26                                            [301]
130 OUT &F8E8,7                                             [295]
140 OUT &F8ED,51                                            [386]
150 OUT &F8EC,136                                           [211]
160 OUT &F8ED,1                                             [413]
170 OUT &F8EE,1                                             [202]
180 CALL &BB81                                              [318]
190 t=INP(&F8EF)                                            [522]
200 rs=INP(&F8ED)                                           [1187]
210 IF rs<128 THEN GOTO 200                                 [709]
220 z=INP(&F8EF)                                            [560]
230 z$=CHR$(z)                                              [600]
240 zeit$=zeit$+z$                                          [980]
250 zeit$=RIGHT$(zeit$,11)                                  [1468]
260 PRINT#7,z$;                                             [546]
270 IF INSTR(zeit$,jahr$) THEN PRINT#7," "                  [2714]
;:zeit$="":GOTO 200
280 GOTO 200                                                [429]
```



# Zwei 5¼"-Laufwerke am Schneider CPC 6128

**Um den CPC 6128 unter CP/M zu betreiben, sind zwei 5 1/4"-Laufwerke sinnvoll, da man nur noch mit einem Diskettenformat arbeitet und die Disketten preiswerter sind.**

Die Umrüstung ist recht einfach. Zuerst wird der Computer geöffnet, die Stecker werden entfernt und die Platine gelöst.

Neben dem 3"-Laufwerk wird ein 6,5 mm großes Loch in die Abdeckung gebohrt (siehe Skizze). Vom Verbindungskabel zwischen Platine und Laufwerk wird die vierte Ader aufgetrennt (die erste Ader ist rot und jede fünfte grün gekennzeichnet).

Ein zusätzlicher flexibler Draht wird unter die Platine an die Lötstelle von Klemme 24 der Anschlußbuchse für das zweite Laufwerk gelötet und zum Schalter geführt. Die Klemme 24 ist im CPC 6128 unbelegt. Die durchgetrennte vierte Ader wird mit zwei flexiblen Drähten verlängert und ebenfalls zum Schalter geführt.

Die Verschaltung erfolgt gem. Schaltplan. Es ist empfehlenswert, die Leitungen über eine kleine dreipolige Steckverbindung zu führen, damit man beim Öffnen des Gehäuses die Leitungen trennen kann.

Nachdem alles überprüft worden ist, kann alles wieder zusammengebaut werden (siehe Abb. 1 und 2). Wird der Computer eingeschaltet, so verhält er sich wie ein ganz »normaler« CPC, wenn der Schalter auf das 3"-Laufwerk zeigt. Schaltet man um, wird das Signal »Drive Select 0« auf die Anschlußbuchse des zweiten Lauffwerks gegeben. Von dort geht das Signal zum 5 1/4"-A-Laufwerk, das sich im gleichen Gehäuse mit dem B-Laufwerk befindet.

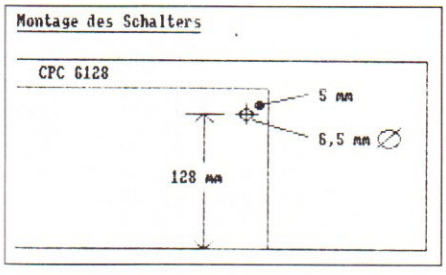



*Abb. 2*

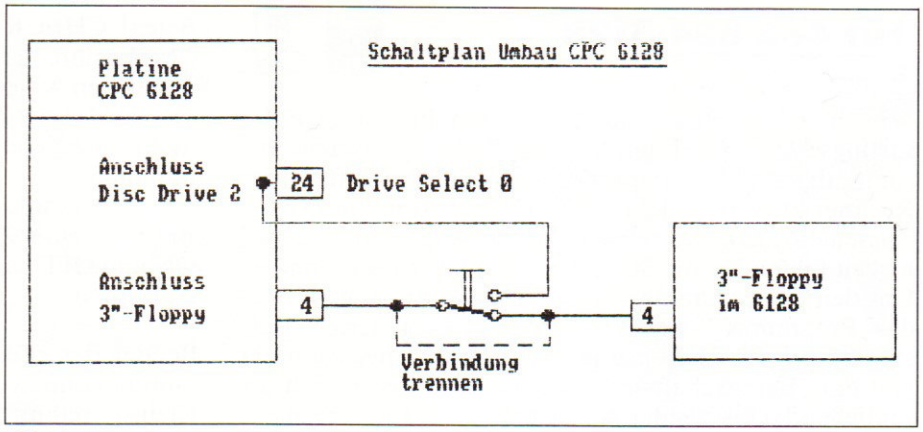



*Abb. 1*

An das Flachbandkabel muß ein zusätzlicher 34-pol. Stecker angeschlossen werden, der das neue Laufwerk versorgt. Jedes Laufwerk erhält eine Platine, die das READY-Signal erzeugt.

Die DIP-Schalter an den Laufwerken müssen auf DS1 für das A-Laufwerk und DS2 für das B-Laufwerk stehen.

Abschlußwiderstände sollten in die Laufwerke nicht eingebaut werden, da sie sich im Computer befinden.

Man sollte 40-Spur-Laufwerke verwenden (Disketten DD 48 tpi).

Falls nur 80-Spur-Laufwerke zur Verfügung stehen, sind diese möglichst im 80-Spur-Format zu betreiben (Disketten DD 96 tpi).

Bei einer Hardwareanpassung von 80 auf 40 Spuren ist das B-Laufwerk nur eingeschränkt und das A-Laufwerk überhaupt nicht funktionsfähig. Im 80-Spur-Format kann auch mit entsprechender Software wie beim 3"-Laufwerk auf 43 Spuren formatiert werden.

Bei Doppelkopflaufwerken kann mit einem zusätzlichen Schalter das HEAD SELECT Signal auf Masse gelegt und auf den oberen Kopf umgeschaltet werden. Wenn dies unter CP/M geschieht, ist wie bei Diskettenwechsel zu verfahren.

Es muß auf ein ausreichend dimensioniertes Netzteil geachtet werden. Zwei getrennte Festspannungsregler L78S5 bzw. L78S12 sind empfehlenswert.

CPC 664:
Die Klemme 24 am CPC 6128 entspricht der 25 am CPC 664. Es muß überprüft werden, daß diese beim 664 nicht auf Masse liegt, ggf. trennen.

(Klaus Munsteiner)

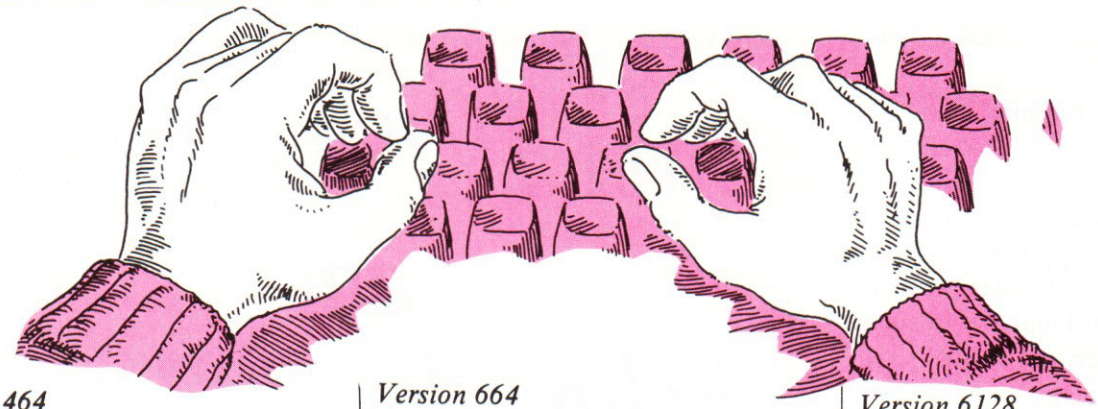
# CPC-Listingservice

## für 464-664-6128

Wie in jedem Heft, finden Sie auch hier wieder den Listingservice, der Ihnen helfen soll, die in unserem Magazin abgedruckte "Paperware" in "Software" für Ihren Rechner zu verwandeln. Neben den Piktogrammen für Kassetten-/Diskettenbetrieb und der Angabe der Lauffähigkeit finden Sie die Super-Eintipphilfe "Checksummer". Um den Checksummer benutzen zu können, genügt es, das Programm einmal aufzurufen; es generiert selbständig den Maschinencode, aktiviert den Checksummer, gibt eine Bereitschaftsmeldung aus und löscht sich anschließend selbst wieder. Ab jetzt wird jede Eingabe, nachdem man ENTER gedrückt hat, mit der Ausgabe einer Quersumme quittiert. Beim Eintippen von Programmlistings kann man nun durch den Vergleich der beiden in eckigen Klammern stehenden Zahlen feststellen, ob man sich vertippt hat und ggf. den Fehler sofort berichtigen.
Des weiteren hat man die Möglichkeit, mit dem RSX-Befehl CHECK,s,z ein Listing mit den dazugehörigen Checksummen zu erzeugen. Die Parameter s und z definieren den Ausgabekanal (s = 0 bis 9) und die Zeilennummer, ab der mit der Ausgabe begonnen werden soll; wird z nicht angegeben, so wird an dessen Stelle 1 angenommen.
Vor dem Druck sollte man mit WIDTH x die maximale Breite (x) einer Zeile festlegen. Mit Hilfe der RSX-Befehle ON und OFF kann der Cheksummer ein- bzw. ausgeschaltet werden.

Bei der Berechnung der Quersumme werden die Zeilennummer und Leerzeichen überlesen; zwischen Groß- und Kleinschreibung wird nur innerhalb zweier Anführungsstriche unterschieden, da ansonsten die Schreibweise von Befehlsworten und Variablennamen keinen Einfluß auf die Funktionstüchtigkeit eines Basicprogrammes hat.

(Thomas Fippl)



*Version 464*

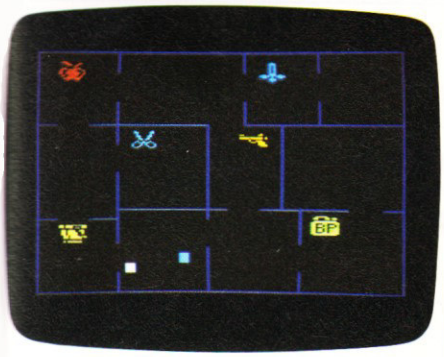
```
100 REM *** CHECKSUM 464 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,98,aa,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 06,dd,11,3a,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a2
310 DATA c1,cd,a3,e7,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,3c,c4,e5,09,e3,cd,63
340 DATA e1,21,a4,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,96,f2,e3,cd,f6
370 DATA a5,cd,4e,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,4e,c3,18,f5
390 DATA 3a,24,ac,d6,08,47,7e,a7
400 DATA c8,cd,45,e1,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,8a,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,5c,c3,3e,5b,cd,5c,c3
520 DATA cd,79,ee,3e,5d,c3,5c,c3
```

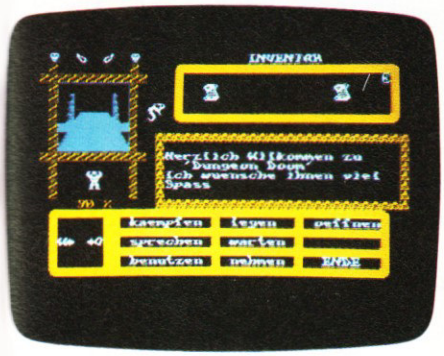
*Version 664*

```
100 REM *** CHECKSUM 664 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 4c,cb,11,5b,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a9
310 DATA c1,cd,69,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,75,c4,e5,09,e3,cd,59
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,58,f3,e3,cd,f6
370 DATA a5,cd,9b,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,9b,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,22,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a3,c3,3e,5b,cd,a3,c3
520 DATA cd,49,ef,3e,5d,c3,a3,c3
```

*Version 6128*

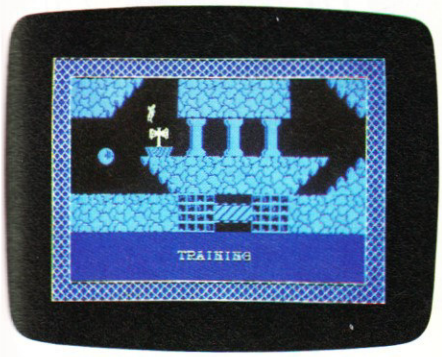
```
100 REM *** CHECKSUM 6128 ***
110 MEMORY &A4FF
120 FOR a%=&A500 TO &A607
130 READ byte$
140 POKE a%,VAL("&"+byte$)
150 NEXT
160 PRINT
170 PRINT"CHECKSUM v2"
180 CALL &A500:|ON:NEW
190 :
200 DATA 21,09,a5,01,0d,a5,c3,d1
210 DATA bc,00,00,00,00,18,a5,c3
220 DATA 2a,a5,c3,2f,a5,c3,43,a5
230 DATA 4f,ce,4f,46,c6,43,48,45
240 DATA 43,cb,00,00,cf,02,ac,c3
250 DATA a8,a5,21,27,a5,18,03,21
260 DATA 24,a5,28,06,cd,00,b9,c3
270 DATA 49,cb,11,5e,bd,01,03,00
280 DATA ed,b0,c9,4f,cd,00,b9,0d
290 DATA 28,08,0d,20,ea,dd,7e,02
300 DATA 18,04,7b,11,01,00,cd,a6
310 DATA c1,cd,64,e8,e5,4e,23,46
320 DATA 23,5e,23,56,e1,78,b1,c8
330 DATA cd,72,c4,e5,09,e3,cd,54
340 DATA e2,21,8a,ac,cd,7a,a5,e1
350 DATA 18,e2,e5,cd,ba,a5,e3,cd
360 DATA 98,a5,cd,53,f3,e3,cd,f6
370 DATA a5,cd,98,c3,e1,7e,a7,c8
380 DATA cd,98,a5,cd,98,c3,18,f5
390 DATA 3a,09,ac,d6,08,47,7e,a7
400 DATA c8,cd,1d,e2,23,10,f7,c9
410 DATA cd,24,a5,f5,c5,d5,e5,cd
420 DATA ba,a5,cd,f6,a5,e1,d1,c1
430 DATA f1,c9,eb,1b,af,47,67,6f
440 DATA 2f,32,23,a5,13,1a,d6,30
450 DATA 38,04,fe,0a,38,f6,1a,13
460 DATA a7,c8,4f,3a,23,a5,a1,fe
470 DATA 20,28,f3,79,fe,22,20,07
480 DATA 3a,23,a5,2f,32,23,a5,3a
490 DATA 23,a5,a7,79,c4,ab,ff,4f
500 DATA ad,07,6f,09,18,d8,3e,20
510 DATA cd,a0,c3,3e,5b,cd,a0,c3
520 DATA cd,44,ef,3e,5d,c3,a0,c3
```












# CPC Power-Spiele-Paket

**für CPC 464 · 664 · 6128**

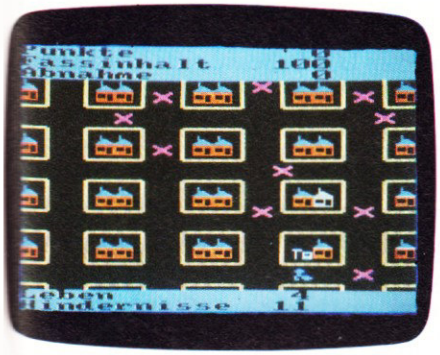


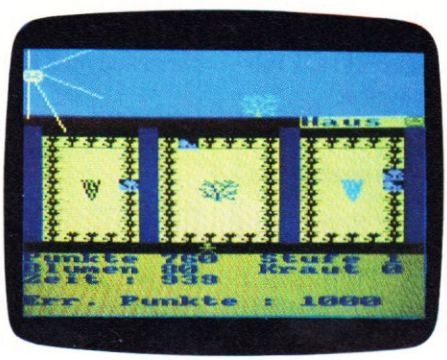

## Nur solange Vorrat reicht:

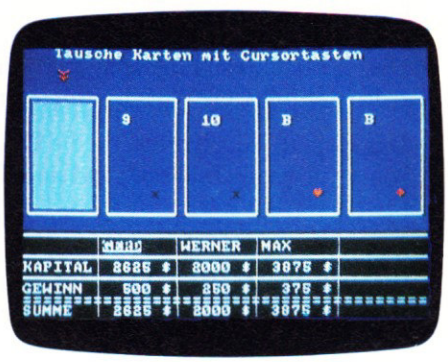
Die beliebten 4 Spielesammlungen zum Knüllerpreis
Goldene 7 ①, Goldene 7 ②
Gamebox 1, Gamebox 2
das bedeutet insgesamt **18** tolle Spiele für alle CPC Computer!

**Folgende Titel sind enthalten:**
Secret of Wizard, Turlen, Zadora, 3D Labyrinth, Die alte Burg, Space Race, Galaxis, Schatz, Garten Manager, Berg der Monster, Poker, Super Chance, Captain Starships Test, Oil Willi, Anduril, Mörderjagd, Ball Hunter, Dungeon Doom

Power-Spiele-Paket für CPC 464/664/6128 = 4 Kassetten nur **50,– DM**
= 4 Disketten 3" nur **70,– DM**

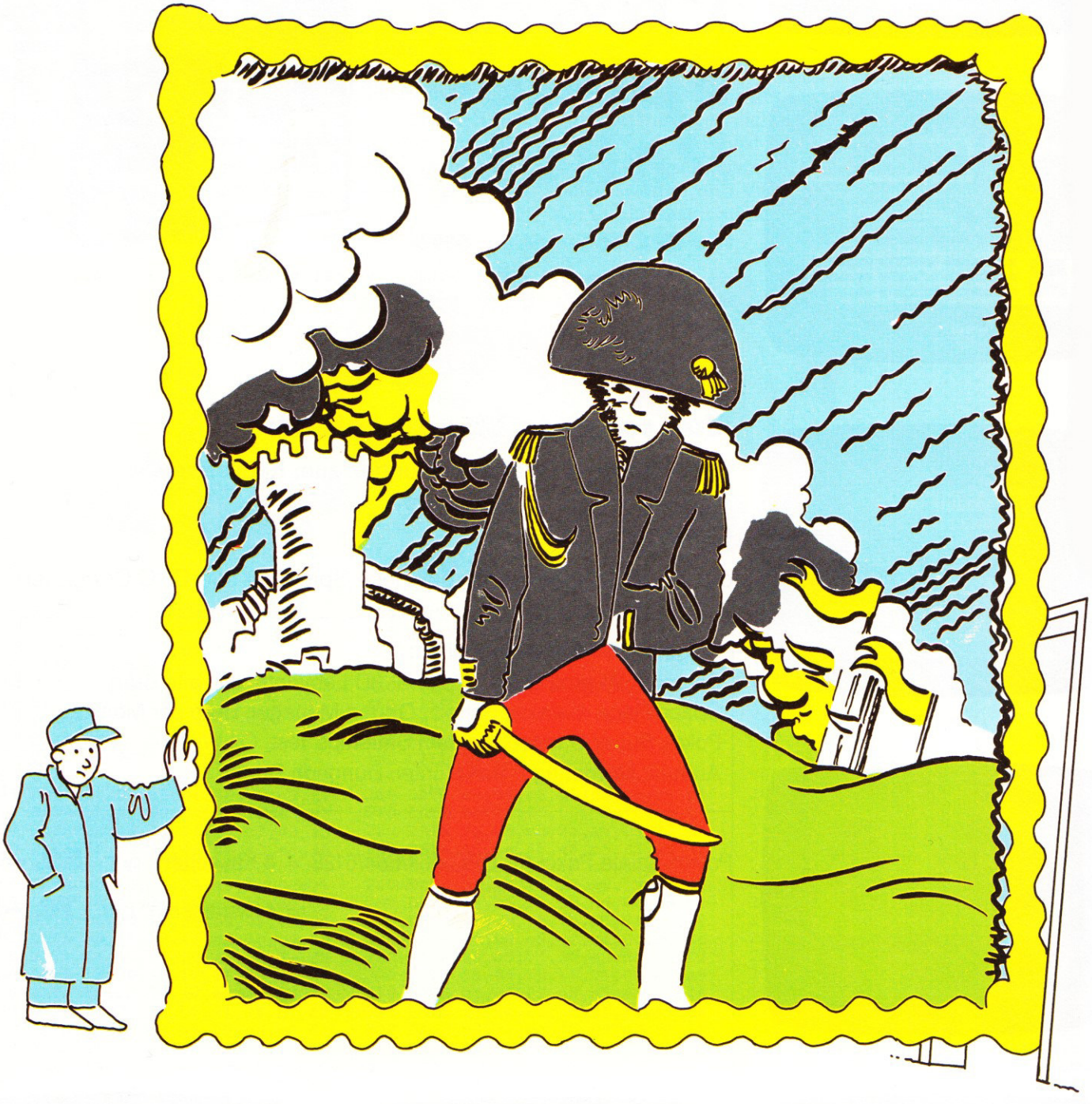







**DMV, Postfach 250, 3440 Eschwege**

# CONVERT.BAS



CONVERT.BAS setzt Bilder aus dem Programm Profi Painter (von Data Becker) in normale Bilder für den Schneider CPC um.

Die Idee zu diesem Programm kam mir, als ich ein Bild aus Profi-Painter mit COPYSHOP weiterverarbeiten wollte.

Das klappte jedoch nicht, da die Dokumente doppelt so groß wie ein normales Bild sind, und sie außerdem noch so abgespeichert sind, daß man sie mit dem LOAD-Befehl nicht laden kann.

## Wie's geht

Nach dem Start muß die Datendiskette aus Profi-Painter ins Laufwerk A eingelegt werden. Das Programm fragt dann nach dem Namen des Dokuments, und bringt eine Fehlermeldung, falls das Programm nicht im Directory gefunden wird.

Wenn das File gefunden wird, werden die belegten Blöcke in den Speicher ab Adresse 2600 Hexa eingelesen.

Danach werden die ersten 17 K des Bildes ins Video-RAM (ab Adresse C000 Hexa) übertragen, und mit einem SAVE-Befehl mit der Extension TB1 auf Diskette abgespeichert.

Genauso wird auch die zweite Hälfte des Bildes übertragen und mit der Extension TB2 abgespeichert. Die beiden Teilbilder können dann als normale Screen-Bilder weiterbearbeitet werden. (COPYSHOP, Hardcopy ,...)

Besitzer eines Zweitlaufwerks können im Programm (Zeile 380) den Wert der Variablen »drive« von 0 in 1 ändern, um die Datendiskette aus Profi Painter auch vom Laufwerk B einlesen zu können.

(Michael Fuchs)

## für 464-664-6128



```
10 ' ********************************        [1223]
20 ' *                              *        [175]
30 ' *   CONVERT.BAS                *        [882]
40 ' *                              *        [175]
50 ' *   Umsetzen eines Dokuments   *        [1162]
60 ' *   aus ProfiPainter in        *        [1487]
70 ' *   zwei Normale Bilder        *        [1868]
80 ' *                              *        [175]
90 ' *   (C) Michael Fuchs 1987     *        [1154]
100 '*                              *        [175]
110 '********************************        [1223]
120 '                                       [117]
130 '                                       [117]
140 '*******************************        [1252]
150 '       STEUER - SECTION                [1092]
160 '*******************************        [1252]
170 '                                       [117]
180 OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOU [2303]
T
190 MEMORY &25FF                            [538]
200 '                                       [117]
210 WHILE NOT progende                      [2156]
220 '                                       [117]
230    GOSUB 350:    ' Intitialisierung     [2060]
240    GOSUB 520:    ' Verarbeitung         [1199]
250    GOSUB 2330:   ' Test auf Programmend [1790]
e
260 '                                       [117]
270 WEND                                    [390]
280 END:             ' Programmende         [1307]
290 '                                       [117]
300 '********************************       [1223]
310 '          INITIALISIERUNG              [1300]
320 '********************************       [1223]
330 '                                       [117]
340 '                                       [117]
350 MODE 1                                  [506]
360 '                                       [117]
370 DIM block$(37):       ' Array fuer Bloc [2868]
knummern
380 drive=0:              ' Parameter zum S [1108]
ectorlesen
390 found=0                                 [896]
400 befehl=&84:           ' Disk-Befehl fue [2814]
r Sectorleser
410 RESTORE 1770                            [589]
420 GOSUB 1770:           ' Data's einpoken [2000]
430 '                                       [117]
440 RETURN:               ' Zurueck zur Ste [2848]
uer-Section
450 '                                       [117]
460 '                                       [117]
470 '********************************       [1285]
480 '          VERARBEITUNG                 [906]
490 '********************************       [1285]
500 '                                       [117]
510 '                                       [117]
520 GOSUB 780:      ' Eingabe-Dokument ausw [3637]
aehlen
530 GOSUB 1010:     ' Directory nach Filen  [2906]
amen durchsuchen
540 '                                       [117]
550 IF found=0 THEN RETURN: ' Dateiname ni  [3126]
cht gefunden !
560 '                                       [117]
570 GOSUB 1350:     ' Einlesen der Bloecke  [2441]
des File's
580 '                                       [117]
590 PRINT:PRINT "Bitte Diskette fuer Ausga  [4635]
be einlegen"
600 PRINT "          <TASTE>":CALL &BB18:!  [2546]
DIR
610 '                                       [117]
620 !USER,0:MODE 1                          [851]
630 GOSUB 1930:  ' Teilbild 1 ins Video-Ra  [3898]
m poken
640 SAVE FILE$+".TB1",b,&C000,&3FFF: ' und  [2159]
 abspeichern
650 '                                       [117]
660 offs=0: start=&6481: ziel=&C000: ' Wer  [4610]
te fuer 2.Teilbild aendern
670 '                                       [117]
680 MODE 1                                  [506]
690 GOSUB 1940: ' Teilbild 2 ins Video-Ram  [2289]
 poken
700 SAVE FILE$+".TB2",b,&C000,&3FFF: ' und  [3248]
 abspeichern
710 '                                       [117]
720 RETURN: ' Zurueck zur Steuer-Section    [2848]
730 '                                       [117]
740 '********************************       [1391]
750 '    Auswaehlen des Dokuments           [1555]
760 '********************************       [1391]
770 '                                       [117]
780 MODE 1                                  [506]
790 LOCATE 1,10:PRINT "Datendiskette aus P  [5023]
rofi-Painter einlegen"
800 LOCATE 18,12:PRINT "<TASTE>":CALL &BB1  [1645]
8
810 '                                       [117]
820 CLS: !USER,2: !DIR: ' Diskettenformat   [2655]
festlegen
830 '                                       [117]
840 PRINT:PRINT "Welches Dokument einlesen  [3122]
 "
850 INPUT "(ohne .PIC)              : ";in$  [1214]
860 '                                       [117]
870 in$=UPPER$(in$):file$=in$               [1251]
880 '                                       [117]
890 IF LEN(in$)>8 THEN PRINT CHR$(7): GOTO  [4415]
 820:       ' Name zu lang ?
900 IF INSTR(in$,".")<>0 THEN PRINT CHR$(7  [5051]
):GOTO 820: ' Punkt eingegeben ?
910 IF LEN(in$)<8 THEN l%=8-LEN(in$):in$=i  [3812]
n$+STRING$(l%," ")
920 '                                       [117]
930 in$=in$+"PIC": ' Extension mit anhaeng  [3006]
en
940 '                                       [117]
950 RETURN:         ' Zurueck zur Verarbeit [2501]
ung
960  '                                      [117]
970 '********************************       [1391]
980 '   Directory und Blocks einlesen       [1946]
990 '********************************       [1391]
1000 '                                      [117]
1010 bufbeg=9728: bi=1                      [1825]
1020 track=0: sector=&C1: found=0           [1081]
1030 '                                      [117]
1040 WHILE sector < &C6: ' Bis Sector &C5   [3006]
nach Eintrag suchen
```

```
1050 '                                          [117]
1060     GOSUB 2230: ' Sektor in Buffer ei [3035]
nlesen
1070 '                                          [117]
1080     FOR adr=&2600 TO &27FF STEP 32     [2447]
1090       user=PEEK(adr)                    [734]
1100       IF user<>2 THEN 1160: ' Dokumen [3883]
t steht unter Usernr.2
1110       name$=""                          [404]
1120       FOR i=1 TO 11                     [314]
1130         name$=name$+CHR$(PEEK(adr+i)) [2971]
: ' Filename zusammensetzen
1140       NEXT i                            [375]
1150       IF in$=name$ THEN found=-1:GOSU [4002]
B 1260:    ' Richtiger Filename ?
1160     NEXT adr                          [547]
1170     GOSUB 2120: ' Sektor+1            [1710]
1180 WEND                                  [390]
1190 IF found=0 THEN PRINT:PRINT "Datei st [4599]
eht nicht auf der Diskette !!":
                        PRINT "
 <TASTE>":CALL &BB18
1200 RETURN:          ' Zurueck zur Verarbe [2501]
itung
1210 '                                          [117]
1220 ' ********************************       [1223]
1230 '      BLOCKNUMMERN EINLESEN              [1204]
1240 ' ********************************       [1223]
1250 '                                          [117]
1260 FOR i=adr+16 TO adr+31                [1358]
1270   IF HEX$(PEEK(i))="0" THEN 1290 ELSE [4850]
 block$(bi)=HEX$(PEEK(i)):bi=bi+1
1280 NEXT i                                [375]
1290 RETURN                                [555]
1300 '                                          [117]
1310 ' ********************************       [1252]
1320 '   BLOCKS IN BUFFER EINLESEN         [1603]
1330 ' ********************************       [1252]
1340 '                                          [117]
1350 bufbeg=9728                           [967]
1360 FOR i=1 TO bi-2                       [588]
1370 '                                          [117]
1380     block$=block$(i)                  [1037]
1390 '                                          [117]
1400     GOSUB 1670: ' Blocknummer in Trac [3831]
k und Sektor umrechnen
1410     GOSUB 2230: ' 1.Blockhaelfte einl [2466]
esen
1420 '                                          [117]
1430     bufbeg=bufbeg+480: ' Bufferanfang [3677]
 fuer 2.Blockhaelfte festlegen
1440     high=INT(bufbeg/256):low=bufbeg-h [1892]
igh*256
1450 '                                          [117]
1460     POKE &A608,high:POKE &A607,low: ' [4071]
 und neue Werte einpoken
1470     POKE &AF57,high:POKE &AF56,low    [2040]
1480     POKE &A60A,high:POKE &A609,low    [1358]
1490 '                                          [117]
1500     GOSUB 2120: ' Sector + 1          [1905]
1510     GOSUB 2230: ' 2.Blockhaelfte einl [1223]
esen
1520 '                                          [117]
1530     bufbeg=bufbeg+480: ' Bufferanfang [4960]
 fuer naechsten Block festlegen
1540     high=INT(bufbeg/256):low=bufbeg-h [1892]
igh*256
1550 '                                          [117]
1560     POKE &A608,high:POKE &A607,low: ' [4071]
 und neue Werte einpoken
1570     POKE &A60A,high:POKE &A609,low    [1358]
1580     POKE &AF57,high:POKE &AF56,low    [2040]
1590 NEXT i                                [375]
1600 '                                          [117]
1610 RETURN: ' Zurueck zur Verarbeitung    [2501]
1620 '                                          [117]
1630 ' ********************************       [1252]
1640 '     BLOCKNUMMER UMRECHNEN           [2152]
1650 ' ********************************       [1252]
1660 '                                          [117]
1670 block$="&"+block$: a=VAL(block$)      [1229]
1680 track=INT(a*2/9)+0                    [953]
1690 sector=a*2-track*9+0*9                [1275]
1700 sector=sector+&C1                     [1566]
1710 RETURN                                [555]
1720 '                                          [117]
1730 ' ********************************       [1143]
1740 '        DATA'S  EINPOKEN             [1141]
1750 ' ********************************       [1143]
1760 '                                          [117]
1770 FOR i=&AF2A TO &AF5C                  [673]
1780 READ byte: POKE i,byte: NEXT          [1463]
1790 DATA &dd,&46,&05,&dd,&4e,&04,&dd,&66, [3780]
&03,&dd,&6e,&02,&dd,&56,&01,&dd
1800 DATA &5e,&00,&ed,&b0,&c9,&00          [1276]
1810 DATA &21,&00,&A6,&CD,&D4,&BC,&22,&01, [2056]
&A6,&79,&32,&03,&A6,&21,&04,&A6
1820 DATA &5E,&23,&56,&23,&4E,&21,&00,&26, [2237]
&DF,&01,&A6,&C9,&2D
1830 '                                          [117]
1840 FOR i=&A600 TO &A60C                  [1421]
1850 READ byte: POKE i,byte: NEXT          [1463]
1860 DATA &84,&00,&00,&00,&00,&01,&00, [3170]
&26,&FF,&27,&00,&5E
1870 RETURN                                [555]
1880 '                                          [117]
1890 '********************************     [1252]
1900 '   Bild ins Video-Ram poken          [1714]
1910 '********************************     [1252]
1920 '                                          [117]
1930 offs=0: ziel=&C000: start=&2601       [1311]
1940 FOR y=1 TO 25                         [1112]
1950   IF y=1 THEN quelle=start: ziel=&C00 [3989]
0 ELSE ziel=zielalt+&50
1960   zielalt=ziel                        [1346]
1970   FOR yy=1 TO 8                       [1570]
1980     quelle=quelle+offs                [1095]
1990     CALL &AF2A,79,quelle,ziel: ' Pixe [3148]
lzeile verschieben
2000     quelle=quelle+79:ziel=ziel+79     [1882]
2010     offs=1                            [259]
2020     ziel=ziel+&801-&50                [1040]
2030   NEXT yy                             [469]
2040 NEXT y                                [359]
2050 RETURN: ' Zurueck zur Verarbeitung    [2501]
2060 '                                          [117]
2070 ' ********************************     [1285]
2080 '    SECTOR UND TRACK ERRECHNEN       [3328]
2090 ' ********************************     [1285]
2100 '                                          [117]
2110 '                                          [117]
2120 ' *** SECTOR + 1                      [1127]
2130 sector=-(sector<&C9)+sector+((sector= [5380]
&C9)*(PEEK(&A8A0+(drive*64))-1))
2140 IF sector=&C1 THEN GOTO 2160 ELSE RET [2363]
URN
2150 '                                          [117]
2160 ' *** TRACK + 1                       [721]
2170 track=0-(track<39)+track+((track=39)* [2668]
39):RETURN
2180 '                                          [117]
2190 ' ****************************       [1164]
2200 '          SECTOR LESEN              [1066]
2210 ' ****************************       [1164]
2220 '                                          [117]
2230 POKE &A600,befehl                     [335]
2240 POKE &A604,drive                      [1083]
2250 POKE &A605,track                      [580]
2260 POKE &A606,sector-&C1+PEEK(&A89F+driv [1101]
e*&40)
2270 CALL &AF40: RETURN                    [987]
2280 '                                          [117]
2290 ' ****************************       [1164]
2300 '    TEST AUF PROGRAMMENDE           [766]
2310 ' ****************************       [1164]
2320 '                                          [117]
2330 MODE 1:                               [443]
2340 '                                          [117]
2350 PRINT:PRINT "Noch ein Dokument umsetz [3250]
en [J/N]"
2360 INPUT jn$                             [538]
2370 '                                          [117]
2380 IF UPPER$(jn$)<>"J" AND UPPER$(jn$)<> [4832]
"N" THEN PRINT CHR$(7):GOTO 2330
2390 '                                          [117]
2400 IF UPPER$(jn$) = "J" THEN progende=0: [2555]
 ERASE block$
2410 '                                          [117]
2420 IF UPPER$(jn$) = "N" THEN progende=-1 [1810]
2430 '                                          [117]
2440 RETURN: ' Zurueck zur Steuer-Section  [2848]
```

# DISKTOOL

Dieses Programm wendet sich an alle, die ihre VORTEX-Floppy noch besser ausnutzen wollen. Die einprägsamen RSX-Befehle lassen den Umgang mit der Floppy zum Genuß werden. Ein kurzer Befehl und Ihre Floppy lädt Programme bis zu 20% schneller; nun können Sie (auch während des Programmierens in BASIC!) mal schnell die CP/M-Systemspuren oder einzelne Files kopieren, sich ihre Header anzeigen lassen, Tracks und Sektoren laden, speichern, hexadezimal in ASCII anzeigen lassen, wichtige Laufwerksparameter einfach auslesen und verändern – die Möglichkeiten sind fast nur durch Ihre eigene Kreativität begrenzt (einige Ansätze können Sie in der DEMO sehen . . . )

Nun aber zu den Befehlen und ihrer Handhabung: da es sich um eine RSX-Befehlserweiterung handelt, muß vor jedem neuen Befehl ein senkrechter Balken (SHIFT + Klammeraffe) stehen. Besonders anwenderfreundlich sind u. a. die Fehlermeldung »Parameter Error« bei falscher Parameteranzahl und der **LIST**-Befehl, der alle neuen Befehle auflistet **und dann auf einen Tastendruck wartet**.

Zuerst zu den Befehlen, die die Floppy-Parameter verändern: Mit **SET. LV, x** können Sie die Anzahl der Leseversuche festlegen. (Problematischen Disketten kann man mit 255 Leseversuchen nachhelfen). Mit **SET. HZ, x** setzen Sie die Hochlaufverzögerungszeit des Floppymotors. (Der Motor braucht immer einige Sekunden, bis er die vereinbarte Drehzahl erreicht hat). Mit **SET. NZ, x** wählen Sie, wie lange der Motor nach einem Zugriff nachlaufen soll. (Wenn oft auf Diskette zugegriffen wird, ist eine lange Nachlaufzeit günstig, da so die Hochlaufzeit eingespart wird, weil der Motor ja noch läuft). **SET. SP, x** setzt die Wartezeit des Kopfes, da er immer etwas braucht, um sich zu beruhigen. Wenn Sie diesen Wert z. B. auf 10 setzen, erhöht sich der SPeed der Floppy ganz schön . . .

Die Befehle **GET. LV,** @ **GET. HZ,** @**, GET. NZ,** @ und **GET. SP,** @ lesen die Werte für Leseversuche, Hoch- und Nachlaufzeit des Motors, und den SPEED der Floppy aus. Wichtig ist, daß die mit »@« übergebene Variable x% vor dem ersten Befehlsaufruf bereits irgendeinen Wert zugewiesen bekommen haben MUSS, da es sonst zur Fehlermeldung kommt (also gleich am Programmanfang x%=0 eingeben). Mit **GET. DRIVE,** @ fragen Sie ab, welches Laufwerk gerade angeschlossen ist. x%=0 bedeutet, daß Laufwerk A, x%=1, daß Laufwerk B gerade selektiert ist. **GET. USER,** @ übergibt den aktuellen User(bereich) in x%. Den »krönenden Abschluß« dieser Befehlsgruppe bilden »FDISC« und »SDISC«: **FDISC** (FastDISC) macht Ihre Floppy bis zu 20% schneller und **SDISC** stellt wieder den Ausgangszustand her.

Kommen wir nun zu den Befehlen, die auf die Diskette zugreifen. Gleich zu Anfang möchten wir sie aber warnen, »Spielen« Sie nie mit einer Disk, von der Sie keine Sicherheitskopie haben! Das könnte katastrophale Folgen haben. Hoffen wir aber, daß Sie dies vermeiden und beschreiben wir jetzt die Befehle: **READ. SECTOR, sect, trk, drv, buff** liest den Sektor **sect** von Track **trk** in Laufwerk **drv** nach **buff**. Bei VORTEX sind die Sektornummern 1,2,. . . ,9 und die Tracks von 0 bis 159 nummeriert (ab Track 80 wird auf die andere Seite der Diskette zugegriffen). »Hinter« bzw. »über« **buff** müssen noch 512 Bytes für den Sektorinhalt frei sein. Um z.B. den ersten Sektor des Inhaltsverzeichnisses (das sich auf Track 2 befindet) nach 20000 zu laden, geben Sie ein READ. SECTOR, 2, 1,**0**, 20000 für Laufwerk **A** und READ. SECTOR, 2, 1, 1, 2000 für Laufwerk **B**. Analog zu READ. SECTOR gibt es den **WRITE. SECTOR, sect, trk, drv, buff**-Befehl, der (es ist kaum zu erraten) einen Sector im RAM ab **buff** auf Diskette speichert. Die Parameter geben an, welcher Sektor der Disk beschrieben werden soll; sie entsprechen denen von READ. SECTOR. Da es natürlich sehr umständlich und unkomfortabel ist, mit den Sektorbefehlen ganze Tracks (TRACK= eine der 160 Spuren, auf denen sich die jeweils neun Sektoren befinden) zu laden und zu SAVEn, gibt es die Befehle **READ. TRACK, trk, drv, buff** und **WRITE. TRACK, trk, drv, buff**. Der durch **trk** und **drv** definierte Track wird entweder nach **buff** gelesen (bei READ. TRACK) oder von da aus auf Diskette geschrieben (bei WRITE. TRACK). »trk« und »drv« entsprechen den SECTOR-Parametern, bei »buff« ist zu beachten, daß darüber bzw. dahinter 4096 Bytes (4KB) für den Trackinhalt frei sein müssen. Der **SEEK. TRACK, trk, drv**-Befehl sucht (suchen=engl. to **seek**) einen Track. Das heißt, daß der angegebene Track angefahren wird. Durch geschickte Positionierung des Kopfes kann eine Menge Zeit gespart werden. Wie schon oben gesagt, gibt es bei VORTEX 160 Tracks, die von 0-159 durchnummeriert sind. Andere Tracknummern können unvorhersehbare Folgen haben – also lieber nicht zu riskant experimentieren;

Weil man einen Sektor aber nicht nur laden und speichern, sondern erst recht untersuchen, also ansehen möchte, gibt es den Befehl **DUMP. SECTOR, sect, trk, drv, buff**. Er liest einen Sector mit den gleichen Parametern wie READ. SECTOR, zeigt ihn aber danach gleich hexadezimal und in ASCII an – er DUMPt den Sektor, wie man sagt. (Wer will, kann so z. B. die CP/M-Fehlermeldungen suchen und ins Deutsche übersetzen. . . ). Die Unterbrechungsmöglichkeiten während des DUMPens sind dieselben wie in BASIC, einmal ESC=Stop, das zweite Mal ESC=Abbruch und jede andere Taste=weiter DUMPen. Damit man auch sonst von der DUMP-Routine profitieren kann, gibt es den »reinen« DUMP-Befehl: **DUMP, adr**:er DUMPt einen Speicherbereich ab adr bis zum Abbruch durch zweimal ESC.

Neben den eben beschriebenen »Low Level«-Disk-Befehlen gibt es natürlich noch »High-Level«-Befehle: **HEADER, filename$** zum Beispiel. Dieser Befehl gibt den »Header« des Files »filename$« an. Unter einem Header versteht man den »Kopfsatz« eines Files, in dem seine Startadresse, Länge, Autostartadresse (engl. »Entry«) und sein Typ (BASIC, binär, ASCII, unbekannt) vermerkt sind. Wundern Sie sich nicht, daß ASCII-Files, auch wenn sie bei CAT z. B. 11K lang sind, eine Länge von Null im Header haben. Bei diesen Files definiert sich das VDOS nämlich (um CP/M-kompatibel zu bleiben) einen eigenen Header, der nicht so einfach ausgelesen werden kann.

Die restlichen »High-Level«-Befehle enthalten alle »COPY«, sie dienen also alle dazu, etwas zu kopieren. Das beginnt bei **FCOPY, filename$**, das einzelne Files kopiert (folgen Sie einfach den Anweisungen) und geht über **DCOPY**, das die gesamte Diskette mit Track-Anzeige kopiert (folgen Sie auch hier den Anweisungen) bis zu **CPMCOPY**, das ein einfaches Kopieren der CP/M-Spuren ermöglicht (auch hier den Anweisungen folgen). Nur **DCOPY** löscht Ihr gerade im Speicher stehendes BASIC-Programm mit 100%tiger Sicherheit – **CPMCOPY** verschont Ihr Programm, solange es nicht über ca. 22K lang ist (was nicht sehr oft passieren dürfte). Wenn Sie mit **FCOPY** kopieren, wird Ihr aktuelles BASIC-Programm überschrieben, wenn Sie ein BASIC (».BAS«)-Programm kopieren. Bei Binärprogrammen (». BIN«) kann Ihr Programm, es muß aber nicht, überschrieben werden. (Dies als Information um Anfragen und Mißverständnissen gleich vorzubeugen).

So, das wären dann alle 24 VORTEX-DISKTOOL V1. 0 – Befehle. Wie gesagt muß vor jedem Befehl der »senkrechte Strich«, den man mit SHIFT + Klammeraffe erreicht, stehen, da es sich um eine RSX-Befehlserweiterung handelt.

## Alle Befehle auf einen Blick

Zum Abschluß nochmal eine kurze Zusammenfassung der Befehle, ihrer Parameter und Wirkungen:
**LIST** neue Befehle auflisten

### Parameter setzen:

| | | | |
|---|---|---|---|
| **SET. LV, x** | x Leseversuche | **SET. NZ, x** | x ms Nachlaufzeit |
| **SET. SP, x** | x ms Kopfwartezeit | **FDISC** | Floppy max. 20% schneller |
| **SET. HZ, x** | x ms Hochlaufzeit | **SDISC** | Floppy normal |

### Parameter auslesen:

| | | | |
|---|---|---|---|
| **GET. LV, @** | x=Leseversuche | **GET. NZ, @** | x=Nachlaufzeit |
| **GET. SP, @** | x=Kopfwartezeit | **GET. DRIVE, @** | x=akt. Laufwerk |
| **GET. HZ, @** | x=Hochlaufzeit | **GET. USER, @** | x=aktueller User |

### »Low-Level«-Disk-Befehle:

**READ. SECTOR, sect, trk, drv, buff**
Sektor sect von Track trk von Laufwerk drv nach buff einlesen
**WRITE. SECTOR, sect, trk, drv, buff**
Sektor sect von Track trk von Laufwerk drv aus buff neu beschreiben
**READ. TRACK, trk, drv, buff**
Track trk von Laufwerk drv nach buff einlesen
**WRITE. TRACK, trk, drv, buff**
Track trk von Laufwerk drv aus buff neubeschreiben
**SEEK. TRACK, trk, drv**
Track trk von Drive drv anfahren
**DUMP. SECTOR, sect, trk, drv, buff**
Sektor wie bei READ.SECTOR lesen und dazu noch hexadezimal und in ASCII anzeigen
(»sect« von 1 bis 9, »trk« von 0 bis 159, »drv«=0=Laufwerk A, =1=Laufwerk B aktiv)

### »High-Level«-Befehle

**DUMP, adr** RAM ab adr hexadezimal und in ASCII anzeigen
**HEADER, filename$** Header des Files filename$ anzeigen
**FCOPY, filename$**File filename$ kopieren
**DCOPY** gesamte Diskette kopieren
**CPMCOPY** CP/M-Spuren kopieren

Für alle, die sich mit den neuen Befehlen so richtig vertraut machen wollen, ist die DEMO da. In ihr werden fast alle Befehle benutzt und einige der vielen Möglichkeiten angedeutet, die sich durch das VORTEX-DISKTOOL V1. 0 eröffnen. Wenn es mal nicht angezeigt wird, drücken Sie bitte immer eine Taste, um zur nächsten Befehlsdemonstration zu gelangen.
Wir hoffen, daß Sie diese Erweiterung gebrauchen können und daß sie Ihnen genauso viel Freude bereitet wie uns.

(Jan-Mirko Maczewski)

```
1 ' ************************                [768]
2 ' * VORTEX DISKTOOL V1.0 *                [745]
3 ' *        D E M O       *                [194]
4 ' * -------------------- *                [1284]
5 ' * BY  M.Schroeder &    *                [2218]
6 ' *     J-M Maczewski    *                [999]
7 ' * -------------------- *                [1284]
8 ' ************************                [768]
9 '                                         [117]
10 MODE 1:IF PEEK(&A100)=33 THEN 30         [1937]
20 MEMORY &A0FF:LOAD"disktool.bin":CALL &A  [2159]
100
30 INK 1,26:INK 3,0:INK 2,24:BORDER 13:INK  [3799]
 0,13:|GPEN,3
40 FOR i=100 TO 200 STEP 2:PLOT 50,i:DRAWR  [3974]
 450,0:NEXT
50 |GPEN,3:FOR i=100 TO 200 STEP 2:PLOT 50  [3429]
2,i:DRAWR 98,120:NEXT
60 FOR i=50 TO 500 STEP 2:PLOT i,202:DRAWR  [3193]
 98,120:NEXT
70 |GPEN,1:MOVE 50,202:DRAW 148,322:DRAW 6  [4303]
02,322:DRAW 502,202
80 MOVE 602,322:DRAW 602,222:DRAW 502,102   [2228]
90 |GPEN,1:x1=50:y1=100:x2=500:y2=200:GOSU   [2665]
B 160
100 x1=75:y1=140:x2=475:y2=150:GOSUB 160    [2218]
110 x1=80:y1=165:x2=115:y2=175:GOSUB 160    [1541]
120 FOR i=80 TO 115 STEP 2:PLOT i,165:DRAW  [3038]
R 0,10:NEXT
130 LOCATE 10,3:PEN 2:PRINT"VORTEX DISKTOO  [3336]
L V1.0"
140 LOCATE 17,21:PEN 1:PRINT"D E M O":LOCA  [3315]
TE 13,23:PEN 3
150 PRINT CHR$(7)"TASTE DRUECKEN !":CALL &  [4500]
BB18:GOTO 170
160 MOVE x1,y1:DRAW x2,y1:DRAW x2,y2:DRAW   [2944]
x1,y2:DRAW x1,y1:RETURN
170 CALL &BC02:PEN 2:MODE 1:INK 2,15,6:INK  [3496]
 3,26:LOCATE 18,10
180 PRINT" 24 neue RSX Befehle":LOCATE 24,  [2774]
12:PRINT" fuer die"
190 LOCATE 23,14:PRINT"Vortex-Floppy"       [2804]
220 LOCATE 1,1:PEN 1:|LIST:|FAST:INK 0,13:  [3437]
INK 1,0:MODE 2:BORDER 13:PEN 1
230 ' * Track-&Sektor-Monitor               [755]
240 MODE 2:t=1:s=1:d=0:WINDOW#1,8,72,1,1    [2652]
250 PAPER #1,1:CLS #1:PEN #1,0:WINDOW #2,8  [4449]
,72,24,24:CLS #2:|SET.NZ,500
260 PRINT#2,"  Commands: ";CHR$(240);" T.u   [3119]
p  ";CHR$(241);
270 PRINT#2," T.down  ";CHR$(242);" S.down  [4993]
  "CHR$(243);" S.up  D Drive  E Exit
280 WINDOW 4,78,3,22:PRINT#1,"   DMON (c)   [3636]
1987 by M.S & J-M.M";SPC(3)
290 PRINT#1,"Track:";HEX$(t,2);"  Drive:";  [4328]
CHR$(65+d);" Sector:";HEX$(s,2)
300 a$="":WHILE a$="":a$=INKEY$:WEND:SOUND  [2643]
 7,80,20,6
310 IF ASC(a$)>=240 THEN 370                [1638]
320 IF UPPER$(a$)="E" THEN 440              [951]
330 IF UPPER$(a$)="D" THEN d=d XOR 1:GOSUB  [3203]
 340:GOTO 260 ELSE 300
340 PRINT#1,"   DMON (c) 1987 by M.S & J-M  [7233]
 M   Track:";HEX$(t,2);"  Drive:";CHR$(65+
d);" Sector:";HEX$(s,2)
350 PRINT#2,SPC(20)"*** Liste Sektorinhalt  [4316]
 ***"CHR$(18):CLS
360 |DUMP.SECTOR,t,s,d,&5000:RETURN         [3017]
370 ON ASC(a$)-239 GOTO 380,390,400,420     [1340]
380 t=t+1:IF t>159 THEN t=0:GOSUB 340:GOTO  [3470]
 260 ELSE 410
390 t=t-1:IF t<0 THEN t=159:GOSUB 340:GOTO  [2433]
 260 ELSE 410
400 s=s-1:IF s<1 THEN s=9:GOTO 390          [886]
410 GOSUB 340:GOTO 260                      [1312]
420 s=s+1:IF s>9 THEN s=1:GOTO 380          [2171]
430 GOSUB 340:GOTO 260                      [1312]
440 ' * Neuer Cat mit Blocknr.anzeige       [1462]
450 MODE 2:PRINT SPC(8)"Blockbelegung alle  [5281]
r Files auf Diskette ermitteln"
460 PRINT:|READ.TRACK,2,0,&5000             [1355]
470 FOR s=&5000 TO &6200 STEP 32            [1499]
480 IF PEEK(s+1)=&E5 THEN 575               [1211]
490 a=(s-&5000)/&20+1:PRINT USING "###";a;  [1382]
500 PRINT SPC(3);"User ";HEX$(PEEK(s),2);S  [1717]
PC(4)
510 FOR i=s+1 TO s+12:IF PEEK(i)=0 THEN 54  [2025]
0
520 PRINT CHR$(PEEK(i));:IF i-s=8 THEN PRI  [2869]
NT".";
530 NEXT                                    [350]
540 PRINT SPC(3);"Block(s):";:FOR i=s+&10   [3494]
TO s+&20
550 IF PEEK(i)=0 OR PEEK(i)=&E5 THEN 570    [2244]
560 PRINT HEX$(PEEK(i),2);:IF PEEK(i+1)<>0  [2365]
 THEN PRINT"/";
570 NEXT:PRINT:NEXT                         [1354]
575 PRINT:PRINT"Bitte eine Taste druecken   [4889]
...":CALL &BB18
580 ' * Disk- & File-Info                   [796]
590 MODE 2:PRINT" Info-File  (c) J-M.M/M.S  [3242]
 '87
600 a%=0:|GET.USER,@a%:b%=0:|GET.DRIVE,@b%  [3094]
610 PRINT:PRINT"Laufwerk"SPC(6)": "CHR$(b%  [3431]
+65)
620 PRINT"Aktueller User:";a%:|GET.HZ,@a%:  [6421]
```

```
PRINT"Hochlaufzeit  :"a%:|GET.NZ,@a%
630 PRINT"Nachlaufzeit  :"a%:|GET.LV,@a%:P [7718]
RINT"Leseversuche  :"a%:|GET.SP,@a%
640 PRINT"Kopfberuhigung:"a%:PRINT:PRINT C [3582]
HR$(7);
650 INPUT"Geben Sie den Filenamen ein oder [4436]
 <ENTER> fuer Optionen:",a$
660 WINDOW 1,80,10,25:CLS                 [1188]
670 IF a$="" THEN 690                     [793]
680 |HEADER,a$:CALL &BB18:GOTO 590        [1522]
690 PRINT" Optionen :                     [1243]
700 PRINT:PRINT" 1 . Catalog              [1531]
710 PRINT" 2 . Laufwerk wechseln          [2717]
720 PRINT" 3 . User wechseln"             [2300]
730 PRINT" 4 . Exit Prg."                 [1385]
740 PRINT:PRINT" ?"                       [748]
750 a$=INKEY$:IF a$="" THEN 750           [1426]
760 PRINT:PRINT                           [743]
770 a=VAL(a$):ON a GOTO 790,800,810,820   [2308]
780 CLS:GOTO 690                          [983]
790 MODE 2:|DIR:CALL &BB18:GOTO 590       [3132]
800 INPUT"Laufwerk:",a$:IF UPPER$(a$)="A" [3839]
THEN |A:GOTO 590 ELSE |B:GOTO 590
810 INPUT"User:",a$:|SELECT,a$:GOTO 590   [1575]
820 END                                   [110]
```

```
1 DATA 210AA6,011BA1,CDD1BC,21ADA3,559    [2228]
2 DATA CD11A2,21B0A3,CD11A2,21B3A3,5EB    [2784]
3 DATA C311A2,65A1C3,43A2C3,56A2C3,6A2    [1689]
4 DATA 69A2C3,7CA2C3,ADA2C3,4CA2C3,772    [1860]
5 DATA 5FA2C3,72A2C3,B7A2C3,8FA2C3,7AB    [1561]
6 DATA 9EA2C3,D3A5C3,85A2C3,C2A2C3,84F    [2025]
7 DATA D0A2C3,07A3C3,0EA3C3,15A3C3,691    [1556]
8 DATA 20A3C3,F2A5C3,B3A4C3,B6A3C3,816    [2506]
9 DATA 27A5C3,96A553,45542E,4CD653,559    [1426]
10 DATA 45542E,48DA53,45542E,4EDA53,47E   [2494]
11 DATA 45542E,53D047,45542E,445249,3D7   [1517]
12 DATA 56C547,45542E,4CD647,45542E,459   [1311]
13 DATA 48DA47,45542E,4EDA47,45542E,466   [1221]
14 DATA 555345,D24644,4953C3,534449,488   [1731]
15 DATA 53C34C,4953D4,474554,2E53D0,503   [1830]
16 DATA 44554D,D04455,4D502E,534543,3F5   [2179]
17 DATA 544FD2,524541,442E53,454354,3EE   [1356]
18 DATA 4FD257,524954,452E53,454354,409   [2047]
19 DATA 4FD252,454144,2E5452,4143CB,460   [1759]
20 DATA 575249,54452E,545241,43CB53,401   [1785]
21 DATA 45454B,2E5452,4143CB,484541,3C6   [2323]
22 DATA 4445D2,46434F,50D944,434F50,482   [1533]
23 DATA D94350,4D434F,50D900,E5CDD4,5FA   [2035]
24 DATA BCDDE1,DD7500,DD7401,DD7102,66E   [1459]
25 DATA C9FE04,200BEB,DD5E02,DD4E04,54D   [1962]
26 DATA DD5606,C9D121,9BA3C3,92A3FE,728   [721]
27 DATA 0320F5,EBDD5E,02DD56,04C9FE,63E   [2033]
28 DATA 0120EA,7B3264,BEC9FE,0120E1,5A3   [2538]
29 DATA EB3A64,BE77C9,FE0120,D7ED53,6BD   [908]
30 DATA 44BEC9,FE0120,CEEB3A,44BE77,656   [2000]
31 DATA C9FE01,20C4ED,5346BE,C9FE01,6B8   [1876]
32 DATA 20BBEB,3A46BE,77C9FE,0120B1,614   [2707]
33 DATA 7B3249,BEC9FE,0120A8,EB3A49,5B2   [1624]
34 DATA BE77C9,212800,2244BE,3E0132,3DC   [1930]
35 DATA 64BE3E,0518E3,213200,2244BE,3D7   [2207]
36 DATA 3E1032,64BE3E,1918D4,FE0120,404   [2232]
37 DATA 80EB3A,06A777,C9FE01,C231A2,626   [2013]
38 DATA EB3A05,A777C9,FE01C2,31A232,5D7   [1195]
39 DATA 09A6EB,22DEA2,180DCD,21A2AF,5A0   [2045]
40 DATA 3209A6,22DEA2,DFADA3,210000,4D3   [1446]
41 DATA 062011,1000CD,3DA319,3E0DCD,325   [1754]
42 DATA 5ABB3E,0ACD5A,BBCD09,BB3006,506   [1413]
43 DATA CD18BB,FEFCC8,10E53A,09A6B7,6F7   [1066]
44 DATA C818D9,CD21A2,DFADA3,C9CD21,72F   [2655]
45 DATA A2DFB0,A3C921,ADA322,34A3CD,6D4   [1720]
46 DATA 37A218,0921B0,A32234,A3CD37,46B   [1671]
47 DATA A20E00,0C79FE,0AC8D5,E5C5DF,663   [1975]
48 DATA 0000C1,E1D124,2418EE,C5E5D5,640   [1878]
49 DATA E5CD6F,A30603,CD8AA3,06107E,55B   [1390]
50 DATA CD74A3,3E09CD,5ABB23,10F406,53A   [1708]
51 DATA 03CD8A,A3E106,103E01,CD5ABB,515   [1549]
52 DATA 7ECD5A,BB2310,F4D1E1,C1C97C,73F   [2138]
53 DATA CD74A3,7D5F1F,1F1F1F,CD7DA3,529   [2005]
54 DATA 7BE60F,FE0A38,02C607,C630C3,538   [2395]
55 DATA 5ABB3E,09CD5A,BB10F9,C97ECD,65B   [1240]
56 DATA 5ABBB7,C82318,F70750,617261,551   [1402]
57 DATA 6D6574,65722D,466568,6C6572,4A0   [1123]
58 DATA 008400,008500,008700,00FE01,28F   [1511]
59 DATA C231A2,212CA4,CD92A3,CD18BB,628   [2635]
60 DATA CDC9A3,1818DD,5601DD,5E00EB,5C3   [1628]
61 DATA 46235E,2356EB,2201A6,ED4303,427   [2224]
62 DATA A6CD77,BCC9D5,F5C5D5,111F00,703   [1345]
63 DATA 195E23,56ED53,07A6D1,EBCD83,5E9   [1073]
64 DATA BCCD7A,BC211E,A4CD92,A3CD18,689   [2135]
65 DATA BB2A01,A6ED4B,03A6CD,8CBCD1,653   [1370]
66 DATA F1E1ED,4B07A6,CD98BC,CD8FBC,7F0   [1469]
67 DATA 214CA4,C392A3,426974,746520,521   [2007]
68 DATA 205A69,656C07,0D0042,697474,35B   [1486]
69 DATA 652051,75656C,6C6469,736B65,498   [1044]
70 DATA 747465,206569,6E6C65,67656E,4B4   [1792]
71 DATA 3A070D,004B6F,706965,72766F,39D   [1668]
72 DATA 726761,6E6720,626565,6E6465,492   [1204]
73 DATA 742E20,202020,202020,0A0D00,199   [1155]
74 DATA 0D0A46,696C65,6E616D,653A00,372   [1759]
75 DATA 537461,727420,20203A,004C61,355   [1862]
76 DATA 656E67,652020,3A0045,6E7472,3B2   [1229]
77 DATA 792020,203A00,446174,656974,36E   [1309]
78 DATA 79703A,005472,61636B,200020,358   [1644]
79 DATA 6B6F70,696572,74120D,0A00FE,425   [1571]
80 DATA 01C231,A2CDC9,A33200,A6C5D5,641   [785]
81 DATA E5111F,0019EB,1A6F13,1A6722,358   [2458]
82 DATA FEA521,6CA4CD,92A3E1,23CD92,739   [1645]
83 DATA A3CD1A,A52178,A4CD92,A3E1DF,72E   [2369]
84 DATA 24A5CD,1AA521,82A4CD,92A3E1,67F   [1033]
85 DATA DF24A5,CD1AA5,218CA4,CD92A3,687   [1570]
86 DATA 2AFEA5,DF24A5,CD1AA5,2196A4,65C   [2851]
87 DATA CD92A3,3A00A6,6F2600,DF24A5,51F   [2330]
88 DATA CD1AA5,CD7ABC,3E0ACD,5ABB3E,5F7   [1935]
89 DATA 0DC35A,BB79EE,000614,ED5B06,4B4   [1562]
90 DATA A71600,C5D5CD,63A5D1,CD82A5,6F1   [2030]
91 DATA D5E521,A0A4CD,92A37A,D6086F,6E8   [1697]
92 DATA 2600DF,24A53E,2DCD5A,BB6A2D,4B2   [2123]
93 DATA 2600DF,24A521,A7A4CD,92A3E1,61D   [3288]
94 DATA D1C110,CFC318,A4212C,A4CD92,640   [2281]
95 DATA A3CD18,BB21AD,A32234,A32100,4CE   [2046]
96 DATA 010608,C5D5CD,29A3D1,14C110,4F8   [1936]
97 DATA F6C921,1EA4CD,92A3CD,18BB21,665   [1888]
98 DATA B0A322,34A321,000118,DFED5B,4AD   [1213]
99 DATA 06A716,00212C,A4CD92,A3CD18,49B   [971]
100 DATA BB21AD,A32234,A32100,50CD29,48C  [1137]
101 DATA A314CD,29A321,1EA4CD,92A3CD,602  [945]
102 DATA 18BB16,0021B0,A32234,A32100,377  [1798]
103 DATA 50CD29,A314CD,29A3C3,18A421,536  [1454]
104 DATA 65A13E,7CCD5A,BB7EF5,CBBFCD,76C  [1907]
105 DATA 5ABBF1,23CB7F,28F3CD,1AA57E,698  [1335]
106 DATA B7CA18,BB18E4,FE02C2,31A2DD,6C2  [2109]
107 DATA 5602DF,B3A3C9,000000,000000,356  [2423]
150 '                                     [117]
160 ' ***  VORTEX DISKTOOL V1.0  ***      [2214]
170 '       written in 1987 by            [1190]
175 '                                     [117]
180 '  Jan-Mirko  und  Martin             [1798]
190 '  Maczewski       Schroeder          [2782]
195 '                                     [117]
200 MODE 1:MEMORY &A0FF:adr=&A100:z=0     [1575]
210 PRINT"Bitte warten ... lese DATAs     [3240]
220 FOR i=adr TO &A5F0 STEP 12            [1103]
230  FOR ii=0 TO 3:READ b$(ii):NEXT       [2737]
240  p$=b$(0)+b$(1)+b$(2)+b$(3)           [1388]
250  FOR ii=0 TO 11                       [803]
260   b=VAL("&"+MID$(p$,(ii*2)+1,2))      [2052]
270   POKE i+ii,b:sum=sum+b                [1013]
280  NEXT ii:z=z+1                        [999]
290  READ c$:IF sum<>VAL("&"+c$) THEN PRIN [4436]
T CHR$(7)"DATAFEHLER in"z:END
300 sum=0:NEXT                            [685]
310 '!!! CPC 664/6128-User bitte eingeben  [2108]
:"POKE &A523,&49:POKE &a524,&EF"
320 PRINT"DATAs OK.Bitte Kass/Disk zum SAV [6767]
En ein- legen !":CALL &BB18
330 SAVE"disktool.bin",b,&A100,&500        [2281]
340 CALL &A100 'INIT                      [729]
350 PRINT"Neue Befehle :":|LIST           [2727]
```

# Ready to use Tip

## SIEVE OF ERASTOSTHENES
## Von 0 auf 8190 in 0,9 Sekunden

**Wie man Primzahlen per Computer berechnen kann, zeigt das Sieb des Erastosthenes. Dieser Algorithmus ist sehr leicht umzusetzen ...**

...sofern man eine geeignete, mehr oder weniger strukturierte Hochsprache hat. BASIC und Pascal als bekannteste bieten sich da förmlich an. Daß dieses Programm aber (beinahe) genausoleicht in »der Königin der Programmiersprachen« (nur ein kleiner Scherz am Rande), in Assembler nämlich, programmiert werden kann, zeigt unser diesmaliges Programm. Vorausgesetzt muß natürlich werden, daß man zumindest Grundkenntnisse in Z80 Assembler mitbringt. Die Remarks zeigen jeweils, wie dieser Programmteil in einer imaginären Hochsprache aussehen würde.

Die Bytefelder könnten noch wesentlich platzsparender bearbeitet werden, nähme man sich statt einem Byte jeweils nur ein Bit vor. Das würde zwar nur mehr 1/8 des Speicherplatzes benötigen, aber die Berechnung ginge sehr schnell auf Kosten der Zeit.

*Dieses Programm, das etwas aus der Reihe der übrigen der Ready to use-Reihe tanzt, soll erstens dazu Mut machen, auch komplexere Probleme in Assembler anzugehen und zweitens zeigen, daß solche Programmierer »auch nur mit Wasser kochen« ...*

Eckehart Röscheisen

### für 464-664-6128

```
                        1020 ; SIEVE OF ERASTOSTHENES - Z80 Assembler V1.0
                        1030 ;
                        1040 ; (C) 26.01.1987 by COMPUTER project Software
                        1050 ;
                        1060
8000                    1070          org  #8000                ; START &8000
                        1080
8000                    1090          ent  $                    ;
                        1100
8000  3A0600            1110 start:   ld   a,(#0006)            ; (* Test CPC Version *)
8003  FE80              1120          cp   #80                  ;
8005  2006              1130          jr   nz,primes            ;
8007  2186B2            1140          ld   hl,#b286             ;
800A  223781            1150          ld   (patch+1),hl         ;
800D  21FE1F            1160 primes:  ld   hl,8190              ; size=8190 DIV 2
8010  A7                1170          and  a                    ;
8011  CB1C              1180          rr   h                    ;
8013  CB1D              1190          rr   l                    ;
8015  22B981            1200          ld   (size_v),hl          ;
8018  213A81            1210          ld   hl,text1             ; PRINT "Computing ..." (
* Header *)
801B  CDEE80            1220          call print                ;
                        1230
801E  210000            1240          ld   hl,0                 ; count=0
8021  22BB81            1250          ld   (cnt_v),hl           ;
8024  22BD81            1260          ld   (i_v),hl             ;
8027  E5                1270          push hl                   ;
8028  2AB981            1280          ld   hl,(size_v)          ; FOR i=0 TO size
802B  44                1290          ld   b,h                  ;
802C  4D                1300          ld   c,l                  ;
802D  21CA81            1310          ld   hl,flag$             ;
8030  3601              1320          ld   (hl),1               ;   flag(i)=1
8032  54                1330          ld   d,h                  ;
8033  5D                1340          ld   e,l                  ;
8034  13                1350          inc  de                   ;
8035  EDB0              1360          ldir                      ; NEXT i
8037  2AB981            1370          ld   hl,(size_v)          ; FOR i=0 TO size
803A  44                1380          ld   b,h                  ;
803B  4D                1390          ld   c,l                  ;
803C  E1                1400          pop  hl                   ;
803D  11CA81            1410 next2:   ld   de,flag$             ;
8040  19                1420          add  hl,de                ;
8041  7E                1430          ld   a,(hl)               ;   IF flag(i)
8042  B7                1440          or   a                    ;
8043  2837              1450          jr   z,not1               ;
8045  2ABD81            1460          ld   hl,(i_v)             ;     prime=i+i+3
8048  E5                1470          push hl                   ;
8049  29                1480          add  hl,hl                ;
804A  110300            1490          ld   de,3                 ;
804D  19                1500          add  hl,de                ;
804E  22BF81            1510          ld   (prm_v),hl           ;
8051  EB                1520          ex   de,hl                ;     k=i+prime
8052  E1                1530          pop  hl                   ;
8053  19                1540          add  hl,de                ;
8054  22C181            1550          ld   (k_v),hl             ;
                        1560          xor  a                    ;     WHILE k<size
8057  AF                1570 wend:    ld   hl,(k_v)             ;
8058  2AC181            1580          push hl                   ;
805B  E5                1590          ld   de,flag$             ;       flag(k)=0
805C  11CA81            1600          add  hl,de                ;
805F  19                1610          ld   (hl),a               ;
8060  77                1620          pop  de                   ;       k=k+prime
8061  D1                1630          ld   hl,(prm_v)           ;
8062  2ABF81            1640          add  hl,de                ;
8065  19                1650          ld   (k_v),hl             ;
8066  22C181            1660          ld   hl,(size_v)          ;     WEND
8069  2AB981            1670          ex   de,hl                ;
806C  EB                1680          ld   hl,(k_v)             ;
806D  2AC181            1690          and  a                    ;
8070  A7                1700          sbc  hl,de                ;
8071  ED52              1710          jr   c,wend               ;
8073  38E3              1720          ld   hl,(cnt_v)           ;     count=count+1
8075  2ABB81            1730          inc  hl                   ;
8078  23                1740          ld   (cnt_v),hl           ;   ENDIF
8079  22BB81            1750 not1:    ld   hl,(i_v)             ; NEXT i
807C  2ABD81            1760          inc  hl                   ;
807F  23                1770          ld   (i_v),hl             ;
8080  22BD81            1780          push hl                   ;
8083  E5                1790          and  a                    ;
8084  A7                1800          dec  hl                   ;
8085  2B                1810          sbc  hl,bc                ;
8086  ED42              1820          pop  hl                   ;
8088  E1                1830          jr   c,next2              ;
8089  38B2              1840
                        1850          ld   hl,text2             ; PRINT "Number of primes
808B  218B81            * "
808E  CDEE80            1860          call print                ;
8091  2ABB81            1870          ld   hl,(cnt_v)           ; PRINT count
8094  CDF780            1880          call pnum                 ;
8097  219E81            1890          ld   hl,text3             ; PRINT "List of primes (
Y/N)"
809A  CDEE80            1900          call print                ;
809D  CDEB60            1910          call crlf                 ; PRINT
80A0  32C381            1920          ld   (x),a                ; XCOR 1
80A3  CD06BB            1930 again:   call #bb06                ; again: GET i
80A6  CBAF              1940          res  5,a                  ; i=UPPER(i)
80A8  FE59              1950          cp   "Y"                  ; IF i="Y"
80AA  2805              1960          jr   z,ok                 ;   GOTO ok: ENDIF
80AC  FE4E              1970          cp   "N"                  ; IF i="N"
80AE  C8                1980          ret  z                    ;   END: ENDIF
80AF  18F2              1990          jr   again                ; ENDIF:GOTO again
                        2000
80B1  210200            2010 ok:      ld   hl,2                 ; PRINT 2
80B4  CDF780            2020          call pnum                 ;
80B7  CD1EB1            2030          call tab                  ; TAB
80BA  2AB981            2040          ld   hl,(size_v)          ; FOR i=0 TO size
80BD  44                2050          ld   b,h                  ;
80BE  4D                2060          ld   c,l                  ;
80BF  0B                2070          dec  bc                   ;
80C0  210000            2080          ld   hl,0                 ;
80C3  22BD81            2090          ld   (i_v),hl             ;
80C6  11CA81            2100 next3:   ld   de,flag$             ;   IF flag(i)
80C9  19                2110          add  hl,de                ;
80CA  7E                2120          ld   a,(hl)               ;
80CB  B7                2130          or   a                    ;
80CC  280E              2140          jr   z,not2               ;
80CE  2ABD81            2150          ld   hl,(i_v)             ;       PRINT i+i+3
80D1  29                2160          add  hl,hl                ;
80D2  110300            2170          ld   de,3                 ;
80D5  19                2180          add  hl,de                ;
80D6  CDF780            2190          call pnum                 ;
80D9  CD1EB1            2200          call tab                  ;       TAB
80DC  2ABD81            2210 not2:    ld   hl,(i_v)             ; NEXT i
80DF  23                2220          inc  hl                   ;
80E0  22BD81            2230          ld   (i_v),hl             ;
80E3  E5                2240          push hl                   ;
80E4  A7                2250          and  a                    ;
80E5  2B                2260          dec  hl                   ;
80E6  ED42              2270          sbc  hl,bc                ;
80E8  E1                2280          pop  hl                   ;
80E9  38DB              2290          jr   c,next3              ;
80EB  21B681            2300 crlf:    ld   hl,text4             ; PRINT  (* SUB write CR/
LF *)
80EE  7E                2310 print:   ld   a,(hl)               ;        (* SUB write str
ing *)
80EF  B7                2320          or   a                    ;
80F0  C8                2330          ret  z                    ; END
80F1  23                2340          inc  hl                   ;
80F2  CD5ABB            2350          call #bb5a                ;
80F5  18F7              2360          jr   print                ;
                        2370
80F7  111027            2380 pnum:    ld   de,10000             ;        (* SUB write num
ber *)
80FA  CD1281            2390          call subtst               ;
80FD  11E803            2400          ld   de,1000              ;
8100  CD1281            2410          call subtst               ;
8103  116400            2420          ld   de,100               ;
8106  CD1281            2430          call subtst               ;
8109  110A00            2440          ld   de,10                ;
810C  CD1281            2450          call subtst               ;
810F  110100            2460          ld   de,1                 ;
8112  3E2F              2470 subtst:  ld   a,47                 ;
8114  A7                2480          and  a                    ;
8115  ED52              2490 subcnt:  sbc  hl,de                ;
8117  3C                2500          inc  a                    ;
8118  30FB              2510          jr   nc,subcnt            ;
811A  19                2520          add  hl,de                ;
811B  C35ABB            2530          jp   #bb5a                ;
                        2540
811E  3AC381            2550 tab:     ld   a,(x)                ;        (* SUB write tab
ulator *)
8121  FE36              2560          cp   54                   ;
8123  200C              2570          jr   nz,notlf             ;
8125  AF                2580          xor  a                    ;
8126  32C381            2590          ld   (x),a                ;
8129  CD1BBB            2600          call #bb1b                ;
812C  DC18BB            2610          call c,#bb18              ;
812F  18BA              2620          jr   crlf                 ;
8131  C606              2630 notlf:   add  a,6                  ;
8133  32C381            2640          ld   (x),a                ;
8136  3227B7            2650 patch:   ld   (#b727),a            ;
8139  C9                2660          ret                       ;
                        2670
813A  040218            2680 text1:   defb 4,2,24
813D  20536965          2690          defm " Sieve Of Erastosthenes "
8155  1820A4            2700          defb 24,32,164
8158  31393837          2710          defm "1987 by COMPUTER project Software"
8179  0A0A0D            2720          defb 10,10,13
817C  436F6D70          2730          defm "Computing ..."
8189  0D00              2740          defb 13,0
                        2750
818B  4E756D62          2760 text2:   defm "Number of primes: "
819D  00                2770          defb 0
                        2780
819E  2E204C69          2790 text3:   defm ". List of primes ? (Y/N)"
81B6  0A0D00            2800 text4:   defb 10,13,0
                        2810
81B9                    2820 size_v:  defs 2                    ; (* Variablendeklaration
en *)
81BB                    2830 cnt_v:   defs 2                    ;
81BD                    2840 i_v:     defs 2                    ;
81BF                    2850 prm_v:   defs 2                    ;
81C1                    2860 k_v:     defs 2                    ;
81C3                    2870 x:       defs 1                    ;
81C4                    2880 string:  defs 6                    ;
                        2890
81CA                    2900 flag$:   equ  $                    ; (* Datenfeld fuer Primz
ahlen *)
                        2910

again  80A3  cnt_v  81BB  crlf   80EB
flag$  81CA  i_v    81BD  k_v    81C1
next2  803D  next3  80C6  not1   807C
not2   80DC  notlf  8131  ok     80B1
patch  8136  pnum   80F7  primes 800D
print  80EE  prm_v  81BF  size_v 81B9
start  8000  string 81C4  subcnt 8115
subtst 8112  tab    811E  text1  813A
text2  818B  text3  819E  text4  81B6
wend   8058  x      81C3

Table used:   348  from   529
Executes: 32768
```

# Ready to use Tip

## MC SOFTWARE CLOCK
## Des Schneiders Ührli

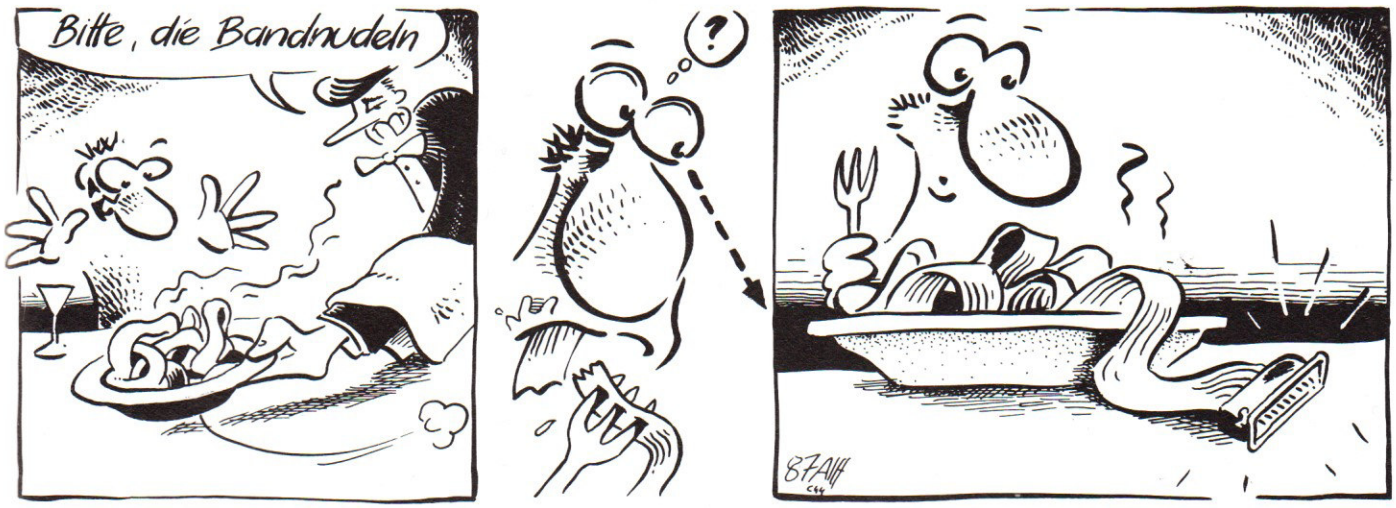
**Für Spiele, Anwendungen und andere Software, bei der Zeit eine Rolle spielt, kann der Schneider dank seiner ausgefeilten Interrupttechnik ein Ührli nebenher unterhalten.**

Die Uhr wird per clock init mit call clinit eingeschaltet. Sie kann mit call tzero auf 00:00:00 gestellt werden oder über set time mit call settim individuell gestellt werden: Beim Einsprung müssen A die Stunden, B die Minuten und C die Sekunden der einzustellenden Zeit enthalten sein. Über get time kann ein String in ein in HL angegebenes Workarea kopiert werden (call gettim). Dieser hat die Form HH:MM:SS und damit eine Länge von 8 Bytes, die dann als Characters via ×BB5A ausgegeben werden können.

(Eckehart Röscheisen)

### für 464-664-6128

```
                        1000
A000                    1010         org  #a000
                        1020
                        1030 ********************************
                        1040 ********************** SUB TIMER
                        1050 ****************** 464, 664, 6128
                        1060 ********** Digital realtime clock
                        1070 ********************** 94 Bytes
                        1080 ********************************
                        1090
A000  214FA0            1100 clinit: ld   hl,tickbl
A003  013200            1110         ld   bc,50
A006  50                1120         ld   d,b
A007  59                1130         ld   e,c
A008  C3E9BC            1140         jp   #bce9
                        1150
A00B  215DA0            1160 timint: ld   hl,time3
A00E  34                1170         inc  (hl)
A00F  7E                1180         ld   a,(hl)
A010  FE3C              1190         cp   60
A012  D8                1200         ret  c
A013  3600              1210         ld   (hl),0
A015  2B                1220         dec  hl
A016  34                1230         inc  (hl)
A017  7E                1240         ld   a,(hl)
A018  FE3C              1250         cp   60
A01A  D8                1260         ret  c
A01B  3600              1270         ld   (hl),0
A01D  2B                1280         dec  hl
A01E  34                1290         inc  (hl)
A01F  7E                1300         ld   a,(hl)
A020  FE18              1310         cp   24
A022  D8                1320         ret  c
A023  3600              1330         ld   (hl),0
A025  C9                1340         ret
                        1350
A026  AF                1360 tzero:  xor  a
A027  47                1370         ld   b,a
A028  4F                1380         ld   c,a
                        1390
                        1400 ; A = Hours   (00-24)
                        1410 ; B = Minutes (00-60)
                        1420 ; C = Seconds (00-60)
                        1430
A029  215BA0            1440 settim: ld   hl,time1
A02C  F3                1450         di
A02D  77                1460         ld   (hl),a
A02E  23                1470         inc  hl
A02F  70                1480         ld   (hl),b
A030  23                1490         inc  hl
A031  71                1500         ld   (hl),c
A032  FB                1510         ei
A033  C9                1520         ret
                        1530
                        1540 ; HL = String address for HH:MM:SS
                        1550
A034  115BA0            1560 gettim: ld   de,time1
A037  0603              1570         ld   b,3
A039  1A                1580 nxttim: ld   a,(de)
A03A  0E2F              1590         ld   c,47
A03C  0C                1600 nxtdez: inc  c
A03D  D60A              1610         sub  10
A03F  30FB              1620         jr   nc,nxtdez
A041  C63A              1630         add  a,58
A043  71                1640         ld   (hl),c
A044  23                1650         inc  hl
A045  77                1660         ld   (hl),a
A046  05                1670         dec  b
A047  C8                1680         ret  z
A048  23                1690         inc  hl
A049  363A              1700         ld   (hl),":"
A04B  23                1710         inc  hl
A04C  13                1720         inc  de
A04D  18EA              1730         jr   nxttim
                        1740
A04F                    1750 tickbl: defs 8
A057  00                1760         defb 0
A058  81                1770         defb #81
A059  0BA0              1780         defw timint
                        1790
A05B                    1800 time1:  defs 1
A05C                    1810 time2:  defs 1
A05D                    1820 time3:  defs 1
                        1830


clinit A000  gettim A034  nxtdez A03C
nxttim A039  settim A029  tickbl A04F
time1  A05B  time2  A05C  time3  A05D
timint A00B  tzero  A026

Table used:   152  from   233
```



# WERTPAPIERBILANZ

**Das vorliegende Programm ermöglicht dem Aktien- und Anleihenbesitzer eine einfach zu handhabende, übersichtliche Verwaltung seiner Wertpapiere und eine laufende Erfolgskontrolle der einzelnen Papiere und des eingesetzten Kapitals. Es hilft bei regelmäßiger Anwendung, fundierte Kauf- und Verkaufentscheidungen treffen zu können und nicht zuletzt wurde dabei an den jungen Sparer gedacht, der erst einmal »Trockenübungen« machen möchte.**

Erstellt wurde das Programm auf dem SCHNEIDER CPC 464, mit FLOPPY DISC DDI-I und DRUCKER NLQ 401. Es läuft jedoch auch mit Kassette.

Gegenüber dem Sparbuchsparer, der den Erfolg seines Sparens einmal im Jahr per Zinsgutschrift mitgeteilt bekommt, ist es für den Wertpapiersparer ungleich schwieriger, den Erfolg oder Mißerfolg seines eingesetzten Geldes zu beurteilen. Wertpapiere haben veränderliche Kurse! Käufe und Verkäufe werden mit Spesen belastet; der Bestand an Wertpapieren wird oftmals umgeschichtet; Renditen schwanken mehr oder weniger stark. Alle diese Dinge erschweren die Erfolgskontrolle.

Im vorliegenden, praxiserprobten Programm wurde nun folgender Weg beschritten, um trotzdem Erfolge oder Mißerfolge in DM oder % ausdrücken zu können: Am Beginn eines Jahres z.B. wird der Wert jeder einzelnen Wertpapierposition über den aktuellen Börsenkurs ermittelt. Alle Positionen zusammengenommen ergeben den Wert des Depot's an diesem Tage. Dazu kommt ein Kassenbestand, der zu weiteren Käufen dient und der anfangs mehr oder weniger willkürlich festgelegt werden kann. Depot und Kassenbestand addiert,ergeben das eingesetzte Kapital. Von diesem Kapital ausgehend, wird nun in regelmäßigen Zeitabständen der, aufgrund der veränderlichen Börsenkurse, neue Kapitalwert ermittelt und mit dem alten Kapitalwert verglichen. Die Differenz hieraus ist der Gewinn bzw. der Verlust. Bei Verkäufen verringert sich die Anzahl der Stücke und damit der Depotwert;gleichzeitig erhöht sich der Kassenbestand. Bei Käufen erhöht sich die Anzahl der Stücke und der Kassenbestand sinkt entsprechend. Käufe und Verkäufe sind demgemäß wertneutral. Es wird empfohlen, einmal wöchentlich einen Programmlauf durchzuführen und die dabei ausgedruckten Bilanzen zu sammeln.

Beim erstmaligen Start des Programms ist eine Wertpapieraufstellung anzufertigen. Der Befehl

< K-artei erstellen >

ist dafür zu benutzen. DM ist unbedingt einzugeben, weil sonst eine falsche Bilanzberechnung nach Stückzahl erfolgt. Bei späteren Programmstarts wird nur der Befehl

< L-aden >

benutzt, um als erstes die DATEI zu laden. Sie enthält alle wichtigen Daten der letzten Bilanz, wie Kurse, Depot, Kasse usw. Bei zwischenzeitlich erfolgten Änderungen der einzelnen Positionen ist sodann auf < K > zu schalten und die entsprechende Korrektur durchzuführen. Es erscheint nunmehr eine Übersicht über die vorhandenen Wertpapiere mit den jeweiligen Kursen der letzten Bilanz und dem Kassenbestand. Nach Drücken von < N > wird nach Datum, Änderung des Kassenbestands durch Käufe oder Verkäufe, Spesen, Zinsen etc. gefragt. Außerdem fragt das Programm nach Dollarkurs, Durchschnittsrendite, und FAZ-Index. Diese Daten sind aus den Wirtschaftsteilen der Zeitungen zu entnehmen und wichtig für Kauf- oder Verkaufentscheidungen; ihre Eingabe ist wahlfrei. Sodann werden die aktuellen Kurse abgefragt. Aus der folgenden Wertpapierbilanz (siehe Bild 1) können alle Daten, wie Gewinne,Depotwert, Kassenstand etc. entnommen werden. Diese auf Wunsch auszudruckende Bilanz ist sehr hilfreich bei Anlageentscheidungen. Die gesammelten Exemplare, die im Format bewußt klein gehalten wurden, zeigen die Vermögensentwicklung über einen längeren Zeitraum und lassen Anlagefehler klar erkennen. Vor Abschalten des Computers werden die neu eingegebenen Daten wieder in der DATEI abgespeichert. Bei einem erneuten Computerlauf rücken diese neuen Daten dann in die Spalte der alten Daten usw. .

Von Zeit zu Zeit empfielt sich ein Vergleich zum Jahresbeginn oder zu allen Kaufkursen. Dafür ist im Programm eine Korrekturmöglichkeit für den »Zeitraum« vorgesehen. Das Programm ist für die Verwaltung von 40 verschiedenen Wertpapieren eingerichtet. Es ist vorteilhaft, die maximal zu erwartende Anzahl (auch größer 40) einzugeben. Dies geschieht durch Änderung der Variablen »V« in Zeile 20. (Hans Mocho/SH)

**für 464-664-6128**

```
1 REM --- Programm: WERTPAPIERBILANZ          [2569]
2 REM --- SAVE"WEPABILZ"                      [1176]
3 REM --- DATEI: AKDATA                       [1403]
4 REM --- (c) by Hans Mocho,Gaertnerweg 14   [1982]
10 CLEAR:MODE 2:BORDER 5                      [2343]
20 V=40                                       [502]
30 DIM A(V,2),A$(V,2),B(V,2),C$(V)            [1933]
40 CLS                                        [91]
50 ON ERROR GOTO 2170                         [1481]
60 WINDOW#3,24,42,2,2                         [1168]
70 WINDOW#2,1,80,25,25                        [1376]
80 '                                          [117]
90 REM --- Menue ---                          [1277]
100 '                                         [117]
110 PRINT"================================== [5359]
==========================================
======";
120 PRINT"***              W  E  R  T  P      [3207]
A  P  I  E  R  B  I  L  A  N  Z
   ***";
130 PRINT"================================== [5359]
==========================================
```

```
======";
140 PRINT                                     [361]
150 PRINT" Dieses  Programm  ermoeglicht d [7323]
em Aktien- und Anleihenbesitzer eine einfa
ch zu"
160 PRINT" handhabende,  uebersichtliche   [7230]
Verwaltung seiner Wertpapiere und eine lau
fende"
170 PRINT" Erfolgskontrolle der einzelnen  [6570]
Papiere und des eingesetzten Kapitals. Es
hilft"
180 PRINT" bei  regelmaessiger  Anwendung, [6989]
   fundierte  Kauf-  und  Verkaufentscheid
ungen"
190 PRINT" treffen  zu  koennen  und nicht [7807]
 zuletzt ist es als  Ansporn  fuer junge S
parer"
200 PRINT" gedacht, die erst einmal 'Trock [5852]
enuebungen' machen wollen."
210 PRINT:PRINT" Es wird empfohlen, einmal [9179]
 woechentlich einen Programmlauf  durchzuf
uehren  und"
220 PRINT" die  dabei ausgedruckten  Bilan [7893]
zen zur kritischen Beurteilung  zu sammeln
. Das"
230 PRINT" Programm  ist fuer  die Verwalt [7689]
ung von 40  Wertpapieren  eingerichtet. Es
```

```
  ist"
240 PRINT" vorteilhaft, die  zu  erwartend [5541]
e  max. Anzahl  einzugeben, durch Aenderun
g der"
250 PRINT" Variablen 'V' in Zeile 20."     [2282]
260 PRINT" Der Befehl  < K-artei erstellen [7850]
 >  wird  nur fuer die  erstmalige  Wertpa
pier-"
270 PRINT" Aufstellung  benoetigt.  Dabei  [5893]
ist  ' DM '  unbedingt einzugeben,  weil
sonst"
280 PRINT" eine falsche Bilanzberechnung   [7596]
nach Stueckzahl erfolgt.  Bei spaeteren  S
tarts"
290 PRINT" wird  nur  der  Befehl  < L-ade [10161]
n >  benutzt  und  gegebenenfalls  der  B
efehl"
300 PRINT" < K-artei > bei erfolgten Besta [4752]
ndsaenderungen."
310 PRINT:PRINT TAB(50),"Hans Mocho, Gaert [4980]
nerweg 14"
320 PRINT TAB(50),"6000 Frankfurt a.M. 1"  [1903]
330 PRINT#2,CHR$(24)"Bitte eingeben:     < [5642]
 L-aden >              < K-artei erstelle
n >              "+CHR$(24);
340 D$=INKEY$:IF D$="" THEN 340            [1118]
350 IF D$<>"L" AND D$<>"1" AND D$<>"K" AND [3078]
 D$<>"k" THEN 340
360 IF D$="L" OR D$="1" THEN GOTO 630      [858]
370 IF D$="K" OR D$="k" THEN GOTO 410      [860]
380 '                                      [117]
390 REM --- Kartei Aufbau/Aenderung ---     [1668]
400 '                                      [117]
410 CLS:CLS#2                              [239]
420 GOSUB 2210                             [873]
430 CLS#2:PRINT#2,CHR$(24)"Bitte neue Date [7193]
n eingeben !
                        "+CHR$(24);
440 FOR i=0 TO 1500:NEXT                   [837]
450 GOSUB 2290                             [857]
460 I=0                                    [420]
470 I=I+1                                  [444]
480 n=I+4                                  [113]
490 IF I=21 THEN CLS:GOSUB 2210:GOSUB 2290 [1494]
500 IF I>=21 THEN n=I-16                   [891]
510 IF I=V+1 THEN 1110                     [744]
520 PRINT TAB(2);A(I,1);:PRINT TAB(15);A$( [4361]
I,1);:PRINT TAB(31);A$(I,2);:PRINT TAB(49)
;A(I,2)
530 LOCATE 62,n:INPUT C$(I)                [770]
540 IF C$(I)="K" OR C$(I)="k" THEN 590 ELS [4086]
E LOCATE 1,n:INPUT A(I,1)
550 IF C$(I)="" THEN GOTO 1110             [318]
560 LOCATE 13,n:INPUT A$(I,1)              [913]
570 LOCATE 29,n:INPUT A$(I,2)              [1500]
580 LOCATE 48,n:INPUT A(I,2)               [1504]
590 GOTO 470                               [454]
600 '                                      [117]
610 REM --- DATEI laden ---                 [1214]
620 '                                      [117]
630 OPENIN"AKDATA"                         [1000]
640 CLS:PRINT#2," DATEI wird geladen !"    [2551]
650 INPUT#9,altdat$                        [1262]
660 INPUT#9,GN                             [613]
670 INPUT#9,KN                             [845]
680 I=0                                    [420]
690 WHILE NOT EOF                          [1840]
700 I=I+1                                  [444]
710 INPUT#9,A(I,1)                         [923]
720 INPUT#9,A(I,2)                         [877]
730 INPUT#9,A$(I,1)                        [1064]
740 INPUT#9,A$(I,2)                        [1058]
750 INPUT#9,B(I,2)                         [915]
760 IF I=V THEN 780                        [991]
770 WEND                                   [390]
780 CLOSEIN                                [752]
790 GA=GN                                  [299]
800 KA=KN                                  [566]
810 FOR I=1 TO V                           [416]
820 B(I,1)=B(I,2)                          [1064]
830 B(I,2)=0                               [637]
840 NEXT                                   [350]
850 GOTO 1110                              [349]
860 '                                      [117]
870 REM --- Daten sichern ---               [1339]
880 '                                      [117]
890 OPENOUT"AKDATA"                        [802]
900 CLS#2                                  [372]
910 PRINT#2,CHR$(7),CHR$(24)" Daten werden [5643]
 gesichert !
                        "+CHR$(24);
920 PRINT#9,altdat$                        [939]
930 PRINT#9,GN                             [1028]
940 PRINT#9,KN                             [1068]
950 FOR I=1 TO V                           [416]
960 PRINT#9,A(I,1)                         [1058]
970 PRINT#9,A(I,2)                         [1060]
980 PRINT#9,A$(I,1)                        [1139]
990 PRINT#9,A$(I,2)                        [1141]
1000 PRINT#9,B(I,2)                        [1049]
1010 NEXT                                  [350]
1020 CLOSEOUT                              [902]
1030 CLS#2:PRINT#2,CHR$(24),CHR$(7)"Bitte  [7853]
eingeben:     < D-rucken >        < E-nde
>                              "+CHR$(24)
;
1040 D$=INKEY$:IF D$="" THEN 1040          [1252]
1050 IF D$<>"D" AND D$<>"d" AND D$<>"E" AN [3116]
D D$<>"e" THEN GOTO 1040
1060 IF D$="D" OR D$="d" THEN GOTO 1970    [1402]
1070 IF D$="E" OR D$="e" THEN GOTO 2020    [1540]
1080 '                                     [117]
1090 REM --- Gewinn / Verlust ---          [1776]
1100 '                                     [117]
1110 CLS:CLS#2                             [239]
1120 u=0                                   [277]
1130 PRINT#u,"W e r t p a p i e r b i l a  [2777]
n z"
1140 PRINT#u,"---------------------------- [3028]
--------------------------------"
1150 PRINT#u,"Datum: ";datum$;:PRINT#u,TAB [3900]
(26)"Vergleich zu ";altdat$
1160 PRINT#u:PRINT#u,"MENGE NAME           [4238]
     KURS/A  KURS/N  WERT    + %   KAUFKUR
S"
1170 S=0                                   [285]
1180 FOR I=1 TO V                          [416]
1190 IF A(I,1)=0 THEN GOTO 1320            [567]
1200 PRINT#u,TAB(1) USING"#####";A(I,1);   [1737]
1210 PRINT#u,TAB(7) A$(I,2);               [1184]
1220 PRINT#u,TAB(25) B(I,1);:PRINT#u,TAB(3 [3072]
3) B(I,2);
1230 IF A$(I,1)="DM" OR A$(I,1)="dm" THEN  [1191]
1260
1240 SU=A(I,1)*B(I,2)                      [803]
1250 PRINT#u,TAB(41) USING"######";SU;:GOT [2688]
O 1280
1260 SU=A(I,1)*B(I,2)/100                  [1189]
1270 PRINT#u,TAB(41) USING"######";SU;     [1700]
1280 IF B(I,2)=0 OR B(I,1)=0 THEN GOTO 130 [2346]
0;
1290 PRINT#u,TAB(48) USING"###.#";(B(I,2)- [2694]
B(I,1))/B(I,1)*100;
1300 S=S+SU                                [191]
1310 PRINT#u,TAB(55) A(I,2)                [1020]
1320 NEXT I                                [375]
```

```
1330 '                                          [117]
1340 REM --- Bilanz ---                         [862]
1350 '                                          [117]
1360 PRINT#u,"-----------Neu-------Alt--- [3324]
----------------------------------"
1370 PRINT#u,"Depot  = "USING"######";S;:P [3283]
RINT#u,"  = "USING"######";GA-KA;
1380 PRINT#u,TAB(34)"Dollarkurs(DM)";DK     [2653]
1390 KN=KA+KD                               [334]
1400 PRINT#u,"Kasse  = "USING"######";KN;: [5662]
PRINT#u,"  = "USING"######";KA;
1410 PRINT#u,TAB(34)"Rendite Oeff. ";RU;"/ [1623]
"RO
1420 GN=S+KA+KD                             [986]
1430 PRINT#u,"GESAMT = "USING"######";GN;: [5301]
PRINT#u,"  = "USING"######";GA;
1440 PRINT#u,TAB(34)"Rendite Pfand.";PU;"/ [1917]
"PO
1450 E=GN-GA:IF S=0 THEN E=0                [1325]
1460 P=E/GA*100                             [304]
1470 PRINT#u,"Gewinn : "USING"######";E;:P [4712]
RINT#u," DM /"USING"###.#";P;:PRINT#u," %"
;
1480 PRINT#u,TAB(34)"FAZ Akt.-Index";IA     [1864]
1490 PRINT#u,"Kaeufe/Verk./Bem.: ";KD;:PRI [4212]
NT#u,P$
1500 PRINT#u,"---------------------------- [3028]
----------------------------------"
1510 PRINT#u:IF u=8 THEN RETURN             [1469]
1520 PRINT#2,CHR$(24)"Bitte eingeben:   <   [6404]
N-eue Kurse >   < S-ichern >   < D-rucken
>   < K-artei >   "+CHR$(24);
1530 D$=INKEY$:IF D$="" THEN 1530           [1242]
1540 IF D$<>"N" AND D$<>"n" AND D$<>"K" AN [6749]
D D$<>"k" AND D$<>"S" AND D$<>"s" AND D$<>
"D" AND D$<>"d" THEN GOTO 1530
1550 IF D$="N" OR D$="n" THEN GOTO 1620     [1713]
1560 IF D$="K" OR D$="k" THEN GOTO 410      [860]
1570 IF D$="S" OR D$="s" THEN GOTO 890      [1531]
1580 IF D$="D" OR D$="d" THEN GOTO 1970     [1402]
1590 '                                      [117]
1600 REM --- Neue Kurse eingeben ---         [1530]
1610 '                                      [117]
1620 CLS:CLS#2                              [239]
1630 INPUT"Aktuelles  Datum      ";datum$   [1699]
1640 PRINT"Vergleich  zu        "USING"  ' [6224]
\               \'";altdat$
1650 PRINT#2," Anderer Vergleich gewuensch [4624]
t ?  < A >    Sonst beliebige Taste !"
1660 D$=INKEY$:IF D$="" THEN 1660           [1208]
1670 IF D$<>"A" AND D$<>"a" THEN 1700       [2464]
1680 IF D$="A" OR D$="a" THEN 1690          [1264]
1690 CLS#3:LOCATE 23,2:INPUT;altdat$        [1083]
1700 CLS#2                                  [372]
1710 PRINT"Kassenstand-Aenderung ? Wertpap [16066]
ierkaeufe, Spesen und  Gebuehren mit '-'
                          eingeben.
Verkaeufe, Zinsen und Dividenden werden
                       mit '+' eingege
ben !"
1720 INPUT"Bitte Eingeben   DM   ";KD       [2130]
1730 LINE INPUT"Bemerkungen          ? "; [2391]
P$
1740 PRINT"------------------------------ [4376]
----------------------------------------
"
1750 PRINT"Sonstige Eingaben     ? Die fol [12669]
genden Eingabedaten sind wahlfrei. Bitte
                          eingeben, o
der  < E N T E R > druecken !"
1760 PRINT:INPUT"Dollarkurs (DM)       ";D [1865]
K
1770 INPUT"Rendite Oeff. % min.  ";RU       [2302]
1780 INPUT"              % max.  ";RO       [1591]
1790 INPUT"Rendite Pfand.% min.  ";PU       [2178]
1800 INPUT"              % max.  ";PO       [1587]
1810 INPUT"FAZ Akt.-Index        ";IA       [476]
1820 CLS                                    [91]
1830 PRINT"Bitte die aktuellen Kurse einge [5285]
ben !"
1840 PRINT"------------------------------ [2712]
-----"
1850 PRINT"      NAME               KURS/A [1245]
  KURS/N "
1860 FOR I=1 TO V                           [416]
1870 PRINT TAB(1) USING"#####";A(I,1);      [2357]
1880 IF A(I,1)=0 THEN GOTO 1920             [783]
1890 PRINT TAB(7) A$(I,2);                  [333]
1900 PRINT TAB(25) B(I,1);                  [1223]
1910 PRINT TAB(34) :INPUT B(I,2)            [1129]
1920 NEXT                                   [350]
1930 GOTO 1110                              [349]
1940 '                                      [117]
1950 REM --- Drucken ---                     [1529]
1960 '                                      [117]
1970 PRINT#8,CHR$(27);CHR$(120);CHR$(0)     [2093]
1980 u=8:GOSUB 1130                         [1303]
1990 '                                      [117]
2000 REM --- Ende ---                       [959]
2010 '                                      [117]
2020 CLS                                    [91]
2030 PRINT#2,CHR$(24)" Sind Daten Gesicher [4478]
t ?  < J/N >
                  "+CHR$(24);
2040 D$=INKEY$:IF D$="" THEN 2040           [1276]
2050 IF D$<>"J" AND D$<>"j" AND D$<>"N" AN [2879]
D D$<>"n" THEN 2040
2060 IF D$="J" OR D$="j" THEN GOTO 2090     [1401]
2070 IF D$="N" OR D$="n" THEN GOTO 890      [1728]
2080 '                                      [117]
2090 CLS:CLEAR:BORDER 1                     [937]
2100 LOCATE 30,10:PRINT"================== [3147]
==="
2110 LOCATE 30,11:PRINT"*   E   N   D   E  [2972]
  *"
2120 LOCATE 30,12:PRINT"================== [3332]
==="
2130 LOCATE 1,24:END                        [1066]
2140 '                                      [117]
2150 REM --- UP Fehler ---                  [1128]
2160 '                                      [117]
2170 RESUME NEXT                            [703]
2180 '                                      [117]
2190 REM --- UP Karteiblatt ---             [2040]
2200 '                                      [117]
2210 PRINT"K a r t e i b l a t t "          [2015]
2220 PRINT"----------------------"          [1754]
2230 PRINT"  MENGE       DM oder St      N [3637]
AME              KAUFKURS   ";CHR$(24)
" Zeile O.K. ? "+CHR$(24)
2240 PRINT"------------------------------ [4496]
----------------------------------------
-"
2250 RETURN                                 [555]
2260 '                                      [117]
2270 REM --- UP Steuer-Daten ---            [2123]
2280 '                                      [117]
2290 CLS#2:PRINT#2,CHR$(24)" < K-orrekte Z [9451]
eile >      < Z-eilenanfang >      Eingabe
ende < 2 mal ENTER >    "+CHR$(24);
2300 RETURN                                 [555]
```

# Ready to use Tip

## SLIME

### Bildschirm löschen einmal auf andere Weise

**Wie löscht man den Bildschirm in Assembler? – Na, so zum Beispiel:**

```
cls: ld   hl,#c000 ; Startadresse
     ld   de,#c001 ; Verschiebeadr.
     ld   bc,#3fff ; Länge
     ld   (hl),0   ; 1.Byte löschen
     ldir          ; 0 kopieren ...
     ret           ; Fertig
```

So funktioniert der CLS-Befehl in der Maschinen-Ebene. Aber das ist wenig reizvoll.

*Wie wäre es einmal mit einer etwas anderen Lösch-Routine?*

SLIME löscht den Bildschirm von oben nach unten wie eine Art heruntergelassene Jalousie. Allerdings sollte der Bildschirm möglichst noch nicht gescrollt worden sein, denn dann sieht das ganze noch eine Spur besser aus.

Die Routine läuft auf allen drei CPC-Computern und benötigt weniger als 40 Bytes ...

Hermann Röscheisen

**für 464-664-6128**

```
                        1000
A000                    1010         org  #a000
                        1020
                        1030 ********************************
                        1040 ********************** SUB SLIME
                        1050 ******************** 464, 664, 6128
                        1060 ********************** Clear screen
                        1070 ************************ 38 Bytes
                        1080 ********************************
                        1090
A000  2100C0            1100 slime:  ld   hl,#c000
A003  0619              1110         ld   b,25
A005  E5                1120 loop3:  push hl
A006  C5                1130         push bc
A007  0608              1140         ld   b,8
A009  E5                1150 loop2:  push hl
A00A  C5                1160         push bc
A00B  0650              1170         ld   b,80
A00D  3600              1180 loop:   ld   (hl),0
A00F  23                1190         inc  hl
A010  10FB              1200         djnz loop
A012  CD19BD            1210         call #bd19
A015  C1                1220         pop  bc
A016  E1                1230         pop  hl
A017  7C                1240         ld   a,h
A018  C608              1250         add  a,8
A01A  67                1260         ld   h,a
A01B  10EC              1270         djnz loop2
A01D  C1                1280         pop  bc
A01E  E1                1290         pop  hl
A01F  115000            1300         ld   de,80
A022  19                1310         add  hl,de
A023  10E0              1320         djnz loop3
A025  C9                1330         ret
                        1340
loop   A00D  loop2  A009  loop3  A005
slime  A000

Table used:    60  from   164
```


# FÜR NIX GIBT'S NIX

**Aber für gute Programme und Tips & Tricks umso mehr**



**Für den Programmhit des Monats**

**1000,–**

**Und für den Top-Tip des Monats ganze**

**500,–**

**Das sind doch gute Argumente, Ihr Programm auch einmal zum Hit des Monats werden zu lassen.**
**Probieren geht über Studieren.**
**Bitte richten Sie Ihre Einsendungen an:**

**DMV Verlag, Fuldaer Str. 6, 3440 Eschwege**

# Befehlserweiterung für F1-X

**Wollen Sie beim Anschluß eines Vortex-Controllers lieber mit den AMSDOS-Befehlen arbeiten? Wenn ja, dann ist dieses Programm das richtige für Sie.**

Die Erweiterung der Befehle sieht wie folgt aus:
|Tape
|Tape. in
|Tape. out
|Drive, »A« (bzw.|Drive,»B«")
|User, x

Diese Befehle machen die Geräte-Konfiguration CPC + 3" - Erstlaufwerk + Vortex Fl-X wieder AMSDOS-Kompatibel. So brauchen alte Programme nicht angepaßt werden, wenn diese Routine vorher gestartet wurde.
Es kann auf die aufgeführten Vortex-Befehle verzichtet werden, ohne darauf verzichten zu müssen. Das heißt, Sie können wählen zwischen Vortex und AMSDOS-Befehlen.
Die Befehle sind:
CAS – CAS. IN – CAS. OUT – SELECT.
Aber das ist noch nicht alles, es wurden noch andere Befehle implementiert.
|READ, Anfang, Länge:
liest headerloses File von Cassette

|WRITE, Anfang, Länge:
schreibt headerloses File auf Cassette.
Z.B. kann mit folgendem Befehl der Bildschirm headerlos abgesaved werden:

|WRITE, &C000, &4000
und wieder laden mit
|READ, &C000, &4000
und dann gibt es noch den Befehl
|Load, »FILENAME«, adr

mit dem Binärfiles an jede Speicheradresse geladen werden können (z.B. unter HIMEM) oder an Adressen, an die man mit dem normalen LOAD-Befehl nicht gelangt, z.B. dez 40. Sollten Sie die Übersicht verloren haben, so können Sie alle Befehle mit dem kurzen Kommando |DOS auflisten und erklären lassen. Anzumerken wäre noch, daß der neue USER-Befehl komfortabler ist als der alte, es lassen sich statt USER 0 – 15 nun USER 0 – 255 verwenden.

Sie können auch, was nicht zu verachten ist, z.B. mit !USER, 229 auf mit |ERA, gelöschte Files zugreifen. Alle |TAPE-Kommandos greifen auf das AMSDOS-Floppy-Controller System zu. Das heißt, der Zugriff erfolgt auf die Epprom-Nummer 7. Darum alle Bastler: Wenn Sie die Nummer Ihres AMSDOS-Epproms verändert haben, funktionieren die Befehle nicht. Auch laufen die Befehle nicht mit anderen RSX-Befehlen zusammen, da bei mehreren Erweiterungen mehrmals die RSX-Routine #BCD1 aufgerufen wird. Das macht Ihr Computer nicht mit. Alle alten RSX-Befehle einschließlich VDOS-, bzw. AMSDOS-Befehle verschwinden bei einem doppelten Aufruf. Das liegt aber nicht an der Erweiterung, sondern an der Routine, die die Befehle in das Betriebssystem einbindet. Tippen Sie folgendes ab:

```
1. Lader für das Binär-File
1000 SYMBOL AFTER 256
1010 MEMORY &9FFF
1020 LOAD »dos. bin",&a000
1030 CALL &A000
1040 |DOS
```

und speichern Sie es ab. Danach müssen Sie die DATA-Zeilen 1000 – 1040 abtippen und starten und wenn alles geklappt hat wird das ganze als Binär-File abgesaved.

(H.–G. Esser/CD)

## für 6128

```
1000 DATA 01,14,A0,21,10,A0,CD,D1,BC,3E,C9 [3644]
,CE,32,00,A0,C9,10,A0,14,A0
1010 DATA 31,A0,C3,5E,A0,C3,62,A0,C3,66,A0 [3519]
,C3,6A,A0,C3,81,A0,C3,0C,A1
1020 DATA C3,CD,A0,C3,69,A1,C3,89,A1,54,41 [3841]
,50,C5,54,41,50,45,2E,49,CE
1030 DATA 54,41,50,45,2E,4F,55,D4,55,53,45 [2651]
,D2,44,52,49,56,C5,52,45,41
1040 DATA C4,57,52,49,54,C5,4C,4F,41,C4,44 [3481]
,4F,D3,00,DF,75,A0,C9,DF,78
1050 DATA A0,C9,DF,7B,A0,C9,DD,7E,00,DD,2A [3373]
,75,BE,DD,77,05,C9,FD,CC,07
1060 DATA 01,CD,07,18,CD,07,E4,CD,07,FE,01 [2976]
,20,1B,DD,66,01,DD,6E,00,23
1070 DATA 5E,23,56,1A,FE,41,28,27,FE,61,28 [4031]
,23,FE,42,28,28,FE,62,28,24
1080 DATA 21,AD,A0,7E,FE,00,C8,CD,5A,BB,23 [2632]
,18,F6,42,61,64,20,63,6F,6D
1090 DATA 6D,61,6E,64,0D,0A,00,DD,2A,75,BE [4276]
,DD,36,06,00,C9,DD,2A,75,BE
1100 DATA DD,36,06,01,C9,FE,02,20,CF,DD,7E [3625]
,03,67,DD,7E,02,6F,DD,7E,01
1110 DATA 57,DD,7E,00,5F,E5,D5,21,F3,A0,CD [3300]
,A3,A0,D1,E1,3E,00,CD,9E,BC
1120 DATA 30,5D,C9,53,61,76,69,6E,67,20,48 [3159]
,65,61,64,65,72,6C,65,73,73
1130 DATA 20,46,69,6C,65,0D,0A,00,FE,02,20 [2997]
,90,DD,7E,03,67,DD,7E,02,6F
1140 DATA DD,7E,01,57,DD,7E,00,5F,E5,D5,21 [1842]
,35,A1,CD,A3,A0,CD,06,BB,D1
1150 DATA E1,3E,00,CD,A1,BC,30,1B,C9,4C,6F [3884]
,61,64,69,6E,67,20,48,65,61
1160 DATA 64,65,72,6C,65,73,73,20,46,69,6C [2604]
,65,0A,0D,00,21,56,A1,CD,A3
1170 DATA A0,C9,48,65,61,64,65,72,6C,65,73 [2507]
,73,20,45,72,72,6F,72,0A,0D
1180 DATA 00,FE,02,C2,A0,A0,DD,66,03,DD,6E [3340]
,02,46,23,5E,23,56,EB,DD,56
1190 DATA 01,DD,5E,00,D5,CD,77,BC,E1,CD,83 [2796]
,BC,C9,21,8F,A1,C3,A3,A0,04
1200 DATA 02,0A,18,20,20,20,20,20,20,20,20 [2707]
,20,20,20,20,20,20,20,45,78
1210 DATA 74,65,6E,64,65,64,20,46,31,2D,58 [3504]
,20,44,6F,73,20,20,20,28,63
1220 DATA 29,20,31,39,38,37,20,62,79,20,48 [2828]
,61,6E,73,2D,47,65,6F,72,67
1230 DATA 20,45,73,73,65,72,20,20,20,20,20 [3005]
,20,20,20,20,20,20,20,20,20
1240 DATA 20,20,20,18,0D,0A,0A,20,46,6F,6C [3332]
,67,65,6E,64,65,20,41,4D,53
1250 DATA 44,4F,53,2D,42,65,66,65,68,6C,65 [3186]
,20,77,75,72,64,65,6E,20,69
1260 DATA 6E,20,64,61,73,20,56,44,4F,53,20 [4483]
,65,69,6E,67,65,62,75,6E,64
1270 DATA 65,6E,3A,0D,0A,0A,20,7C,54,41,50 [3336]
,45,2C,20,7C,54,41,50,45,2E
1280 DATA 49,4E,2C,20,7C,54,41,50,45,2E,4F [3477]
,55,54,2C,20,7C,55,53,45,52
1290 DATA 2C,78,20,3B,20,7C,44,52,49,56,45 [3164]
```

```
,2C,22,78,22,0D,0A,0A,20,7A
1300 DATA 75,73,61,65,74,7A,6C,69,63,68,65 [3564]
,20,42,65,66,65,68,6C,65,3A
1310 DATA 0D,0A,0A,20,7C,52,45,41,44,2C,20 [3172]
,73,74,61,72,74,2C,6C,61,65
1320 DATA 6E,67,65,20,20,20,20,20,2D,2D,3E [3196]
,20,20,6C,65,73,65,6E,20,65
1330 DATA 69,6E,65,73,20,68,65,61,64,65,72 [2161]
,6C,6F,73,65,6E,20,46,69,6C
1340 DATA 65,73,20,76,6F,6E,20,43,61,73,73 [3145]
,65,74,74,65,0D,0A,20,7C,57
1350 DATA 52,49,54,45,2C,73,74,61,72,74,2C [2601]
,6C,61,65,6E,67,65,20,20,20
1360 DATA 20,20,2D,2D,3E,20,20,73,63,68,72 [2845]
,65,69,62,65,6E,20,65,69,6E
1370 DATA 65,73,20,68,65,61,64,65,72,6C,6F [3575]
,73,65,6E,20,46,69,6C,65,73
1380 DATA 20,61,75,66,20,43,61,73,73,65,74 [2529]
,74,65,0D,0A,0A,20,7C,4C,4F
1390 DATA 41,44,2C,22,46,49,4C,45,4E,41,4D [2634]
,45,2E,45,58,54,22,2C,61,64
1400 DATA 72,65,73,73,65,0D,0A,0A,20,7C,44 [2981]
,4F,53,20,20,20,20,20,20,20
1410 DATA 20,20,20,20,20,20,20,20,20,20,20 [3075]
,20,20,2D,2D,3E,20,20,49,6E
1420 DATA 66,6F,72,6D,61,74,69,6F,6E,65,6E [3816]
,20,61,62,72,75,66,65,6E,0D
1430 DATA 0A,0A,0A,0A,00,00,00,00,00,00,00 [3194]
,00,00,00,00,00,00,00,00,00
1440 adr=&A000                              [332]
1450 ON ERROR GOTO 1500                     [1519]
1460 READ byte$                             [603]
1470 POKE adr,VAL("&"+byte$)                [745]
1480 adr=adr+1                              [392]
1490 GOTO 1460                              [391]
1500 SAVE "dos.bin",b,&A000,adr-&9FFF       [2516]
1510 END                                    [110]
```

```
                1000 ; Anpassung der Vortex F1-X an die AMSDOS-Befehle:
                1010 ; |TAPE ; |TAPE.IN ; |TAPE.OUT ; |USER ; |DRIVE
                1020 ;
                1030 ; |READ und |WRITE zum Laden und Schreiben
                1040 ; von headerlosen Files auf Cassette
                1050 ;
                1060 ; |LOAD laedt Binaer-Files von Cassette oder Diskette an
                1070 ; jede Adresse, auch wenn sie unter HIMEM liegt.
                1080 ;
                1090 ; (c) 1987 by Hans-Georg Esser
                1100 ;
A000            1110         org  #a000
A000            1120         ent  #a000
BCD1            1130 logext: equ  #bcd1
A000 0114A0     1140 extern: ld   bc,rsx
A003 2110A0     1150         ld   hl,kernel
A006 CDD1BC     1160         call logext
A009 3EC9       1170         ld   a,#c9
A00B CE         1180         defb "N"+#80
A00C 3200A0     1190         ld   (extern),a
A00F C9         1200         ret
A010            1210 kernel: defs 4
A014 31A0       1220 rsx:    defw table
A016 C35EA0     1230         jp   tape
A019 C362A0     1240         jp   tapein
A01C C366A0     1250         jp   tapout
A01F C36AA0     1260         jp   user
A022 C381A0     1270         jp   drive
A025 C30CA1     1280         jp   read
A028 C3CDA0     1290         jp   write
A02B C369A1     1300         jp   load
A02E C389A1     1310         jp   dos
A031 544150     1320 table:  defm "TAP"
A034 C5         1330         defb "E"+#80
A035 54415045   1340         defm "TAPE.I"
A03B CE         1350         defb "N"+#80
A03C 54415045   1360         defm "TAPE.OU"
A043 D4         1370         defb "T"+#80
A044 555345     1380         defm "USE"
A047 D2         1390         defb "R"+#80
A048 44524956   1400         defm "DRIV"
A04C C5         1410         defb "E"+#80
A04D 524541     1420         defm "REA"
A050 C4         1430         defb "D"+#80
A051 57524954   1440         defm "WRIT"
A055 C5         1450         defb "E"+#80
A056 4C4F41     1460         defm "LOA"
A059 C4         1470         defb "D"+#80
A05A 444F       1480         defm "DO"
A05C D3         1490         defb "S"+#80
A05D 00         1500         defb 0
A05E DF         1510 tape:   rst  #18
A05F 75A0       1520         defw ams1
A061 C9         1530         ret
A062 DF         1540 tapein: rst  #18
A063 78A0       1550         defw ams2
A065 C9         1560         ret
A066 DF         1570 tapout: rst  #18
A067 7BA0       1580         defw ams3
A069 C9         1590         ret
A06A DD7E00     1600 user:   ld   a,(ix)
A06D DD2A75BE   1610         ld   ix,(#be75)
A071 DD7705     1620         ld   (ix+5),a
A074 C9         1630         ret
A075 FDCC       1640 ams1:   defw #ccfd
A077 07         1650         defb 7
A078 01CD       1660 ams2:   defw #cd01
A07A 07         1670         defb 7
A07B 18CD       1680 ams3:   defw #cd18
A07D 07         1690         defb 7
A07E E4CD       1700 ams4:   defw #cde4
```

```
A080 07         1710         defb 7
A081 FE01       1720 drive:  cp   1
A083 201B       1730         jr   nz,err
A085 DD6601     1740         ld   h,(ix+1)
A088 DD6E00     1750         ld   l,(ix+0)
A08B 23         1760         inc  hl
A08C 5E         1770         ld   e,(hl)
A08D 23         1780         inc  hl
A08E 56         1790         ld   d,(hl)
A08F 1A         1800         ld   a,(de)
A090 FE41       1810         cp   "A"
A092 2827       1820         jr   z,drivea
A094 FE61       1830         cp   "a"
A096 2823       1840         jr   z,drivea
A098 FE42       1850         cp   "B"
A09A 2828       1860         jr   z,driveb
A09C FE62       1870         cp   "b"
A09E 2824       1880         jr   z,driveb
A0A0 21ADA0     1890 err:    ld   hl,bad
A0A3 7E         1900 print:  ld   a,(hl)
A0A4 FE00       1910         cp   0
A0A6 C8         1920         ret  z
A0A7 CD5ABB     1930         call #bb5a
A0AA 23         1940         inc  hl
A0AB 18F6       1950         jr   print
A0AD 426164     1960 bad:    defm "Bad"
A0B0 20         1970         defb 32
A0B1 636F6D6D   1980         defm "command"
A0B8 0D0A00     1990         defb 13,10,0
A0BB DD2A75BE   2000 drivea: ld   ix,(#be75)
A0BF DD360600   2010         ld   (ix+6),0
A0C3 C9         2020         ret
A0C4 DD2A75BE   2030 driveb: ld   ix,(#be75)
A0C8 DD360601   2040         ld   (ix+6),1
A0CC C9         2050         ret
A0CD FE02       2060 write:  cp   2
A0CF 20CF       2070         jr   nz,err
A0D1 DD7E03     2080         ld   a,(ix+3)
A0D4 67         2090         ld   h,a
A0D5 DD7E02     2100         ld   a,(ix+2)
A0D8 6F         2110         ld   l,a
A0D9 DD7E01     2120         ld   a,(ix+1)
A0DC 57         2130         ld   d,a
A0DD DD7E00     2140         ld   a,(ix)
A0E0 5F         2150         ld   e,a
A0E1 E5         2160         push hl
A0E2 D5         2170         push de
A0E3 21F3A0     2180         ld   hl,satext
A0E6 CDA3A0     2190         call print
A0E9 D1         2200         pop  de
A0EA E1         2210         pop  hl
A0EB 3E00       2220         ld   a,0
A0ED CD9EBC     2230         call #bc9e
A0F0 305D       2240         jr   nc,wrong
A0F2 C9         2250         ret
A0F3 53617669   2260 satext: defm "Saving Headerless File"
A109 0D0A00     2270         defb 13,10,0
A10C FE02       2280 read:   cp   2
A10E 2090       2290         jr   nz,err
A110 DD7E03     2300         ld   a,(ix+3)
A113 67         2310         ld   h,a
A114 DD7E02     2320         ld   a,(ix+2)
A117 6F         2330         ld   l,a
A118 DD7E01     2340         ld   a,(ix+1)
A11B 57         2350         ld   d,a
A11C DD7E00     2360         ld   a,(ix)
A11F 5F         2370         ld   e,a
A120 E5         2380         push hl
A121 D5         2390         push de
A122 2135A1     2400         ld   hl,lotext
A125 CDA3A0     2410         call print
A128 CD06BB     2420         call #bb06
A12B D1         2430         pop  de
A12C E1         2440         pop  hl
A12D 3E00       2450         ld   a,0
A12F CDA1BC     2460         call #bca1
A132 301B       2470         jr   nc,wrong
A134 C9         2480         ret
A135 4C6F6164   2490 lotext: defm "Loading Headerless File"
A14C 0A0D00     2500         defb 10,13,0
A14F 2156A1     2510 wrong:  ld   hl,wrtext
A152 CDA3A0     2520         call print
A155 C9         2530         ret
A156 48656164   2540 wrtext: defm "Headerless Error"
A166 0A0D00     2550         defb 10,13,0
A169 FE02       2560 load:   cp   2
A16B C2A0A0     2570         jp   nz,err
A16E DD6603     2580         ld   h,(ix+3)
A171 DD6E02     2590         ld   l,(ix+2)
A174 46         2600         ld   b,(hl)
A175 23         2610         inc  hl
A176 5E         2620         ld   e,(hl)
A177 23         2630         inc  hl
A178 56         2640         ld   d,(hl)
A179 EB         2650         ex   de,hl
A17A DD5601     2660         ld   d,(ix+1)
A17D DD5E00     2670         ld   e,(ix)
A180 D5         2680         push de
A181 CD77BC     2690         call #bc77
A184 E1         2700         pop  hl
A185 CD83BC     2710         call #bc83
A188 C9         2720         ret
A189 21BFA1     2730 dos:    ld   hl,dostxt
A18C C3A3A0     2740         jp   print
A18F            2750 dostxt:
A18F 04020A18   2760         defb 4,2,10,24
A193 20202020   2770         defm "                Extended F1-X Dos "
A1B4 20202863   2780         defm "  (c) 1987 by Hans-Georg Esser"
A1D2 20202020   2790         defm "                "
A1E3 180D0A0A   2800         defb 24,13,10,10
A1E7 20466F6C   2810         defm " Folgende AMSDOS-Befehle wurden in das"
A20D 2056444F   2820         defm " VDOS eingebunden:"
A21F 0D0A0A     2830         defb 13,10,10
A222 207C5441   2840         defm " |TAPE, |TAPE.IN, |TAPE.OUT,"
A23E 207C5553   2850         defm " |USER,x ; |DRIVE,"
A250 2278220D   2860         defb 34,"x",34,13,10,10
A256 207A7573   2870         defm " zusaetzliche Befehle:"
A26C 0D0A0A     2880         defb 13,10,10
A26F 207C5245   2890         defm " |READ, start,laenge      --> "
A28C 206C6573   2900         defm " lesen eines headerlosen Files von Cas
A2B7 0D0A       2910         defb 13,10
A2B9 207C5752   2920         defm " |WRITE,start,laenge      --> "
A2D6 20736368   2930         defm " schreiben eines headerlosen Files auf Cas
te"
A305 0D0A0A     2940         defb 13,10,10
A308 207C4C4F   2950         defm " |LOAD,"
A30F 22         2960         defb 34
A310 46494C45   2970         defm "FILENAME.EXT"
A31C 22         2980         defb 34
A31D 2C616472   2990         defm ",adresse"
A325 0D0A0A     3000         defb 13,10,10
A328 207C444F   3010         defm " |DOS                    --> "
A345 20496E66   3020         defm " Informationen abrufen"
A35B 0D0A0A0A   3030         defb 13,10,10,10,10
A360 00         3040 ende:   nop
```

# Drucken mit dem 8. Bit

Dieser Beitrag wendet sich an die stolzen Besitzer des CPC 464 und der RAM-Erweiterung der Firma Vortex, die nicht auf eine 8-Bit Druckausgabe verzichten wollen, um z.B. Grafiken von Turbo-Pascal unter dem 62k-CP/M zu erstellen. Als Voraussetzung hierfür ist der Eingriff in den Rechner laut Heft 5'86 CPC-International, der das 8. Bit unter Verwendung des Ports C Bit 5 des 8255 (Datenausgabe an Rekorder) bereitstellt, zu sehen. Zur Ansteuerung dieser neu geschaffenen Schnittstelle sind schon viele Programme veröffentlicht worden, die sich aber alle an die »Nur-Basic-Programmierer« richten. Was macht aber der »62k-CP/M-ler"? Der muß, ob er will oder nicht, in die Tiefen des BDOS und der Bankumschaltung hinabtauchen, um zu erforschen, wo er seine Routine zur erweiterten Druckerausgabe zu lassen hat. Dieser Beitrag soll Ihnen das Tieftauchen ersparen (vielleicht gibt es ja auch Nichtschwimmer unter den 62k-CP/M-lern), und einen Einblick in die Bankumschaltung geben.

Alle Leser, zu denen sich noch nicht die RAM-Aufteilung unter dem 62k-CP/M herumgesprochen hat, sollten sich folgendes Schema näher betrachten, (s. Skizze 1–3) welches den Speicher Ihres Rechners zeigt:

Dem aufmerksamen Leser wird auffallen, daß sich der RAM, wie er sich den Programmen unter 62k-CP/M darstellt, in verschiedenen Speicherbänken befindet. BIOS 1, das ist die CP/M Zero-Page, liegt in der Grundbank, dem eingebauten Speicher, genauso wie der untere Teil des TPA. Der obere Teil des TPA und ein weiterer Teil des BIOS liegt in der RAM-Erweiterung auf Bank 1. Durch die Elektronik der Bankumschaltung ist gewährleistet, daß ein CP/M-Programm diese beiden Speicherbereiche als durchgängigen RAM sieht. Der obere Bereich der Grundbank, mit BIOS 2 gekennzeichnet, beinhaltet im wesentlichen die allseits beliebten und nützlichen Sprungleisten und Routinen der Firmware, und ist ansonsten leer. Hier ist viel Platz für eigene Routinen, auch für die erweiterte Druckerausgabe (von 8600 bis A7FF) die den Firmware-Einsprung MC-PRINT-CHAR patched. Dieser Einsprung liegt bei BD2B der Grundbank. Wie bekommt man nun ein Programm in diesen Bereich? – Ganz einfach: man lädt es in den unteren TPA, schaltet den oberen Block des Speichers auf die Grundbank und verschiebt das Programm an die vorgesehene freie Stelle. Dann schaltet man den oberen Block zurück auf Bank 1 und springt WBOOT an. Fertig ist die auf 8 Bit erweiterte Druckerausgabe-Routine. Was man dazu kennen muß, sind freilich die Befehle zur Bankumschaltung. Derer gibt es jedoch zweierlei: Zum einen besitzt die Speichererweiterung ein Bank-Select-Register, was wir hier nicht brauchen, da für das 62k-CP/M immer nur die Bank 1 der Erweiterung aktiv ist, zum anderen werden Bit 5 und Bit 6 des Ports 7Fxx, der auch das Gate-Array anspricht, zum Blockselect und Mapping-Enable benutzt. Dabei bedeuten:

BIT 5: 1 = Schalte Grundbank ab und blende Erweiterung ein
0 = Schalte Erweiterung ab und blende Grundbank ein

BIT 6: 1 = Wähle oberen Block der Erweiterung an 0 = Wähle unteren Block der Erweiterung an

Was wir hier brauchen ist nur das 5. Bit des Ports 7Fxx, da der obere Block der Speichererweiterung immer angewählt bleiben kann, es sei denn, man möchte im Drucker-Spooler irgend etwas manipulieren (natürlich nur von einem Programm aus, was oberhalb 8000 im RAM liegt, da es sich sonst selbst den Teppich unter den Füßen wegzieht).

Da jetzt die grundlegenden Dinge bekannt sind, kann man sich getrost dem Listing zuwenden. Das Programm heißt CENTRONI.COM, wurde mit dem Wordstar geschrieben und danach durch den Makroassembler M80 und den Linker L80 genudelt. Sie können es auch einfach mit dem DDTZ eingeben, da es nicht so umfangreich ist. Wenn Sie das tun, brauchen Sie natürlich nicht auf die Label »Anfang« und »Done« bzw. »Ende« zu achten. Sie tippen einfach die Mnemonics ein und überall, wo im Op-Code ein Label steht, schauen Sie in der Spalte links nach und geben die entsprechenden Hex-Werte ein.

Wie Sie sicher schon erraten haben, sind eigentlich 2 Programme nötig. Einmal die Drucker-Routine selber und als zweites ein Ladeprogramm, welches die Drucker-Routine in die Grundbank schiebt und den Restart bei BD2B durch einen Sprung auf die Routine ersetzt.

Das Ladeprogramm beginnt bei »Start«. Als erstes wird durch DI der Interrupt gesperrt, was man bei Bankumschaltungen tunlichst immer machen sollte, da sich sonst der Rechner diskret verabschiedet. Danach wird die Bankumschaltung durch das Laden von BC mit 7FEE (7F = Portadresse, EE = »gängiger« Wert, der eigentlich alles so läßt wie es ist) vorbereitet und das Bit 5 in C zurückgesetzt (Sie erinnern sich: BIT 5 = 0 schaltet die Erweiterung ab und blendet die Grundbank ein). OUT (C),C vollbringt die Bankumschaltung. Die nächsten Befehle laden an die Stelle BD2B der Firmware-Sprungleiste den Sprung zur neuen Drucker-Routine, hier nach A400. Sie können sich auch einen anderen Speicherbereich aussuchen, der frei ist, müssen dann aber auch die Call-Adressen in der Drucker-Routine entsprechend ändern. Falls Sie mit dem Assembler arbeiten, ändern Sie nur den Wert in der Konstanten »Ziel« und hinter der Assembler-Direktive .Phase. Im Anschluß daran werden in die Register HL die Quelle, in DE das Ziel und in BC die Anzahl der zu verschiebenden Bytes geladen und mit LDIR das Verschieben des Programms realisiert. Zum glücklichen Ende gehört noch die Einblendung der Bank 1 der Erweiterung, das Zulassen des Interrupts und ein Jump nach 0000.

Die Drucker-Routine tippen Sie direkt im Anschluß an das Ladeprogramm ein. Sie rettet zuerst BC und setzt das neue 7. Bit zu 0 (mit D7LOW). Dann wird untersucht, ob das auszugebende Zeichen das 7. Bit gesetzt hat (also >127 ist). Wenn nein wird es einfach ausgegeben (MCPRCA), ansonsten wird in XPRI1 das 7. Bit gesetzt, das Zeichen ausgegeben und das 7. Bit wieder rück- gesetzt. Zum Schluß wird BC restauriert und die Routine mit RET verlassen. Die beiden Routinen D7LOW und D7HIGH laden erst die Portadresse F6 (Port C des 8255) nach B und lesen den Port aus. Dann wird das Bit 5 entweder rückgesetzt (LOW) oder gesetzt (HIGH) und wieder ausgegeben.

Für alle M80 Besitzer sei gesagt, daß der Befehl ».PHASE Adresse« den Assembler anweist, den Code für den Speicher

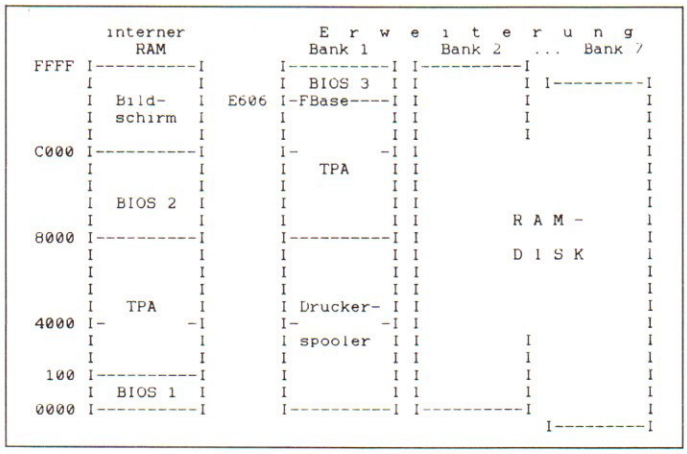



*Abb. 1*

ab Adresse zu erzeugen (hier A400), aber direkt hinter das Ladeprogramm abzulegen. Der Befehl ».DEPHASE« macht diese Direktive rückgängig. Außerdem braucht der M80 die Label »Anfang« und »Start« für die Berechnung der Anzahl Byte, die umgeladen werden sollen. Nach dem Linken steht Ihnen die Datei CENTRONI.COM zur Verfügung.

Diejenigen unter Ihnen, die das Programm mit dem DDT oder DDTZ eingegeben haben, gehen mit CTRL C ins Betriebssystem und speichern mit »Save 1 Centroni.com« das Programm ab.

Zu bemerken ist, daß das Programm nicht mit dem Druckerspooler der RAM-Erweiterung zusammenarbeitet. Schalten Sie also den Spooler ab (mit der Datei »SPOOL.COM«).

Da wir gerade dabei sind in den Speicherbänken rumzuwühlen, können wir auch noch ein anderes Problem behandeln. Haben Sie schon mal versucht von Turbo-Pascal unter 62k-CP/M einige Firmware-Routinen anzuspringen (mittels INLINE-Statements)? -Ja werden Sie sagen, das geht doch tadellos. Tut es auch, solange Ihr Programm den freien Speicher nicht bis zum Anschlag (FBASE) ausnutzt. Falls doch (vielleicht mittels riesiger Arrays) können Sie nach Aufruf von Firmware-Routinen den Rechner nur noch ausschalten oder wenigstens ein System-Reset durchführen, da der Prozessor ins Bodenlose abgetaucht ist. Der Grund dafür ist, daß Ihre Arrays eine Sprungleiste in der Bank 1, die parallel zur Firmware-Sprungleiste der Grundbank liegt, also mitten im TPA, überschrieben und damit zerstört haben. Sehen Sie sich die Sprungleiste im TPA in unzerstörtem Zustand an, sehen Sie nur lauter CALL F554. Alle Firmware-Calls gehen also auf eine Routine in BIOS. Wenn Sie also im Inline-Statement mit CALL BBF6 z.B. die Routine GRA-LINE-ABSOLUT aufrufen, so wird über BBF6: CALL F554 nach F554 gesprungen, wobei die nachfolgende Adresse, hier also BBF9 auf dem Stack liegt. Innerhalb der Routine F554 wird der Wert aber vom Stack geholt und nach 004F in die Zero-Page geladen, außerdem wird das L-Register mit 57 geladen. Danach folgt über die zentrale Bankumschalt-Routine F4FD gekoppelt mit ihrem Teil in der Zero-Page bei 004A der Einsprung in die Grundbank bei 8000. In der dortigen Routine wird das H-Register mit A8 geladen und der Interrupt-Vektor umgebogen, was hier aber nebensächlich ist. Mittels PUSH HL und RET erfolgt ein Sprung nach A857, was wiederum eine Sprungleiste ist. Von dort erfolgt der Einsprung in eine Routine bei 8118 die den Wert aus 4F holt. Dieser Wert stellt das ursprüngliche Sprungziel plus 3 in die Firmware dar. Dieser wird jetzt nun 3 mal dekrementiert und als Adresse angesprungen. Damit ist der Firmware-Aufruf bewerkstelligt. Falls Sie mit Ihren Arrays aber die Sprungleiste mit den CALL's F554 überschrieben haben, funktioniert das ganze nicht. Die Lösung des Problems liegt in dem direkten Ansprung von F554 und zwar nicht über ein CALL sondern ein JUMP, wobei vorher die zu CALLende Adresse+3 auf den Stack gePUSHd wird. Das sieht im Inline-Statement folgendermaßen aus:

```
PROCEDURE Call;
                                        (* in BC steht die zu rufende *)
BEGIN                                                  (* Adresse *)
INLINE ($03/$03/$03/
(* INC BC/INC BC/INC BC/ *)
                              $C5/                     (* PUSH BC *)
                        $C3/$54/$F5);                  (* JP F554 *)
END; (* Call *)
```

Über die Procedure Call laufen alle Firmware-Aufrufe. Die aufrufenden Proceduren übergeben im BC-Register das Sprungziel, also die Firmware-Adresse. Bei dem oben genannten Beispiel GRA-LINE-ABSOLUT = BBF6 sieht das so aus:

```
PROCEDURE GraLineAbsolute
(x,y: Integer);
BEGIN
INLINE ($2A/x/                                  (* LD HL,(x) *)
     $EB/                                       (* EX DE,HL *)
     $2A/y/                                     (* LD HL,(y) *)
     $01/$F6/$BB/                          (* LD BC,BBF6 *)
     $C3/Call);                                 (* JP Call *)
END;                                  (* GraLineAbsolute *)
```

Mit der Procedure Call sind sie jetzt in der Lage, alle Firmware-Routinen zu nutzen.
Wer auf den Geschmack gekommen ist, unter CP/M noch einige trickreiche Routinen, die die Firmware stark beanspruchen, zu basteln, soll sich doch mal an einer Hardcopy-Routine versuchen, die aus einem CP/M-Programm ohne weiteres aufgerufen werden kann (z.B. mit CTRL kleine ENTER-Taste). Man könnte diese Routine vielleicht in die Ticker-Chain (1/50 sec) einklinken .... usw. Viel Spaß beim Programmieren.

**Anleitung zum Test des Drucker-Programmes.**

* Voraussetzung ist ein CPC464 mit 8-Bit Centronics-Schnittstelle
* 62k-CP/M laden
* Druckerspooler abschalten, falls überhaupt initialisiert mittel SPOOL.COM
* Drucker einschalten, am besten auf Engschrift
* Datei TEST.COM aufrufen

– Diese Datei gibt den Zeichensatz von 20hex bis 7Fhex und von A0hex bis FEhex aus. Dabei erscheinen bei A0hex bis FEhex die gleichen Zeichen wie von 20hex bis 7Fhex.

* Jetzt CENTRONI.COM aufrufen
* Datei TEST.COM erneut aufrufen

– Diese Datei gibt den Zeichensatz von 20hex bis 7Fhex und von A0hex bis FEhex aus. Dabei erscheinen bei A0hex bis FEhex jetzt die Grafikzeichen. (L. Papenkort)



```
MACRO-80 3.44   Seite 1


                        Z80                             ;Z80-Code erzeugen
0000'                   ASEG                            ;absolute Lage
                        ORG     100H                    ;ab 100 hex
                ;
87F2            MCPRIN  EQU     07F2H+8000H     ;Adresse von MC PRINT CHAR
A400            ZIEL    EQU     0A400H                  ;dahin kommt die Drucker-Routine
B02B            MCPRCH  EQU     0BD2BH                  ;Firmware-Einsprung in Sprungleiste
                ;
0100  F3        START:  DI                              ;Interrupt sperren
0101  01 7FEE          LD      BC,7FEEH        ;Portadresse laden
0104  CB A9             RES     5,C                     ;Grundbank selektieren
0106  ED 49             OUT     (C),C                   ;und ausfuehren
0108  21 BD2B           LD      HL,MCPRCH       ;MC PRINT CHAR
010B  3E C3             LD      A,0C3H                  ;Code fuer JP
010D  77                LD      (HL),A                  ;und rein !
010E  23                INC     HL                      ;ein Byte weiter
010F  3E 00             LD      A,00                    ;Low-Byte Adresse
0111  77                LD      (HL),A                  ;und rein
0112  23                INC     HL                      ;ein Byte weiter
0113  3E A4             LD      A,0A4H                  ;High-Byte Adresse
0115  77                LD      (HL),A                  ;rein damit !
0116  21 012A           LD      HL,ANFANG       ;Anfang des Programms
0119  11 A400           LD      DE,ZIEL                 ;Verschiebeziel
011C  01 002C           LD      BC,DONE-ANFANG          ;Anzahl der Bytes
011F  ED B0             LDIR                            ;und verschieben
0121  01 7FEE           LD      BC,7FEEH        ;Port-Adresse wieder laden
0124  ED 49             OUT     (C),C                   ;und zurueck auf Bank 1
0126  FB                EI                              ;jetzt koennen wieder Interrupts komen
0127  C3 0000           JP      0000                    ;Warmstart
                ;
012A            ANFANG: DS      0                       ;nur fuer Adressberechnung
                ;
                        PHASE   0A400H                  ;Code fuer A400 erzeugen
                ;
A400  C5                PUSH    BC                      ;BC retten
A401  CD A41B           CALL    D7LOW                   ;D7 LOW setzen
A404  CB 7F             BIT     7,A                     ;ist BIT 7 HIGH ?
A406  20 05             JR      NZ,XPRI1        ;JA ---> setze HIGH
A408  CD A418           CALL    MCPRCA                  ;und ausgeben
A40B  C1                POP     BC                      ;BC zurueck
A40C  C9                RET                             ;ENDE
                ;
A40D  CD A423   XPRI1:  CALL    D7HIGH                  ;setze BIT 7 HIGH
A410  CD A418           CALL    MCPRCA                  ;ausgeben
A413  CD A41B           CALL    D7LOW                   ;setze BIT 7 LOW und ENDE
A416  C1                POP     BC                      ;BC restaurieren
A417  C9                RET                             ;ENDE
                ;
A418  CF        MCPRCA: RST     08
A419  87F2              DEFW    MCPRIN                  ;Ansprung von MC PRINT CHAR
                ;
A41B  06 F6     D7LOW:  LD      B,0F6H                  ;Port C 8255
A41D  ED 48             IN      C,(C)                   ;hole Port
A41F  CB A9             RES     5,C                     ;RES BIT 5
A421  18 06             JR      D7OUT                   ;Portausgabe
A423  06 F6     D7HIGH: LD      B,0F6H                  ;Port C 8255
A425  ED 48             IN      C,(C)                   ;hole Port
A427  CB E9             SET     5,C                     ;SET BIT 5
A429  ED 49     D7OUT:  OUT     (C),C                   ;Portausgabe
A42B  C9                RET                             ;und fertig
                ;
                        DEPHASE                         ;Phase zuruecksetzen
                ;
0156            DONE:   DS      0                       ;Label zur Adressberechnung
                        END     START                   ;Ende des Programms
```


## Weitere Sonderhefte im Verlag erhältlich




**Sonderheft 1/86 und 2/86:**
Software — Listings — Infos für alle Schneider CPC! Sonderheft 1 beinhaltet eine abwechslungsreiche Sammlung beliebter und nützlicher Programme aus den Sparten Anwendung, Spiel und Tips & Tricks.
Der große DFÜ-Sonderteil zeigt Ihnen alles Wissenswerte zur Datenfernübertragung auf und vermittelt Basiswissen. Insgesamt 28 aktuelle Listings — Software satt im CPC Sonderheft




**Sonderheft 3/86:**
Reviews — Spiele — Anwendungen — ein wahres Hit-Sammelsorium birgt das CPC-Sonderheft 3/86.
Die besten Spielprogramme im Überblick und viele Tips, Lösungen und Karten zu Computerspielen- und Abenteuern. Begeistern wird Sie auch der Flugsimulator — ein echter Leckerbissen zum Eintippen! Fantasy- und Adventurefreunde werden sich über das erste Rollenspieladventure Monstergarten sicherlich genauso freuen, wie die Hardware-Freunde über die Echtzeituhr zum Selbstbau. Des weiteren gibt es viele tolle Programme aus den Bereichen Spiel, Anwendung und Utilities sowie fundierte Berichte über die effektive Interruptprogrammierung. Da ist für jeden etwas dabei. Und natürlich: Alle Programme sind auch auf Diskette oder Kassette erhältlich!




**Sonderheft 4/87**
Programmiersprachen — Anwendungen in Turbo-Pascal und mannigfaltige Informationen stehen im Mittelpunkt des 4. CPC Sonderheftes. Mit über 200 Seiten praller CPC-Informationen, Tips und wertvollen Programme das ideale Sammelsurium für jeden CPC-Anwender. Interessiert? — dann sollten Sie sich schnell entscheiden, denn es sind nur noch wenige Restposten verfügbar.

**Richten Sie Ihre Bestellungen an:**
**DMV-Verlag, Fuldaer Str. 6, 3440 Eschwege**

# EDIT-LIST

**Das vorliegende MC-Programm simuliert die drei BASIC-Befehle EDIT., LIST. und LIST,−, welche auf Rechnern, wie z. B. dem EGC Colour-Genie, vorhanden sind.**

Diese Befehle veranlassen den Rechner entweder, von ihm als fehlerhaft erkannte Programmzeilen im EDIT-Modus auf den Bildschirm zu bringen, oder die Fehlerzeile bzw. den Programmbereich ab der Fehlerzeile zu listen,und das alles ohne Eingabe der Zeilennummer. Diese Befehle erhöhen den Komfort der Fehlerbehandlung doch beträchtlich, da das ausgesprochen lästige Eintippen von Zeilennummern nahezu vollkommen entfällt.

Bei der Konzeption des Programms habe ich folgende Anforderungen gestellt:

1. Auf das doch etwas umständliche Einrichten von RSX-Befehlen soll verzichtet werden.
2. Der Rechner soll die Befehle bei Auftreten eines Fehlers nicht automatisch ausführen, sondern sie sollten dem Anwender per Tastendruck zur Verfügung stehen; nicht jede als fehlerhaft erkannte Zeile enthält nämlich auch den Fehler, so daß in diesen Fällen ein automatisches Editieren nicht sonderlich sinnvoll ist.
3. Nach Änderung einiger ROM-Adressen sollte das Programm auch auf dem CPC 664/6128 lauffähig sein.

Um dem Rechnern, die den genannten Befehlen entsprechenden Eigenschaften zu verleihen, müssen folgende Dinge bekannt sein (mit den in Klammern angegebenen Adressen für den 664/6128 dürfte das Programm auch für diese Computer zu verwenden sein):

1. Die Adresse der Fehlerzeile. Sie liegt im Vektor &ADA6 (&AD8C/&AD8C).
2. Die ROM-Routine »Zeilennummer holen«. Sie befindet sich an der Adresse &DDD9 (&DEBD/&DEB8) und lädt in Kenntnis der Zeilenadresse die Zeilennummer ins Register HL.
3. Die ROM-Routine »Programmbereichlisten« beginnt bei &E10D (&E1E8/&E1E3). Die Register BC und DE müssen die Start- bzw. Endzeilennummer enthalten.
4. Die Eingabeschleife, in die der Rechner immer nach Durchführung der ihm aufgetragenen Befehle springt (z.B., um sich mit READY zu melden), beginnt bei &C064 (&C058/&C058). Auch bei Auftreten eines Fehlers läuft er nach vorheriger Ausgabe der Fehlermeldung in die Eingabeschleife. Erst dort wird entschieden, ob es sich bei dem aufgetretenen Fehler um einen SYNTAX-ERROR handelt, d.h., ob der Inhalt der Adresse &ADAA (&AD90/&AD90), welche die Fehlernummer enthält, gleich &02 ist, also gleich der Nummer für den SYNTAX-ERROR. Trifft dies zu, so wird die Fehlerzeile im EDIT-Modus ausgegeben, sofern sich der Rechner im Programmodus befindet. Auch darüber wird in der Eingabeschleife entschieden, und zwar nur dann, wenn ein SYNTAX-ERROR vorliegt (aus diesem Grunde erscheinen Programmzeilen mit anderen Fehlern normalerweise auch nicht im EDIT-Modus auf dem Bildschirm). Ob nun der Programmodus vorliegt, wird in diesem Falle anhand des Inhalts der Adresse &ADAA (&AD90/&AD90; Adresse der Fehlerzeile) geklärt. Ist der Wert größer Null, so ist aus der Sicht des Rechners der Programmodus eingeschaltet.

Mit diesem Wissen ist es leicht, den Rechner davon zu überzeugen, daß jede Fehlerzeile auf dem Bildschirm zu erscheinen hat. Da bei Auftreten eines Fehlers während des Programmablaufs der Inhalt von &ADAA immer von Null verschieden ist, muß von dem Maschinenprogramm aus die Speicherzelle &ADA6 (&AD8C/&AD8C) immer nur mit dem Wert &02 des SYNTAX-ERRORs belegt werden. Auf diese Weise »sieht« der Rechner immer einen SYNTAX-ERROR und muß, da er sich immer im Programmodus glaubt, jede Fehlerzeile editieren.

Nach diesen Ausführungen muß zum Assemblerlistung m.E. nur noch soviel gesagt werden, daß das Freischalten des oberen ROMs (für alle Rechner gleich: CALL &B900) eine absolute Notwendigkeit ist, da sich der Rechner ansonsten im RAM verfängt und abstürzt. Im übrigen kann das MC-Programm durch eine etwas effektivere Programmierung durchaus um einige Bytes kürzer gehalten werden, aus Gründen der Übersichtlichkeit habe ich aber darauf verzichtet.

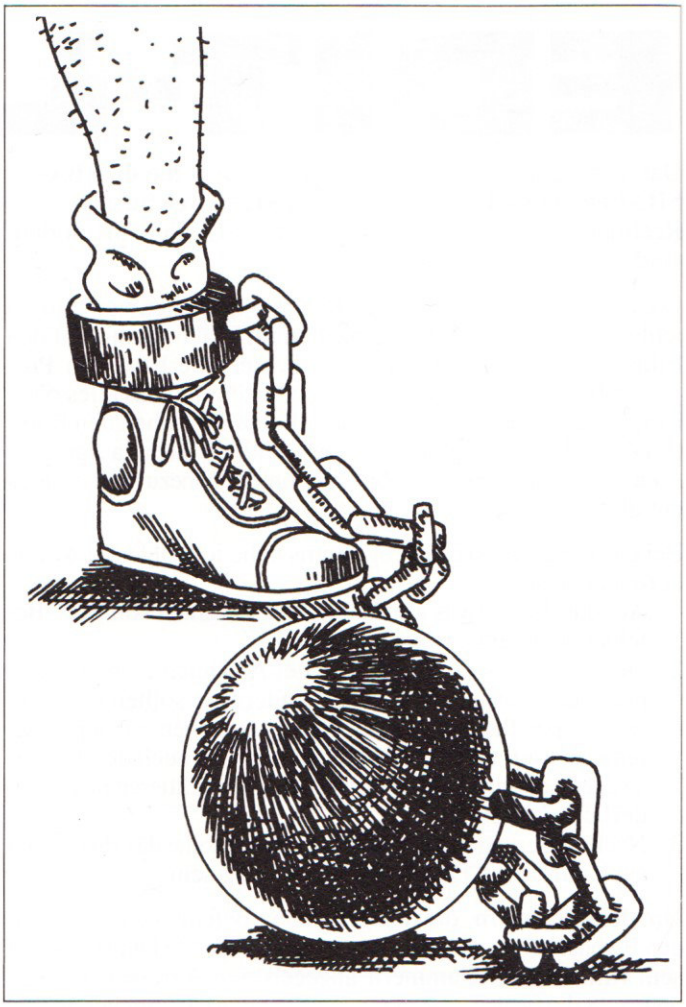
Wer den Basic-Lader benutzt, möge bitte die übliche Reihenfolge Abtippen − Abspeichern − Ausprobieren einhalten. Soll das Programm an einer anderen Stelle als &A000 installiert werden, sind die mit dem Label LIST1 verbundenen Adressen entsprechend zu ändern, ebenso das Memory und die CALLs bei der Tastenbelegung. Die Bedienung des Programms ist sehr einfach, nach dem Laden mit RUN"stehen die Befehle auf den Zahlen 7, 8 und 9 des Tastenblocks zur Verfügung, und zwar in Form von z.B. KEY 135,"CALL &A000"+CHR$(13) für "LIST.". Ein Umdefinieren ist natürlich jedem freigestellt.

Wichtiger Hinweis: Wird vor Benutzen der Befehle im Direktmodus ein Fehler verursacht, werden die Adressen &ADAA und &ADA6 auf Null gesetzt. Die Befehle verlieren dann ihre Wirkung solange, bis wieder ein Fehler im Programmodus auftritt.

(Michael Heyl)

## für 464

```
10 MEMORY &9FFF                                          [134]
20 FOR adr=&A000 TO &A02A:READ a$:POKE adr [3025]
,VAL("&"+a$):NEXT
30 DATA CD,0E,A0,EB,18,14,CD,0E                          [1251]
40 DATA A0,11,FF,FF,18,0C,CD,00                          [1339]
50 DATA B9,2A,A6,AD,CD,DC,DD,44                          [792]
60 DATA 4D,C9,CD,0D,E1,CD,64,C0                          [1751]
70 DATA CD,00,B9,3E,02,32,AA,AD                          [1457]
80 DATA CD,64,C0                                         [453]
90 KEY 135,"CALL &A000"+CHR$(13)                         [2206]
100 KEY 136,"CALL &A006"+CHR$(13)                        [2344]
110 KEY 137,"CALL &A020"+CHR$(13)                        [945]


****** EDIT-LIST-Routine    (C) Michael Heyl   1987 ******


A000             10        ORG  &A000        ;Start MC-Programm
A000  CD0EA0     20 LIST.  CALL LIST1        ;Startzeilennr. f0r LIST. holen
A003  EB         30        EX   DE,HL        ;Startzeilennr.=Endzeilennr.
A004  1814       40        JR   LIST2        ;nach LIST2:Listen
A006  CD0EA0     50 LIST.- CALL LIST1        ;Startzeilennr. f0r LIST.- holen
A009  11FFFF     60        LD   DE,&FFFF     ;Default-Endzeilennr.
A00C  180C       70        JR   LIST2        ;nach LIST2:Listen
A00E  CD00B9     80 LIST1  CALL &B900         ;oberes ROM freischalten
A011  2AA6AD     90        LD   HL,(&ADA6)   ;Zeilenadr. nach HL
A014  CDD9DD     100       CALL &DDD9        ;Zeilennr. holen (nach HL)
A017  44         110       LD   B,H          ;als Startzeilennr. ....
A018  4D         120       LD   C,L          ;..... nach BC
A019  C9         130       RET               ;zur0ck zu LIST. oder LIST.-
A01A  CD0DE1     140 LIST2 CALL &E10D         ;Listen
A01D  CD64C0     150       CALL &C064        ;zur Eingabeschleife
A020  CD00B9     160 EDIT  CALL &B900         ;oberes ROM freischalten
A023  3E02       170       LD   A,&02        ;Nr. f0r SYNTAX-ERROR nach A ....
A025  32AAAD     180       LD   (&ADAA),A    ;.... und nach &ADAA
A028  CD64C0     190       CALL &C064        ;zur Eingabeschleife

Programm:EDIT-LIST   Start:&A000   Ende:&A02A   LÄnge:&002B

Labeltabelle:
A000 = LIST.     A006 = LIST.-
A00E = LIST1     A01A = LIST2
A020 = EDIT
```



# Super Trace

## Programm-Beschreibung

**Das Programm »Super Trace« dient zum Verlangsamen des Programmablaufes (von BASIC- und MC-Programmen).**

Diese hat den Vorteil, daß alle Abläufe zum Teil zeitlupenartig dargestellt werden und somit der Programmfluß in Ruhe überprüft werden kann. Somit können Fehler eventuell schneller erkannt werden und außerdem werden Details sichtbar, die bei normaler Geschwindigkeit kaum auffallen würden. Das Programm selbst ist in Assembler geschrieben und nützt die Interrupt-Möglichkeiten des CPC dazu aus, diesen beliebig in seiner Geschwindigkeit zu beeinflussen.

Die Geschwindigkeit läßt sich über den RSX-Befehl; SPEED einstellen. Zusätzlich gibt es je einen Befehl zum Ein- bzw. Ausschalten der Erweiterung (;TRACE. ON und ;TRACE. OFF). Damit auch bei extrem starker Verlangsamung ein Programm beendet werden kann, bevor dieses zu Ende ist, besteht die Möglichkeit, die Erweiterung durch Drücken der Taste TAB wieder auf normale Arbeitsgeschwindigkeit zu setzen. Dabei wird die Verzögerung auf 1 gesetzt, sie muß jedoch noch separat ausgeschaltet werden.

Bei einer Verzögerung von z. B. 410 läßt sich auch der Aufbau der Zeichen am Bildschirm in Zeitlupe erkennen. Wenn Sie diesen Wert setzen, ist beim Listen eines Programmes deutlich zu erkennen, wie jedes Zeichen Byte für Byte von oben nach unten erstellt wird.

Das Programm ist mittels des BASIC-Ladeprogrammes im Speicher frei verschiebbar und läuft auf allen drei CPC. Allerdings darf es nur in die zentralen 32k geladen werden, da ansonsten die Interrupt-Steuerung sowie die RSX-Einbindung nicht korrekt arbeitet.

**Befehlsliste zu Super Trace**

1. TRACE. ON (keine Parameter). Hiermit schalten Sie die Erweiterung ein.

2. SPEED, Verzögerung. Mit diesem Befehl können Sie die Arbeitsgeschwindigkeit Ihres CPC bestimmen. Dabei erreichen Sie die höchste Geschwindigkeit mit dem Wert 1. Die sinnvoll anzuwendende kleinste Geschwindigkeit liegt bei Wert 410.

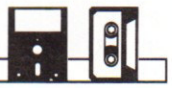
ACHTUNG:
Ein Wert von 0 bedeutet eine Verzögerung von 65535. Diesen Wert sollten Sie nicht verwenden.

WICHTIG:
Sie können die Geschwindigkeit jederzeit durch Drücken der TAB-Taste auf 1 setzen.

3. TRACE. OFF (keine Parameter). Der Befehl dient zum Ausschalten der Erweiterung.

(Klaus Kremer)

## für 464-664-6128



```
10 '------------------------------------ [2143]
-------
20 '         // S U P E R  T R A C E //    [356]
30 '------------------------------------ [2143]
-------
40 '                                       [117]
50 '** fuer Schneider/Amstrad CPC 464/664/ [3073]
6128 **
60 '                                       [117]
70 '       Copyright (c) 1987 by K.Kremer  [1029]
80 '                                       [117]
90 '         Programm-Stand: 17.06.1987    [2395]
100 '                                      [117]
110 '----------------------------------- [2143]
--------
120 '                                      [117]
130 INPUT"Ab welcher Speicherstelle soll S [5531]
uper Trace geladen werden";a
140 dif=a-32768                            [389]
150 '                                      [117]
160 MEMORY a-1:FOR i=a TO a+171:READ x%:PO [2760]
KE i,x%:NEXT
170 '                                      [117]
180 DATA &00,&01,&14,&80,&21,&10,&80,&CD,& [1695]
D1,&BC,&3E
190 DATA &C9,&32,&00,&80,&C9,&00,&00,&00,& [2921]
00,&1F,&80
200 DATA &C3,&92,&80,&C3,&9E,&80,&C3,&69,& [1338]
80,&54,&52
210 DATA &41,&43,&45,&2E,&4F,&CE,&53,&50,& [2200]
45,&45,&C4
220 DATA &54,&52,&41,&43,&45,&2E,&4F,&46,& [1714]
C6,&00,&AF
230 DATA &32,&5B,&80,&32,&64,&80,&21,&57,& [2418]
80,&06,&81
240 DATA &0E,&FC,&11,&75,&80,&CD,&E0,&BC,& [2651]
06,&81,&0E
250 DATA &FC,&11,&81,&80,&21,&60,&80,&C3,& [1907]
D7,&BC,&00
260 DATA &00,&00,&00,&00,&81,&75,&80,&00,& [1814]
00,&00,&59
270 DATA &80,&00,&81,&81,&80,&00,&21,&57,& [2018]
80,&CD,&E6
280 DATA &BC,&21,&60,&80,&C3,&DD,&BC,&2A,& [2210]
90,&80,&AF
290 DATA &2B,&BD,&20,&FC,&BC,&20,&F9,&C9,& [2139]
3E,&44,&CD
300 DATA &1E,&BB,&28,&ED,&21,&01,&00,&22,& [1664]
90,&80,&C9
310 DATA &00,&01,&00,&3A,&8F,&80,&FE,&01,& [1819]
C8,&3C,&32
320 DATA &8F,&80,&18,&98,&FE,&01,&C0,&DD,& [2190]
6E,&00,&DD
330 DATA &66,&01,&22,&90,&80,&C9,&00,&00,& [1873]
00,&00,&00
340 '                                      [117]
350 DATA 32770,32773,32781,32788,32791,327 [2516]
94,32797,32824
360 DATA 32827,32830,32837,32847,32850,328 [2033]
61,32870,32874
370 DATA 32880,32886,32908,32915,32922,329 [1945]
36
380 '                                      [117]
390 RESTORE 350                            [741]
400 FOR i%=1 TO 22:READ a2:a2=a2+dif:wert= [9019]
(PEEK(a2)+PEEK(a2+1)*256):wert=wert+dif:PO
KE a2,wert-FIX(wert/256)*256:POKE a2+1,FIX
(wert/256):NEXT
410 '                                      [117]
420 SAVE"trace",b,a%,172                   [1071]
430 PRINT"Initalisierung durch 'CALL";a;"' [2901]
!"
440 '                                      [117]
450 END                                    [110]
```

```
 10 ;------------------------------
 30 ;   // S U P E R  T R A C E //
 50 ;------------------------------
 70 ;
 90 ;COPYRIGHT (c) 1987 by Klaus Kremer
170 ;
190 ;-------------------------------
```

```
210 ;
230         ENT  *
250 ;
270         ORG  #8000
290 ;
310 ;RSX-Befehle in das System einbinden
330 ;
350 RELOC:  NOP
370         LD   BC,RSX              ;Adresse der RSX-Befehlstabelle
390         LD   HL,KERNAL           ;Adresse des 4-Byte-Speichers
410         CALL LOGEXT
430         LD   A,#C9               ;  nochmaligen Aufruf
450         LD   (RELOC),A           ;  durch einsetzen eines RET-
470         RET                      ;  Befehles unterbinden
490 ;
510 KERNAL: DEFS 4                   ;Speicher fuer Kernal
530 RSX:    DEFW NAMETABELLE         ;Adresse der Befehlsworte
550         JP   T ON                ;1
570         JP   SPEED               ;2
590         JP   T OFF               ;3
610 ;
630 NAMETA: DEFM "TRACE.O"
650         DEFB "N"+#80             ;1
670         DEFM "SPEE"
690         DEFB "D"+#80             ;2
710         DEFM "TRACE.OF"
730         DEFB "F"+#80             ;3
750         DEFB 0
770 ;
790 ;*****************************************
810 ;
830 LOGEXT: EQU  #BCD1               ;RSX-Erweiterung einbinden
850 NEWFTI: EQU  #BCE0               ;KL NEW FAST TICKER
870 DELFTI: EQU  #BCE6               ;KL DEL FAST TICKER
890 NEWFFL: EQU  #BCD7               ;KL NEW FRAME FLY
910 DELFFL: EQU  #BCDD               ;KL DEL FRAME FLY
930 TESTKE: EQU  #BB1E               ;KM TEST KEY
950 WAITCH: EQU  #BB06               ;KM WAIT CHAR
970 CHARRE: EQU  #BB0C               ;KM CHAR RETURN
990 ;
1010 ;*******************************************************
1030 ;
1050 ;Routine zum Einhaengen der Erweiterung
1070 ;
1090 INT:    XOR  A                  ;  Zaehler der beiden Bloecke loeschen,
1110         LD   (BZAEHLER),A       ;  indem sie auf 0 gesetzt
1130         LD   (B2ZAEHLER),A      ;  werden
1150         LD   HL,BLOCK1          ;Adresse des ersten Blockes
1170         LD   B,%10000001        ;Prioritaet der Interrupt-Routine
1190         LD   C,252              ;RAM/ROM-Status
1210         LD   DE,INT ROUTINE     ;Adresse der ersten Routine
1230         CALL NEWFTICKER         ;erste Routine einreihen
1250         LD   B,%10000001        ;Prioritaet der Interrupt-Routine
1270         LD   C,252              ;RAM/ROM-Status
1290         LD   DE,INT 2ROUTINE    ;Adresse der 2. Routine
1310         LD   HL,BLOCK2          ;Adresse des 2. Blockes
1330         JP   NEWFFLY            ;2. Routine einreihen
1350 ;
1370 BLOCK1: DEFB 0,0                ;Interrupt-Block 1
1390         DEFB 0,0
1410 BZAEHL: DEFB 0
1430         DEFB %10000001
1450 BL_AD:  DEFW INT_ROUTINE
1470         DEFB 0
1490 ;
1510 BLOCK2: DEFB 0,0                ;Interrupt-Block 2
1530         DEFB 0,0
1550 B2ZAEH: DEFB 0
1570         DEFB %10000001
1590         DEFW INT 2ROUTINE
1610         DEFB 0
1630 ;
1650 ;*******************************************************
1670 ;
1690 ;Befehl >T R A C E . O F F<
1710 ;
1730 T OFF:  LD   HL,BLOCK1          ;  beide Bloecke aus den
1750         CALL DELFTICKER         ;  Interrupt-Ketten
1770         LD   HL,BLOCK2          ;  ausblenden
1790         JP   DELFFLY
1810 ;
1830 ;*******************************************************
1850 ;
1870 ;INTERRUPT STEUERUNG ***SYSTEM-SCHLEIFE***
1890 ;
1910 ;
1930 INT RO: LD   HL,(VERZOEGERUNG)  ;Verzoegerung holen
1950         XOR  A                  ;A:=0
1970 ;
1990 LOOP1:  DEC  HL                 ;  Hl bis auf 0 herabzaehlen
2010         CP   L
2030         JR   NZ,LOOP1
2050         CP   H
2070         JR   NZ,LOOP1
2090 ;
2110         RET                     ;Fertig
2130 ;
2150 ;Tastatur-Abfrage
2170 ;
2190 INT 2R: LD   A,68               ;  TAB-Taste gedrueckt?
2210         CALL TESTKEY
2230         JR   Z,INT ROUTINE      ;NEIN?>weiter mit der Verzoegerungsr.
2250         LD   HL,1               ;  JA?>Verzoegerung auf 1
2270         LD   (VERZOEGERUNG),HL  ;  setzen
2290         RET                     ;  und fertig
2310 ;
2330 TONOFF: DEFS 1                  ;Speicher fuer EIN/AUS-Flag
2350 VERZOE: DEFB 1,0                ;Speicher fuer die Verzoegerung,
2370 ;Standard-Inhalt:1
2390 ;
2410 ;*******************************************************
2430 ;
2450 ;Befehl >T R A C E . O N<
2470 ;
2490 T ON:   LD   A,(TONOFF)         ;  Flag fuer EIN/AUS holen
2510         CP   1                  ;EIN?
2530         RET  Z                  ;JA?>Fertig
2550         INC  A                  ;  ansonsten Flag
2570         LD   (TONOFF),A         ;  auf "EIN" setzen und
2590         JR   INT                ;  System einschalten
2610 ;
2630 ;*******************************************************
2650 ;
2670 ;Befehl >S P E E D<
2690 ;
2710 SPEED:  CP   1                  ;1 Parameter?
2730         RET  NZ                 ;NEIN?> Routine beenden
2750         LD   L,(IX)             ;  Vergroesserung holen
2770         LD   H,(IX+1)
2790         LD   (VERZOEGERUNG),HL  ;  und merken
2810         RET                     ;das war schon alles
2830 ;
2850 ;ENDE ----------------------------------------------------
```

# LISTER

```
10 MODE 2
20 DIM a$(158):DEFINT a-z
30 ON ERROR GOTO 340
40 WHILE a$(i)<>"$"
50  i=i+1:READ a$(i)
60 WEND
70 anz=i-1
80 INPUT"Name des ASCII-Programms ";1$
90 PRINT"Listing auf <B>ildschirm oder <D>rucker ? ";
100 a$=UPPER$(INKEY$)
110 IF a$="B"THEN an$=CHR$(24):aus$=CHR$(24):win=0:GOTO 140
120 IF a$="D"THEN an$=CHR$(27)+"E":aus$=CHR$(27)+"F":win=8:GOTO
130 GOTO 100
140 PRINT a$
145 OPENIN 1$
150 WHILE NOT EOF
160  LINE INPUT#9,a$
170  FOR j=1 TO anz
180   k=1
190   wo=INSTR(k,a$,a$(j))
200   IF wo=0 THEN 290
210   IF MID$(a$,wo-1,1)>"=" THEN 290
220   x$=MID$(a$,wo+LEN(a$(j)),1)
230   IF x$>"=" THEN 290
240   IF x$="1" THEN 290
250   IF x$="$" THEN 290
260   a$=LEFT$(a$,wo-1)+an$+a$(j)+aus$+RIGHT$(a$,LEN(a$)-wo+1-LEN
270   k=wo+LEN(a$(j))
```

Ein besonderer Vorteil der CPC BASIC-Interpreter ist der, daß sämtliche Befehle beim Auflisten von Programmen in Großbuchstaben ausgegeben werden. Trotzdem wäre es nützlich, sie noch deutlicher hervorzuheben.

Diesem Zweck dient das vorliegende Programm.

Es sucht aus einem als ASCII-Datei abgespeicherten BASIC-Programm (SAVE »name«,a) sämtliche Befehle heraus und gibt die daraufhin geänderten Zeilen aus. Da das Programm recht langsam arbeitet, ist es hauptsächlich für Druckerbesitzer interessant, doch es ist auch eine Bildschirmausgabe möglich. Auf dem Bildschirm erscheinen die Befehle invertiert, auf dem Drucker werden sie durch Fettdruck hervorgehoben.

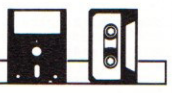
Die Druckercodes in Zeile 120 bedeuten:

ESC »E« = Fettdruck ein
ESC »F« = Fettdruck aus

und können bei Bedarf z.B. in Unterstreichen oder Doppeldruck geändert werden.

(Ralf Wegener/SH)

## für 464-664-6128



```
10 MODE 2                                   [513]
20 DIM a$(158):DEFINT a-z                   [997]
30 ON ERROR GOTO 340                        [1537]
40 WHILE a$(i)<>"@"                         [936]
50  i=i+1:READ a$(i)                        [1250]
60 WEND                                     [390]
70 anz=i-1                                  [577]
```

```
80 INPUT"Name des ASCII-Programms ";1$      [1964]
90 PRINT"Listing auf <B>ildschirm oder <D> [4698]
rucker ? ";
100 a$=UPPER$(INKEY$)                       [789]
110 IF a$="B"THEN an$=CHR$(24):aus$=CHR$(2 [2370]
4):win=0:GOTO 140
120 IF a$="D"THEN an$=CHR$(27)+"E":aus$=CH [3288]
R$(27)+"F":win=8:GOTO 140
130 GOTO 100                                [417]
140 PRINT a$                                [399]
145 OPENIN 1$                               [733]
150 WHILE NOT EOF                           [1840]
160  LINE INPUT#9,a$                       [1529]
170  FOR j=1 TO anz                         [1290]
180   k=1                                   [192]
190   wo=INSTR(k,a$,a$(j))                  [1038]
200   IF wo=0 THEN 290                      [675]
210   IF MID$(a$,wo-1,1)>"=" THEN 290       [2494]
220   x$=MID$(a$,wo+LEN(a$(j)),1)           [1722]
230   IF x$>"=" THEN 290                    [1982]
240   IF x$="1" THEN 290                    [1486]
250   IF x$="$" THEN 290                    [973]
260   a$=LEFT$(a$,wo-1)+an$+a$(j)+aus$+RIG [3063]
HT$(a$,LEN(a$)-wo+1-LEN(a$(j)))
270   k=wo+LEN(a$(j))                       [354]
280   IF k<LEN(a$) THEN 190                 [836]
290  NEXT                                   [350]
300  PRINT#win,a$                           [697]
310 WEND                                    [390]
320 CLOSEIN                                 [752]
330 END                                     [110]
340 IF ERR=15 THEN j=anz:RESUME 290         [979]
350 PRINT"Line"ERL:ERROR ERR:STOP           [2486]
360 DATA OUT,ABS,AFTER,AND,ASC,ATN,AUTO     [2460]
370 DATA BIN$,BORDER,BREAK                  [1124]
380 DATA CALL,CAT,CHAIN,CHR$,CINT,CLEAR,CL [3146]
G,CLOSEIN,CLOSEOUT,CLS,CONT,COS,CREAL
390 DATA DATA,DEC$,DEF,DEFINT,DEFREAL,DEFS [4917]
TR,DEG,DELETE,DI,DIM,DRAW,DRAWR
400 DATA EDIT,EI,ELSE,END,ENT,ENV,EOF,ERAS [3224]
E,ERL,ERR,ERROR,EVERY,EXP
410 DATA FIX,FN,FOR,FRE                     [630]
420 DATA GOSUB,GOTO                         [691]
430 DATA HEX$,HIMEM                         [1089]
440 DATA IF,INK,INKEY,INKEY$,INP,INPUT,INS [2097]
TR,INT
450 DATA JOY                                [115]
460 DATA KEY                                [91]
470 DATA LEFT$,LEN,LET,LINE,LIST,LOAD,LOCA [4125]
TE,LOG,LOG10,LOWER$
480 DATA MAX,MEMORY,MERGE,MID$,MIN,MOD,MOD [2070]
E,MOVE,MOVER
490 DATA NEXT,NEW,NOT                       [820]
500 DATA ON,OPENIN,OPENOUT,OR,ORIGIN        [1483]
510 DATA PAPER,PEEK,PEN,PI,PLOT,PLOTR,POKE [3941]
,POS,PRINT
520 DATA RAD,RANDOMIZE,READ,RELEASE,REM,RE [5478]
MAIN,RENUM,RESTORE,RESUME,RETURN,RIGHT$,RN
D,ROUND,RUN
530 DATA SAVE,SGN,SIN,SOUND,SPACE$,SPC,SPE [4771]
ED,SQ,SQR,STEP,STOP,STR$,STRING$,SWAP,SYMB
OL
540 DATA TAB,TAG,TAGOFF,TAN,TEST,TESTR,THE [4978]
N,TIME,TO,TROFF,TRON
550 DATA UNT,UPPER$,USING                   [1027]
560 DATA VAL,VPOS                           [402]
570 DATA WAIT,WEND,WHILE,WIDTH,WINDOW,WRIT [3311]
E
580 DATA XOR,XPOS                           [614]
590 DATA YPOS                               [481]
600 DATA ZONE                               [337]
610 DATA @                                  [215]
```

# Tonkopfjustage

**Dieses Programm dient dazu, den Tonkopf des CPC optimal auf den Stand des Tonkopfes einzustellen, mit dem das zu ladende Programm abgesaved wurde. (Im Klartext: Read error's zu vermeiden oder auszuschalten)**

Das Programm funktioniert wie folgt: Das Programm »Tonjust. dat« abtippen und starten. Es wird, nachdem das Programm überprüft hat ob alles in Ordnung ist, ein Binärfile (TONJUST. BIN) auf Diskette erzeugt. Das Programm wird folgendermaßen gestartet:

1. Memory setzen: Memory &6400-1
2. Load »Tonjust. Bin«
3. Call &6400

Die Kassette mit dem zu ladenden Programm einlegen. Mit Tastendruck >2< in das Hauptprogramm umschalten. Jetzt müssen Sie mit einem kleinen Schraubendreher die Stellschraube des Tonkopfes durch Drehen derselben auf Rechtsanschlag bringen; dann PLAY und irgendeine Taste drücken.

Nun baut sich der Bildschirm auf. Die grünen (mittleren) Linien kennzeichnen den bestmöglichen Einstellbereich für HEADERLOSE Programme (z.B. Spiele). Die roten Linien kennzeichnen den Einstellbereich für HEADER Programme. Der Kassettenmotor ist bereits eingeschaltet. Wenn Signale von der Kassette gelesen werden, dann werden diese auf dem Bildschirm in Form von weißen Linien dargestellt. Der Hintergrund verändert sich bei jedem neuen gelesenen Byte in die jeweilige Farbe. Wenn das zu sehr ablenkt, kann dies beliebig mit einem Druck auf die Taste »b« ein- oder ausschalten. Nun liegt es an Ihnen und Ihren Augen. Durch Drehen nach links oder rechts versucht man den bestmöglichen »Linienstand« (Tonkopfstand) zu finden.

Am besten ist die Aufnahmequalität, wenn die weißen Linien nicht mehr über die angegebenen Markierungen der jenachdem roten oder grünen Linien treten. Durch Druck auf ENTER gelangen Sie wieder in das Hauptmenü zurück. Jetzt haben Sie keine Probleme mehr mit dem Einladen fremder oder mit hoher Baudrate abgesavter Software. Wenn Sie Ihren Tonkopf eichen wollen, empfiehlt es sich die mitgelieferte Demokassette von Schneider (464) oder gekaufte Software (464, 664, 6128) zu benutzen.

(O. Geisen/CD)

## für 464-664-6128

```
10 '************************************* [1543]
20 '* TONJUST.DAT - DATA-Lader von 'CPC' * [2030]
30 '************************************* [1543]
40 '                                      [117]
50 DATA 6400,3E,01,01,1A,1A,CD,32,BC,1DFC  [2004]
60 DATA 6407,3E,02,01,03,03,CD,32,BC,1C64  [1965]
70 DATA 640F,3E,03,01,09,09,CD,32,BC,1CD4  [1212]
80 DATA 6417,01,00,F6,ED,49,3E,01,CD,12EF  [2409]
90 DATA 641F,0E,BC,21,21,65,CD,14,65,2E61  [1610]
100 DATA 6427,CD,06,BB,FE,32,28,15,FE,7E64 [2520]
110 DATA 642F,33,C8,FE,31,20,F2,CD,14,3416 [1786]
120 DATA 6437,BC,21,BF,65,CD,14,65,CD,41CF [2613]
130 DATA 643F,18,BB,18,D4,CD,14,BC,21,2BE1 [2159]
140 DATA 6447,04,66,CD,14,65,CD,18,BB,03F7 [1676]
150 DATA 644F,CD,14,BC,DD,21,75,64,DD,7919 [1669]
160 DATA 6457,7E,00,FE,00,28,24,DD,23,2089 [1662]
170 DATA 645F,DD,66,00,DD,23,DD,6E,00,7860 [2137]
180 DATA 6467,DD,23,54,5D,1C,01,4F,00,696A [2116]
190 DATA 646F,77,ED,B0,18,E2,0F,C9,E0,111E [1361]
200 DATA 6477,0F,ED,A0,FF,FC,10,FF,FB,2195 [2140]
210 DATA 647F,20,00,01,10,F6,ED,49,3E,1588 [1293]
220 DATA 6487,01,CD,DE,BB,2A,1F,65,11,2247 [1149]
230 DATA 648F,00,00,CD,C0,BB,11,00,00,103C [1505]
240 DATA 6497,D5,CD,DA,64,CD,F5,64,FE,414A [1204]
250 DATA 649F,00,20,05,21,00,00,18,0E,0A8E [1956]
260 DATA 64A7,6F,26,00,11,02,00,CD,BE,3E24 [1804]
270 DATA 64AF,BD,06,35,2B,10,FD,22,1F,583F [1439]
280 DATA 64B7,65,D1,D5,CD,F6,BB,D1,CD,1483 [2435]
290 DATA 64BF,09,BB,FE,0D,CA,18,64,FE,3356 [3048]
300 DATA 64C7,62,CC,03,65,13,7A,FE,02,04BE [1647]
310 DATA 64CF,20,C6,7B,FE,80,20,C1,C3,24C1 [1953]
320 DATA 64D7,50,64,06,F5,0E,00,16,FF,3E33 [2665]
330 DATA 64DF,AF,ED,60,CB,14,30,01,3C,6C2E [2159]
340 DATA 64E7,15,20,F6,FE,FF,20,02,3E,14E2 [2378]
350 DATA 64EF,00,32,1E,65,C9,F5,01,10,0C9E [2271]
360 DATA 64F7,7F,ED,49,E6,1F,F6,40,ED,00CD [2114]
370 DATA 64FF,79,F1,C9,3A,F5,64,FE,F5,1D71 [2957]
380 DATA 6507,28,04,3E,F5,18,02,3E,C9,1DED [1087]
390 DATA 650F,32,F5,64,C9,7E,FE,00,C8,2490 [1211]
400 DATA 6517,CD,5A,BB,23,18,F6,00,00,6648 [1567]
410 DATA 651F,00,0F,01,1F,06,02,18,54,024C [1729]
420 DATA 6527,20,4F,20,4E,20,4B,20,4F,0303 [2209]
430 DATA 652F,20,50,20,46,20,4A,20,55,045D [1915]
440 DATA 6537,20,53,20,54,20,41,20,47,0583 [1359]
450 DATA 653F,20,45,18,0F,03,1F,08,04,02C0 [1214]
460 DATA 6547,57,72,69,74,74,65,6E,20,3FA8 [2462]
470 DATA 654F,62,79,20,4F,6C,69,76,65,2DFD [2176]
480 DATA 6557,72,20,47,65,69,73,65,6E,3D90 [1825]
490 DATA 655F,0F,02,1F,0A,0A,28,31,29,04FB [2395]
500 DATA 6567,20,20,20,20,49,6E,73,74,1D62 [1337]
510 DATA 656F,72,75,6B,74,69,6F,6E,65,2C2D [1591]
520 DATA 6577,6E,1F,0A,0E,28,32,29,20,309A [1474]
530 DATA 657F,20,20,20,50,72,6F,67,72,1B90 [1527]
540 DATA 6587,61,6D,6D,1F,0A,12,28,33,27EB [2489]
550 DATA 658F,29,20,20,20,20,0F,01,18,1BA6 [1121]
560 DATA 6597,45,4E,44,45,18,0F,01,1F,3D31 [2329]
570 DATA 659F,06,19,28,43,29,20,31,39,05E3 [1233]
580 DATA 65A7,38,37,20,20,20,62,79,20,179A [2127]
590 DATA 65AF,20,20,2D,47,2D,2D,53,2D,1887 [2572]
600 DATA 65B7,6F,2D,66,2D,74,2D,00,0F,31CB [2089]
610 DATA 65BF,02,1F,01,01,42,69,74,74,05D8 [921]
620 DATA 65C7,65,20,62,65,66,6F,6C,67,32A3 [1698]
630 DATA 65CF,65,6E,20,53,69,65,20,64,2AC8 [1944]
640 DATA 65D7,69,65,20,41,6E,77,65,69,2FDF [2002]
650 DATA 65DF,73,75,6E,67,65,6E,0D,0A,2DF0 [2861]
660 DATA 65E7,64,65,72,20,62,65,69,6C,253A [1915]
670 DATA 65EF,69,65,67,65,6E,64,65,6E,2534 [2161]
680 DATA 65F7,20,41,6E,6C,65,69,74,75,0951 [1766]
690 DATA 65FF,6E,67,2E,00,0F,01,1F,01,2B43 [2456]
700 DATA 6607,01,4B,61,73,73,65,74,74,1BC0 [1685]
710 DATA 660F,65,20,65,69,6E,6C,65,67,32DD [1724]
720 DATA 6617,65,6E,20,75,6E,64,20,20,28D0 [1917]
730 DATA 661F,50,4C,41,59,20,64,72,75,36B1 [1558]
740 DATA 6627,65,63,6B,65,6E,2E,1F,01,2287 [2461]
750 DATA 662F,03,44,61,6E,6E,20,54,6F,1977 [2270]
760 DATA 6637,6E,6B,6F,70,66,20,61,75,2427 [2166]
770 DATA 663F,66,20,47,72,75,6E,64,73,366B [1549]
780 DATA 6647,74,65,6C,6C,75,6E,67,20,2AFE [2496]
790 DATA 664F,62,72,69,6E,67,65,6E,2E,241E [2221]
800 DATA 6657,1F,0D,19,18,50,72,65,73,0D91 [2151]
810 DATA 665F,73,20,61,6E,79,20,6B,65,38BB [1853]
820 DATA 6667,79,18,00,00,00,00,00,00,3A80 [1295]
830 DATA *ENDE*                           [502]
840 adr=&6400:zeile=50:MEMORY adr-1       [1870]
850 READ d$:IF d$="*ENDE*"THEN 960        [1112]
860 pr=0                                  [117]
870 FOR i=1 TO 8                          [462]
880 READ a$:a=VAL("&"+a$)                 [883]
890 POKE adr,a:adr=adr+1                  [717]
900 pr=pr*2:IF pr>65535 THEN pr=pr-65535  [2485]
910 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+655 [1890]
36
920 NEXT i                                [375]
930 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THE [2610]
N pr2=pr2+65536
940 IF pr<>pr2 THEN PRINT"Pruefsummenfehle [3615]
r in Zeile";zeile:STOP
950 zeile=zeile+10:GOTO 850               [1911]
960 SAVE"TONJUST.BIN",B,&6400,&270        [2014]
970 PRINT d$:END                          [346]
```



# Deluxe Data System

**Dateiverwaltungen gibt es zur Zeit wie Sand am Meer. Neben speziellen Dateiverwaltungen für Adressen, Vokabeln etc. gibt es auch solche, die universell einsetzbar sind. Doch leider arbeiten die meisten dieser Programme mit sequentiellen Dateien, das heißt, man kann eine bestimmte Anzahl von Datensätzen eingeben, diese stehen dann im RAM Speicher und können editiert und auf Diskette abgespeichert werden. Jedoch ist der Speicher des Schneider CPC, speziell bei den Modellen 464/664, nicht besonders groß, so daß man, wenn man größere Datenbestände zu verwalten hat, seine Daten auf mehrere Dateien aufteilen muß.**

Abhilfe schafft hier eine relative Dateiverwaltung, bei der die Datensätze sofort auf Disc geschrieben werden und bei der Ausgabe von der Diskette gelesen werden. Leider unterstützt das sonst sehr gute Schneider Locomotive Basic keine relativen Dateien, so daß ich mein Programm in Turbo Pascal 3.0 geschrieben habe.

## Der Aufbau des Programmes

Das Programm DELUXE DATA SYSTEM ist eine universelle Dateiverwaltung, die mit relativen Dateien arbeitet. Das Programm läuft unter Turbo Pascal und ist als Programmbibliothek aufgebaut. Dies hat den Vorteil, daß man das Programm sehr leicht verbessern kann. DATA DELUXE SYSTEM besteht aus 10 einzelnen Modulen:

1. DDS .PAS Hauptprogramm
2. SCR-INIT.INC Initialisierung des Bildschirms
3. NEW-COM .INC Prozeduren zum Abspeichern, Laden, Ausgabe eines Datensatzes etc.
4. INIT .INC INITMENU.INC Menuepunkt Initialisierung
5. EINGABE .INC Menuepunkt Eingabe
6. AUSGABE .INC Menuepunkt Ausgabe
7. SUCHEN .INC Menuepunkt Suchen
8. AENDERN .INC Menuepunkt Ändern
9. LOESCHEN.INC Menuepunkt Löschen
10. MENU .INC Hauptmenue

Das 4. Modul paßt nicht als ganze Textfile in den Turbo-Editor des Schneider CPC 464/664, so daß dieses Modul in die Dateien INIT.INC und INITMENU.INC aufgeteilt werden muß. Das Programm wird wie folgt compiliert:

1. Turbo starten
2. Frage »Include error message (Y/N) ?« mit N beantworten
3. Mit M DDS.PAS als Mainfile angeben
4. Mit O das Compiler-Options-Menue anwählen
5. Mit C Compiler-Option auf COM-File einstellen
6. Mit Q das Compiler-Options-Menue verlassen
7. Mit C compilieren
8. Mit Q Turbo verlassen

Nun kann man mit DDS das Programm DELUXE DATA SYSTEM aufrufen.

## DELUXE DATA SYSTEM im Detail

Nachdem man DELUXE DATA SYSTEM gestartet hat, befindet man sich im Hauptmenue. In den ersten beiden Zeilen befindet sich die Überschrift und der Copyrightvermerk. In der 4. Zeile stehen die Menuepunkte Initialisierung, Eingabe, Ausgabe, Suchen, Ändern, Löschen und Ende. Mit den Cursortasten ← und → kann man die Menuepunkte anwählen. Durch Drücken der Leertaste wird der angewählte Menuepunkt ausgeführt. Im folgenden werden die einzelnen Menuepunkte erläutert:

## Initialisierung

Unter dem Hauptmenue öffnet sich ein Dialog Window. In diesem erscheint ein Untermenue mit folgenden Menuepunkten:

<1> Farben einstellen
Falls Ihnen die Grundeinstellung der Farben nicht gefällt, können Sie die Farben ändern. Die Farbcodes des CPC stehen im Handbuch. Orientieren Sie sich nicht nach dem geöffneten Window, sondern nach der Überschrift oder nach dem Hauptmenue, da beim Öffnen eines Windows Hintergund- und Stiftfarben vertauscht werden (Inversdarstellung).

<2> Maske eingeben
Unter diesem Menuepunkt können Sie die Datenmaske eingeben. Eine Datenmaske beinhaltet die Struktur eines Datensatzes. Sie müssen die Anzahl der Datenfelder pro Datensatz eingeben. Danach müssen Sie für jedes Datenfeld ein Maskenstring und die maximale Länge des Datenfeldes eingeben. Ein Maskenstring ist die Bezeichnung eines Datenfeldes. Er kann bis zu 20 Zeichen lang sein. Sie können bis zu 30 Datenfelder mit je 255 Zeichen definieren. So ist es möglich, sehr komplexe Datenbestände zu verwalten.

<3> Maske speichern
Um nicht immer wieder beim Neustart des DELUXE DATA SYSTEM die Datenmaske neu eingeben zu müssen, können Sie die Maske auf Diskette abspeichern und später wieder unter Menuepunkt <4> von Diskette laden. Sie müssen einen Dateinamen eingeben, der nicht länger als 8 Zeichen lang sein darf.

Übrigens sei hier angemerkt, daß man unter CP/M viel mehr Möglichkeiten zur Gestaltung des Dateinamens hat, als unter AMSDOS. Zeichen wie »/«, »□« oder Klammeraffe werden ohne weiteres angenommen. Allerdings kann man auf diese Dateien von AMSDOS nicht zugreifen. Das Löschen der Datei mit ERA oder das Umbenennen mit REN ist von AMSDOS ebenfalls unmöglich.

<4> Maske laden
Hier kann man eine Maske von Diskette laden. Wird ein Dateiname eingegeben, der nicht existiert, erhält man eine Fehlermeldung.

<5> Suchmaske
Sie können hier festlegen, nach welchem Datenfeld gesucht werden soll. Wird hier kein Datenfeld als »Suchfeld« (dieser Ausdruck wird im Laufe dieser Beschreibung noch häufiger vorkommen) definiert, so wird das erste Datenfeld als Suchfeld angenommen.

<6> Ende
Rückkehr zum Hauptmenue.

DELUXE DATA SYSTEM benutzt zwei Standardprozeduren, die in jedem Menuepunkt gleich sind. Da diese ja immer gleich sind, erkläre ich hier die Funktion dieser Prozeduren. Zum einen wird, falls kein Dateiname gespeichert ist, die Eingabe eines Dateinamens verlangt.
Ist schon ein Dateiname gespeichert, hat man die Möglichkeit, diesen zu ändern, um ggf. eine andere Datei zu bearbeiten. Ist die Datei nicht vorhanden, so erhalten Sie bei den Menuepunkten Ausgabe, Suchen, Ändern, und Löschen eine Fehlermeldung und gelangen in das Hauptmenue zurück.
Bei dem Menuepunkt >Eingabe< wird eine neue Datei eröffnet.
Bei den Menuepunkten >Ausgabe<, >Suchen<, und >Ändern< haben Sie die Möglichkeit, den Bereich, in dem nach einem Datensatz gesucht werden soll, zu beschränken. Zum Beispiel können Sie einen Datensatz im Bereich vom 132. bis zum 296. Datensatz suchen lassen. Der Rest der Datei wird ignoriert.
Zuerst werden Sie gefragt: »Ab welchem Datensatz«. Drücken Sie nur <ENTER>, so wird 1 angenommen. Danach wird die Frage »Bis zu welchem Datensatz« gestellt. Drücken Sie nur <ENTER>, so wird bis zum Ende der Datei gesucht. Dies geht nur, falls die Datei nicht mehr als 9999 Datensätze hat.
Übersteigt die eingegebene Zahl die Anzahl der Datensätze der Datei, so wird ebenfalls bis zum Ende der Datei gesucht. Falls keine Maske gespeichert ist, erhalten Sie bei den Menuepunkten Eingabe, Ausgabe, Suchen, Ändern und Löschen eine Fehlermeldung und gelangen zum Hauptmenue zurück.

**Eingabe:**
Hier können Sie Datensätze in eine Datei schreiben. Sind in der Datei schon Datensätze enthalten, so wird die Datei erweitert. Der ersten Zeile des Dialog Windows können Sie entnehmen, welchen Datensatz Sie im Moment eingeben. Drücken Sie <ENTER>, wenn die Eingabe des ersten Datenfeldes gefordert wird, gelangen Sie in das Hauptmenue zurück.

**Ausgabe:**
Unter diesem Menuepunkt können Sie sich einen Überblick über Ihre Datenbestände verschaffen. Es wird immer das Suchfeld eines Datensatzes innerhalb des Bereichs entweder auf dem Bildschirm oder Drucker ausgeben.
Wenn Sie den Bildschirm als Ausgabegerät wählen, wird das Dialog Window vollgeschrieben, dann wartet das Programm bis Sie die Leertaste drücken, dann wird die Ausgabe fortgesetzt.

**Suchen:**
Nach eventueller Eingabe eines neuen Dateinamens und Eingabe des Bereichs, werden Sie aufgefordert einen Suchbegriff einzugeben. Alle Datensätze (innerhalb des Bereichs), in deren Suchfeld der Suchbegriff enthalten ist, werden komplett auf dem Bildschirm ausgegeben. Erscheint ein Datensatz auf dem Bildschirm, so können Sie entscheiden, ob dieser Datensatz ausgedruckt werden soll.

**Ändern:**
Nach den üblichen Standardprozeduren müssen Sie einen Suchbegriff und den neuen Inhalt des Suchfeldes eingeben. Wird ein Datensatz gefunden, in dessen Suchfeld der Suchbegriff enthalten ist, so wird dieser wieder auf dem Bildschirm ausgegeben, und Sie entscheiden, ob das Suchfeld verändert werden soll oder nicht.

**Löschen:**
Dieser Menuepunkt unterscheidet sich kaum vom Menuepunkt Suchen. Sie entscheiden nur nicht, ob der Datensatz ausgedruckt werden soll, sondern ob der Datensatz gelöscht werden soll.

**Ende:**
Beenden des Programms und Rückkehr zu CP/M.

## Kompatibel???

Wie Sie als Turbo-User sicher wissen, gibt es Turbo-Pascal für die unterschiedlichsten Systeme (CP/M 80, CP/M 86, MS-DOS, PC-DOS u.a.). Es ist aber fraglich, ob DELUXE DATA SYSTEM auf anderen Computern lauffähig ist. Das Programm benutzt nämlich Steuerzeichen zur Farbeinstellung und zur Windowtechnik. Sicher ist aber, daß das Programm auf dem CPC 464, CPC 664 sowie auf dem CPC 6128 läuft.

## Benutzertips

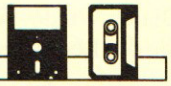
Das Programm ist so gestaltet, daß es bedingt durch Falscheingaben nicht abbricht oder sogar abstürzt. Ein Auftreten von Run-Time-Errors ist unwahrscheinlich. Wechseln Sie während des Programmablaufs nie die Diskette!!! Dann kommt es nämlich zum gefürchteten BDOS-Error, und das Programm wird abgebrochen. Das Programm (COM-File) nimmt 17 KByte in Anspruch, eine Maskendatei belegt etwa 1 KByte auf der Diskette, so daß 151 KB für Dateien frei sind.

(Jochen Skulj)

Die Zeilennummern in den Listings dürfen nicht mit eingetippt werden, sie sind nur als Eingabe-HILFE gedacht.

Übrigens ist das lauffähige Programm auch als COM-File auf unserer Data-Box.

## für 464-664-6128



```
(-------------------------------------------------------------------------)
(                  Programmname : Deluxe Data System                      )
(                  Dateiname    : DDS.PAS                                 )
(                  Modul        : Hauptprogramm                           )
(                  Autor        : Jochen Skulj, 1987                      )
(-------------------------------------------------------------------------)

PROGRAM Deluxe_Data_System;

(-------------------------------------------------------------------------)
( Konstanten-, Typen- und Variablendefinition                             )
(-------------------------------------------------------------------------)

CONST
     Extension_1 = '.DAT';
     Extension_2 = '.MSK';

TYPE
     Datentyp   = ARRAY [1..30] OF STRING [255];
     Maskentyp  = ARRAY [1..30] OF STRING [20];
     Strtyp     = STRING [255];

VAR
   Daten        : Datentyp;
   Maske        : Maskentyp;
   Datei        : FILE OF CHAR;
   Max_Laenge   : ARRAY [1..30] OF INTEGER;
   Maximal      : INTEGER;
   Drucker      : INTEGER;
   Dateiname    : STRING [8];
   Such_Num     : INTEGER;
   Start_Pos    : INTEGER;
   End_Pos      : INTEGER;
   Position     : INTEGER;
   Exist        : BOOLEAN;

(-------------------------------------------------------------------------)
( INCLUDE - Dateien nachladen                                             )
(-------------------------------------------------------------------------)

(*I SCR-INIT.INC )                    ( Bildschirm aufbauen )
(*I NEW-COM.INC )                     ( Neue Kommandos      )
(*I INIT.INC )                        ( Initialisierung     )
(*I INITMENU.INC )
(*I EINGABE.INC )                     ( Eingabe             )
(*I AUSGABE.INC )                     ( Ausgabe             )
(*I SUCHEN.INC )                      ( Suchen              )
(*I AENDERN.INC )                     ( Aendern             )
(*I LOESCHEN.INC )                    ( Loeschen            )
(*I MENU.INC )                        ( Hauptmenue          )

(-------------------------------------------------------------------------)
( Hauptprogramm                                                           )
(-------------------------------------------------------------------------)

BEGIN
     Dateiname:='';
     Maximal:=0;
     Such_Num:=1;
     Screen_Init;
     Hauptmenue;
END.
```

```
(-------------------------------------------------------------------------)
(                  Programmname : Deluxe Data System                      )
(                  Dateiname    : SCR-INIT.INC                            )
(                  Modul        : Unterprogramm, Teil I                   )
(                                 Bildschirm-Initialisierung              )
(                  Autor        : Jochen Skulj, 1987                      )
(-------------------------------------------------------------------------)

PROCEDURE Window (Links, Rechts, Oben, Unten : INTEGER);

BEGIN
     Write (#26,Chr (Links),Chr (Rechts),Chr (Oben),Chr (Unten));
END;

(-------------------------------------------------------------------------)

PROCEDURE Paper (Farbe : INTEGER);

BEGIN
     Write (#28,Chr (0),Chr (Farbe),Chr (Farbe));
END;

(-------------------------------------------------------------------------)

PROCEDURE Pen (Farbe : INTEGER);

BEGIN
     Write (#28,Chr (1),Chr (Farbe),Chr (Farbe));
END;

(-------------------------------------------------------------------------)

PROCEDURE Border (Farbe : INTEGER);

BEGIN
     Write (#29,Chr (Farbe),Chr (Farbe));
END;

(-------------------------------------------------------------------------)

PROCEDURE Dialog_Window;

BEGIN
     Window (1,80,6,23);
     LowVideo;
     ClrScr;
END;

(-------------------------------------------------------------------------)

PROCEDURE Fehler_Window;

VAR
   Lauf : INTEGER;

BEGIN
     Window (1,80,25,25);
     LowVideo;
     ClrScr;
END;

(-------------------------------------------------------------------------)

PROCEDURE Close_Window;

BEGIN
     NormVideo;
     ClrScr;
END;

(-------------------------------------------------------------------------)

PROCEDURE Screen_Init;

BEGIN
     ClrScr;
     Window (1,80,1,2);
     Paper (0);
     Border (0);
     Pen (9);
     GotoXY (20,1);
     Write ('D E L U X E   D A T A   S Y S T E M   1.0');
     GOTOXY (24,2);
     Write ('(C) Copyright 1987 by Jochen Skulj');
     Window (1,80,3,5);
     GotoXY (1,2);
     Write ('Initialisierung');
     GotoXY (20,2);
     Write ('Eingabe');
     GotoXY (31,2);
     Write ('Ausgabe');
     GotoXY (42,2);
     Write ('Suchen');
     GotoXY (52,2);
     Write ('Aendern');
     GotoXY (63,2);
     Write ('Loeschen');
     GotoXY (75,2);
     Write ('Ende');
     Window (1,80,24,24);
     Drucker:=1;
END;


(-------------------------------------------------------------------------)
```

```
(-------------------------------------------------------------------------)
(                  Programmname : Deluxe Data System                      )
(                  Dateiname    : NEW-COM.INC                             )
(                  Modul        : Unterprogramm, Teil II                  )
(                                 Befehlserweiterung                      )
(                  Autor        : Jochen Skulj, 1987                      )
(-------------------------------------------------------------------------)

PROCEDURE Data_Write;

Var
   Lauf_1  :INTEGER;
   Lauf_2  :INTEGER;
   WriteCh :CHAR;

Begin
     Position:=Position+1;
     For Lauf_1:=1 To Maximal Do
     Begin
          If Length (Daten[Lauf_1])<Max_Laenge[Lauf_1] Then
          Begin
               For Lauf_2:=Length (Daten[Lauf_1])+1 To Max_Laenge[Lauf_1] Do
                    Daten[Lauf_1]:=Daten[Lauf_1]+' ';
          End;
          For Lauf_2:=1 To Max_Laenge[Lauf_1] Do
          Begin
               WriteCh:=Copy (Daten[Lauf_1],Lauf_2,1);
               Write (Datei, WriteCh);
          End;
     End;
End;

Procedure Data_Read;

Var
   Lauf_1, Lauf_2 :INTEGER;
   ReadCh :CHAR;

Begin
     Position:=Position+1;
     For Lauf_1:=1 To Maximal Do
     Begin
          Daten[Lauf_1]:='';
          For Lauf_2:=1 To Max_Laenge[Lauf_1] Do
          Begin
               Read (Datei, ReadCh);
               Daten(.Lauf_1.):=Concat (Daten(.Lauf_1.), ReadCh);
          End;
     End;
End;

(-------------------------------------------------------------------------)


PROCEDURE Set_Pointer (Anzahl : INTEGER);      ( Filepointer auf den n-ten )
                                               ( Datensatz                 )
VAR
   Max_Summe : INTEGER;
   Lauf      : INTEGER;
   Okay      : BOOLEAN;

BEGIN
     Max_Summe:=0;
     Position:=0;
     For Lauf:=1 To Maximal Do
          Max_Summe:=Max_Summe+Max_Laenge [Lauf];
     Seek (Datei,0);
     For Lauf:=1 To Anzahl Do
     BEGIN
          IF EOF (Datei)=FALSE Then
          BEGIN
               Seek (Datei, FilePos(Datei)+Max_Summe);
               Position:=Position+1;
          END;
     END;
END;

(-------------------------------------------------------------------------)


PROCEDURE Fehler;

BEGIN
```

```
84:       Fehler_Window;
85:       GotoXY (31,1);
86:       Write ('Eingabe - Fehler');
87:       Delay (2000);
88:       Close_Window;
89:       Window (1,80,6,23);
90: END;
91:
92: {------------------------------------------------------------------------}
93:
94: PROCEDURE Test;  ( Andere Datei eroeffnen ? )
95:
96: VAR
97:    Wahl : CHAR;
98:
99: BEGIN
100:      If Length (Dateiname)>0 Then
101:      BEGIN
102:           Fehler_Window;
103:           Write ('Dateiname : ',Dateiname,' - Moechten Sie den Namen aendern (J/N) ? ');
104:           Repeat
105:                Read (KBD, Wahl);
106:           Until (Wahl='J') OR (Wahl='N');
107:      END;
108:      If (Length (Dateiname)=0) OR (Wahl='J') Then
109:      BEGIN
110:           Fehler_Window;
111:           ($I-)
112:           Write ('Bitte geben Sie den Dateinamen ein (max. 8) ! ');
113:           Read (Dateiname);
114:           ($I+)
115:      END;
116:      Exist:=TRUE;
117:      Assign (Datei, Dateiname+Extension_1);
118:      ($I-)
119:      Reset (Datei);
120:      If IOResult<>0 Then
121:      BEGIN
122:           Exist:=FALSE;
123:           Rewrite (Datei);
124:      END;
125:      ($I+)
126:      Close_Window;
127: END;
128:
129: {------------------------------------------------------------------------}
130:
131: PROCEDURE Data_Out (Kanal : Strtyp);  ( Datensatz auf Drucker oder Bildschirm )
132:                                       ( oder Drucker ausgeben                  )
133: VAR
134:    Lauf_1 : INTEGER;
135:    Lauf_2 : INTEGER;
136:    Laenge : INTEGER;
137:    Out    : TEXT;
138:
139: BEGIN
140:      If Kanal='Printer' Then
141:      BEGIN
142:           NormVideo;
143:           Assign (Out,'LST:');
144:      END
145:      ELSE
146:          Assign (Out,'CON:');
147:      Laenge:=Length (Maske [1]);
148:      For Lauf_1:=2 To Maximal Do
149:      BEGIN
150:           If Length (Maske [Lauf_1])>Laenge Then
151:              Laenge:=Length (Maske [Lauf_1]);
152:      END;
153:      Writeln (Out,'Datensatz Nr.: ',Position);
154:      Writeln (Out);
155:      For Lauf_1:=1 To Maximal Do
156:      BEGIN
157:           Write (Out,Maske [Lauf_1]);
158:           For Lauf_2:=Length (Maske [Lauf_1]) To Laenge Do
159:               Write (Out,' ');
160:           Write (Out,': ');
161:           Writeln (Out, Daten [Lauf_1]);
162:      END;
163:      If Kanal='Printer' Then
164:         LowVideo;
165: END;
166:
167:
168: {------------------------------------------------------------------------}
169:
170: PROCEDURE Pruefen;
171:
172: BEGIN
173:      If (Maximal=0) AND (Exist=FALSE) Then
174:      BEGIN
175:           Fehler_Window;
176:           GotoXY (24,1);
177:           Write ('Maske und  Datei nicht vorhanden');
178:           Delay (2000);
179:           Close_Window;
180:      END
181:      ELSE
182:      BEGIN
183:           If Maximal=0 Then
184:           BEGIN
185:                Fehler_Window;
186:                GotoXY (26,1);
187:                Write ('Es ist keine Maske vorhanden');
188:                Delay (2000);
189:                Close_Window;
190:           END;
191:           If Exist=FALSE Then
192:           BEGIN
193:                Fehler_Window;
194:                GotoXY (26,1);
195:                Write ('Es ist keine Datei vorhanden');
196:                Delay (2000);
197:                Close_Window;
198:           END;
199:      END;
200: END;
201:
202: {------------------------------------------------------------------------}
203:
204: PROCEDURE Bereich;
205:
206: BEGIN
207:      Repeat
208:           Repeat
209:                Fehler_Window;
210:                Write ('Ab welchem Datensatz : ');
211:                ($I-)
212:                Read (Start_Pos);
213:                ($I+)
214:           Until IOResult=0;
215:           Repeat
216:                Fehler_Window;
217:                Write ('Bis zu welchem Datensatz : ');
218:                ($I-)
219:                Read (End_Pos);
220:                ($I+)
221:                Close_Window;
222:           Until IOResult=0;
223:           If End_Pos=0  Then
224:           BEGIN
225:                End_Pos:=9999;
226:           END;
227:      Until Start_Pos<End_Pos;
228: END;
```

```
1: {------------------------------------------------------------------------}
2: {                Programmname : Deluxe Data System                       }
3: {                Dateiname    : INIT.INC                                 }
4: {                Modul        : Unterprogramm, Teil III                  }
5: {                               Initialisierung                          }
6: {                Autor        : Jochen Skulj, 1987                       }
7: {------------------------------------------------------------------------}
8:
9: PROCEDURE Initialisierung;
10:
11: VAR
12:    Menu_Wahl : INTEGER;
13:    Lauf_Var  : INTEGER;
14:
15:    PROCEDURE Farben;  ( Farben neu einstellen )
16:
17:    VAR
18:       Stift       : INTEGER;
19:       Hintergrund : INTEGER;
20:       Okay        : BOOLEAN;
21:       Lauf_Var    : INTEGER;
22:
23:    BEGIN
24:         Repeat
25:              Okay:=FALSE;
26:              Dialog_Window;
27:              Writeln;
28:              Repeat
29:                   Write ('Bitte geben Sie die Stift-Farbe ein (0 - 26) ! ');
30:                   ($I-)
31:                   Readln (Stift);
32:                   ($I+)
33:              Until IOResult=0;
34:              Writeln;
35:              Repeat
36:                   Write ('Bitte geben Sie die Hintergrund-Farbe ein (0 - 26) ! ');
37:                   ($I-)
38:                   Readln (Hintergrund);
39:                   ($I+)
40:              Until IOResult=0;
41:              If (Stift>-1) OR (Stift<27) OR (Stift<>Hintergrund) Then
42:                 Okay:=TRUE;
43:              IF (Hintergrund>-1) OR (Hintergrund<27) OR (Hintergrund<>Stift) Then
44:                 Okay:=TRUE;
45:              IF Okay=FALSE Then
46:                 Fehler;
47:         Until Okay=TRUE;
48:         Close_Window;
49:         Pen (Stift);
50:         Paper (Hintergrund);
51:         Border (Hintergrund);
52:    END;
53:
54: {------------------------------------------------------------------------}
55:
56:    PROCEDURE Maske_Eingeben;
57:
58:    VAR
59:       Lauf   : INTEGER;
60:
61:    BEGIN
62:         Repeat
63:              Dialog_Window;
64:              Writeln;
65:              Write ('Bitte geben Sie die Anzahl der Datensaetze ein (1 - 30) ! ');
66:              ($I-)
67:              Readln (Maximal);
68:              ($I+)
69:              If (Maximal<1) OR (Maximal>30) OR (IOResult<>0) Then
70:                 Fehler;
71:         Until (Maximal>0) AND (Maximal<31) AND (IOResult=0);
72:         For Lauf:=1 To Maximal Do
73:         BEGIN
74:              Dialog_Window;
75:              Writeln;
76:              ($I-)
77:              Write ('Bitte geben Sie den ',Lauf,'. Maskenstring ein (max. 20) ! ');
78:              Readln (Maske [Lauf]);
79:              ($I+)
80:              Writeln;
81:              Repeat
82:                   GotoXY (1,4);
83:                   For Lauf_Var:=1 To 80 Do
84:                       Write (' ');
85:                   GotoXY (1,4);
86:                   Write ('Bitte geben Sie die maximale Laenge ein (max. 255) ! ');
87:                   Read (Max_Laenge [Lauf]);
88:                   If (Max_Laenge [Lauf]<1) OR (Max_Laenge [Lauf]>255) OR (IOResult<>0)
                      then
89:                      Fehler;
90:              Until (Max_Laenge [Lauf]>0) AND (Max_Laenge [Lauf]<256) AND (IOResult=0);
91:         END;
92:         Close_Window;
93:    END;
94:
95: {------------------------------------------------------------------------}
96:
97:    PROCEDURE Maske_Speichern;
98:
99:    TYPE
100:       Masken_Record = RECORD
101:                          Str : STRING [20];
102:                          Len : INTEGER;
103:                       END;
104:
105:    VAR
106:       Masken_Satz  : Masken_Record;
107:       Masken_Datei : FILE OF Masken_Record;
108:       Lauf_Var     : INTEGER;
109:
110:    BEGIN
111:         Dialog_Window;
112:         Writeln;
113:         ($I-)
114:         Write ('Bitte geben Sie den Dateinamen ein (max. 8) ! ');
115:         Readln (Dateiname);
116:         ($I+)
117:         Assign (Masken_Datei, Dateiname+Extension_2);
118:         Rewrite (Masken_Datei);
119:         Masken_Satz.Str:='';
120:         Masken_Satz.Len:=Maximal;
121:         Write (Masken_Datei, Masken_Satz);
122:         For Lauf_Var:=1 To Maximal Do
```

```
123:        BEGIN
124:             Masken_Satz.Str:=Maske[Lauf_Var];
125:             Masken_Satz.Len:=Max_Laenge[Lauf_Var];
126:             Write #Masken_Datei, Masken_Satz;
127:        END;
128:        Close (Masken_Datei);
129:        Close_Window;
130:   END;
131:
132: {------------------------------------------------------------------------}
133:
134:   PROCEDURE Maske_Laden;
135:
136:   TYPE
137:       Masken_Record = RECORD
138:                         Str : STRING[20];
139:                         Len : INTEGER;
140:                       END;
141:
142:   VAR
143:      Masken_Satz  : Masken_Record;
144:      Masken_Datei : FILE OF Masken_Record;
145:      Lauf_Var     : INTEGER;
146:
147:   BEGIN
148:        Dialog_Window;
149:        Writeln;
150:        ($I-)
151:        Write ('Bitte geben Sie den Dateinamen ein (max. 8) ! ');
152:        Readln (Dateiname);
153:        Assign (Masken_Datei, Dateiname+Extension_2);
154:        Reset (Masken_Datei);
155:        ($I+)
156:        If IOResult<>0 Then
157:        BEGIN
158:             Fehler_Window;
159:             GotoXY (30,1);
160:             Write ('Maske  nicht vorhanden');
161:             Delay (2000);
162:             Close_Window;
163:        END
164:        ELSE
165:        BEGIN
166:             Read (Masken_Datei, Masken_Satz);
167:             Maximal:=Masken_Satz.Len;
168:             For Lauf_Var:=1 To Maximal Do
169:             BEGIN
170:                  Read (Masken_Datei, Masken_Satz);
171:                  Maske[Lauf_Var]:=Masken_Satz.Str;
172:                  Max_Laenge[Lauf_Var]:=Masken_Satz.Len;
173:             END;
174:             Close (Datei);
175:        END;
176:        Dialog_Window;
177:        Close_Window;
178:   END;
179:
180: {------------------------------------------------------------------------}
181:
182:   PROCEDURE Suchmaske;
183:
184:   BEGIN
185:        Dialog_Window;
186:        Writeln;
187:        Writeln ('    Suchmasken - Menue    ':53);
188:        Writeln ('    ------------------    ':53);
189:        Writeln;
190:        For Lauf_Var:=1 To Maximal Do
191:        BEGIN
192:             Write ('<':32);
193:             If Lauf_Var<10 Then
194:                Write ('0');
195:             Write (Lauf_Var);
196:             Write ('> ');
197:             Writeln (Maske[Lauf_Var]);
198:             Writeln;
199:        END;
200:        Repeat
201:             Fehler_Window;
202:             GotoXY (35,1);
203:             ($I-)
204:             Write ('Ihre Wahl :');
205:             ($I+)
206:             Read (Such_Num);
207:        Until (Such_Num>0) AND (Such_Num<Maximal+1) AND (IOResult=0);
208:        Close_Window;
209:        Window (1,80,6,23);
210:        Close_Window;
211:   END;
212:
213: {------------------------------------------------------------------------}
```

```
1: {------------------------------------------------------------------------}
2: {                  Programmname : Deluxe Data System                     }
3: {                  Dateiname    : EINGABE.INC                            }
4: {                  Modul        : Unterprogramm, Teil IV                 }
5: {                                 Eingabe der Datensaetze                }
6: {                  Autor        : Jochen Skulj, 1987                     }
7: {------------------------------------------------------------------------}
8:
9: PROCEDURE Eingabe;
10:
11: VAR
12:    Lauf_Var_1 : INTEGER;
13:    Lauf_Var_2 : INTEGER;
14:    Laenge     : INTEGER;
15:
16: BEGIN
17:      If Maximal<1 Then
18:      BEGIN
19:           Fehler_Window;
20:           GotoXY (26,1);
21:           Write ('Es ist keine Maske vorhanden');
22:           Delay (2000);
23:           Close_Window;
24:      END
25:      ELSE
26:      BEGIN
27:           Test;
28:           Laenge:=Length (Maske[1]);
29:           For Lauf_Var_1:=2 To Maximal Do
30:           BEGIN
31:                If Length (Maske[Lauf_Var_1])>Laenge Then
32:                   Laenge:=Length (Maske[Lauf_Var_1]);
33:           END;
34:           Set_Pointer (9999);
35:           Position:=Position+1;
```

```
36:           Repeat
37:                Dialog_Window;
38:                Writeln ('Datensatz : ',Position);
39:                Writeln;
40:                Write (Maske[1]);
41:                For Lauf_Var_2:=Length (Maske[1]) To Laenge Do
42:                    Write (' ');
43:                Write (': ');
44:                ($I-)
45:                Readln (Daten[1]);
46:                If Length (Daten[1])>0 Then
47:                BEGIN
48:                     For Lauf_Var_1:=2 To Maximal Do
49:                     BEGIN
50:                          Write (Maske[Lauf_Var_1]);
51:                          For Lauf_Var_2:=Length (Maske[Lauf_Var_1]) To Laenge Do
52:                              Write (' ');
53:                          Write (': ');
54:                          ($I-)
55:                          Readln (Daten[Lauf_Var_1]);
56:                          ($I+)
57:                     END;
58:                     Data_Write;
59:                END;
60:           Until Length (Daten[1])=0;
61:           Close (Datei);
62:           Close_Window;
63:      END;
64: END;
65:
66:
67:
68:
```

```
1: {------------------------------------------------------------------------}
2: {                  Programmname : Deluxe Data System                     }
3: {                  Dateiname    : AUSGABE.INC                            }
4: {                  Modul        : Unterprogramm, Teil V                  }
5: {                                 Ausgabe der Datensaetze                }
6: {                  Autor        : Jochen Skulj, 1987                     }
7: {------------------------------------------------------------------------}
8:
9: PROCEDURE Ausgabe;
10:
11: VAR
12:    Lauf_Var_1 : INTEGER;
13:    Lauf_Var_2 : INTEGER;
14:    Wait_Chr   : CHAR;
15:    Wahl       : CHAR;
16:
17: BEGIN
18:      Test;
19:      Pruefen;
20:      If (Exist=TRUE) AND (Maximal>0) Then
21:      BEGIN
22:           Bereich;
23:           ($I-)
24:           Repeat
25:                Fehler_Window;
26:                Write ('Ausgabe auf Drucker (J/N) ? ');
27:                Read (Wahl);
28:                Wahl:=Upcase (Wahl);
29:           Until (IOResult=0) AND ((Wahl='J') OR (Wahl='N'));
30:           ($I+)
31:           If Wahl='N' Then
32:           BEGIN
33:                Close_Window;
34:                Start_Pos:=Start_Pos-1;
35:                Set_Pointer (Start_Pos);
36:                While (Not EOF (Datei)) AND (Position<End_Pos) Do
37:                BEGIN
38:                     Dialog_Window;
39:                     Writeln ('Nr :  ',Maske[Such_Num]);
40:                     For Lauf_Var_2:=1 To 16 Do
41:                     BEGIN
42:                          If (EOF (Datei)=FALSE) AND (Position<End_Pos) Then
43:                          BEGIN
44:                               Data_Read;
45:                               Write (Position:4);
46:                               Write (': ');
47:                               Writeln (Daten[Such_Num]);
48:                          END;
49:                     END;
50:                     Fehler_Window;
51:                     GotoXY (25,1);
52:                     Write ('>>> Weiter mit <Leertaste> <<<');
53:                     Wait_Chr:='#';
54:                     NormVideo;
55:                     Repeat
56:                          Read (KBD, Wait_Chr);
57:                     Until Wait_Chr=' ';
58:                     LowVideo;
59:                     Close_Window;
60:                END;
61:                Dialog_Window;
62:                Close_Window;
63:           END;
64:           If Wahl='J' Then
65:           BEGIN
66:                Set_Pointer (Start_Pos);
67:                Writeln ('Nr :  ',Maske[Such_Num]);
68:                While (Not EOF (Datei)) AND (Position<End_Pos) Do
69:                BEGIN
70:                     If (EOF (Datei)=FALSE) AND (Position<End_Pos) Then
71:                     BEGIN
72:                          Data_Read;
73:                          Write (LST, Position:4);
74:                          Write (LST, ': ');
75:                          Writeln (LST, Daten[Such_Num]);
76:                     END;
77:                END;
78:           END;
79:           Close (Datei);
80:      END;
81: END;
82:
```

```
{-------------------------------------------------------------------------}
{                  Programmname : Deluxe Data System                      }
{                  Dateiname    : SUCHEN.INC                              }
{                  Modul        : Unterprogramm, Teil VI                  }
{                                 Suchen eines Datensatzes               }
{                  Autor        : Jochen Skulj, 1987                      }
{-------------------------------------------------------------------------}

PROCEDURE Suchen;

VAR
   Wahl     : CHAR;
   Such_Str : STRING [255];

BEGIN
     Test;
     Pruefen;
     If (Exist=TRUE) AND (Maximal>0) Then
     BEGIN
          Bereich;
          Dialog_Window;
          Write (Maske [Such_Num],' : ');
          ($I-)
          Readln (Such_Str);
          ($I+)
          Start_Pos:=Start_Pos-1;
          Set_Pointer (Start_Pos);
          While (Not EOF (Datei)) OR (Position>End_Pos) Do
          BEGIN
               Data_Read;
               If Pos (Such_Str, Daten [Such_Num])>0 Then
               BEGIN
                    Dialog_Window;
                    Writeln;
                    Data_Out ('Screen');
                    ($I-)
                    Repeat
                         Fehler_Window;
                         GotoXY (34,1);
                         Write ('Drucken (J/N) ? ');
                         Read (Wahl);
                         Wahl:=Upcase (Wahl);
                         Close_Window;
                    Until (IOResult=0) AND ((Wahl='J') OR (Wahl='N'));
                    ($I+)
                    If Wahl='J' Then
                       Data_Out ('Printer');
               END;
          END;
          Dialog_Window;
          Close_Window;
          Close (Datei);
     END;
END;
```

```
{-------------------------------------------------------------------------}
{                  Programmname : Deluxe Data System                      }
{                  Dateiname    : AENDERN.INC                             }
{                  Modul        : Unterprogramm, Teil VII                 }
{                                 Aendern der Datensaetze                }
{                  Autor        : Jochen Skulj, 1987                      }
{-------------------------------------------------------------------------}

PROCEDURE Aendern;

VAR
   Alt_Str   : STRING [255];
   Neu_Str   : STRING [255];
   Wahl      : CHAR;

BEGIN
     Test;
     Pruefen;
     If (Exist=TRUE) AND (Maximal>0) Then
     BEGIN
          Bereich;
          Dialog_Window;
          ($I-)
          Write (Maske [Such_Num],' (Alt) : ');
          Readln (Alt_Str);
          Writeln;
          Write (Maske [Such_Num],' (Neu) : ');
          Readln (Neu_Str);
          ($I+)
          Start_Pos:=Start_Pos-1;
          Set_Pointer (Start_Pos);
          While (Not EOF (Datei)) OR (Position>End_Pos) Do
          BEGIN
               Data_Read;
               If Pos (Alt_Str, Daten [Such_Num])>0 Then
               BEGIN
                    Dialog_Window;
                    Writeln;
                    Data_Out ('Screen');
                    ($I-)
                    Repeat
                         Fehler_Window;
                         GotoXY (32,1);
                         Write ('Aendern  (J/N) : ');
                         Read (Wahl);
                         Wahl:=Upcase (Wahl);
                         Close_Window;
                    Until (IOResult=0) AND ((Wahl='J') OR (Wahl='N'));
                    ($I+)
                    If Wahl='J' Then
                    BEGIN
                         Daten [Such_Num]:=Neu_Str;
                         Position:=Position-1;
                         Set_Pointer (Position);
                         Data_Write;
                    END;
               END;
          END;
          Dialog_Window;
          Close_Window;
          Close (Datei);
     END;
END;
```

```
{-------------------------------------------------------------------------}
{                  Programmname : Deluxe Data System                      }
{                  Dateiname    : MENU.INC                                }
{                  Modul        : Unterprogramm, Teil IX                  }
{                                 Hauptmenue                             }
{                  Autor        : Jochen Skulj, 1987                      }
{-------------------------------------------------------------------------}

PROCEDURE Hauptmenue;

CONST
     Links      : CHAR = #242;
     Rechts     : CHAR = #243;
     Choose     : CHAR = #32;
     Items      : ARRAY [1..7] OF STRING [15] = ('Initialisierung',
                                                 'Eingabe',
                                                 'Ausgabe',
                                                 'Suchen',
                                                 'Aendern',
                                                 'Loeschen',
                                                 'Ende');
     Item_Pos   : ARRAY [1..7] OF INTEGER     = (1,
                                                 20,
                                                 31,
                                                 42,
                                                 52,
                                                 63,
                                                 75);

VAR
   Wahl : INTEGER;
   Key  : CHAR;

BEGIN
     Wahl:=1;
     While Wahl<>7 Do
     BEGIN
          Key:='#';
          While Key<>Choose Do
          BEGIN
               Window (1,80,4,4);
               GotoXY (Item_Pos [Wahl],1);
               LowVideo;
               Write (Items [Wahl]);
               NormVideo;
               Key:='#';
               While (Key<>Rechts) AND (Key<>Links) AND (Key<>Choose) Do
                    Read (KBD, Key);
               GotoXY (Item_Pos [Wahl],1);
               Write (Items [Wahl]);
               If Key=Links Then
               BEGIN
                    Wahl:=Wahl-1;
                    If Wahl<1 Then
                       Wahl:=1;
               END;
               If Key=Rechts Then
               BEGIN
                    Wahl:=Wahl+1;
                    If Wahl>7 Then
                       Wahl:=7
               END;
          END;
          Case Wahl Of
               1: Initialisierung;
               2: Eingabe;
               3: Ausgabe;
               4: Suchen;
               5: Aendern;
               6: Loeschen;

          END;
     END;
     Window (1,80,1,25);
     ClrScr;
END;
```

```
{-------------------------------------------------------------------------}
{ Initialisierungsmenue                                                   }
{-------------------------------------------------------------------------}

BEGIN
     Repeat
          Dialog_Window;
          Writeln;
          Writeln ('   Initialisierung    ':50);
          Writeln ('   --------------     ':50); Writeln;
          Writeln ('<1>  Farben Einstellen':50); Writeln;
          Writeln ('<2>  Maske Eingeben   ':50); Writeln;
          Writeln ('<3>  Maske Speichern  ':50); Writeln;
          Writeln ('<4>  Maske Laden      ':50); Writeln;
          Writeln ('<5>  Suchmaske        ':50); Writeln;
          Writeln ('<6>  Hauptmenue       ':50); Writeln;
          Repeat
               Fehler_Window;
               GotoXY (31,1);
               Write ('Ihre Wahl (1 - 6) : ');
               ($I-)
               Read (Menu_Wahl);
               ($I+)
               Close_Window;
          Until (Menu_Wahl>0) AND (Menu_Wahl<8) AND (IOResult=0);
          Case Menu_Wahl Of
               1: Farben;
               2: Maske_Eingeben;
               3: Maske_Speichern;
               4: Maske_Laden;
               5: Suchmaske;
          END;
     Until Menu_Wahl=6;
     Dialog_Window;
     Close_Window;
END;
```

# Windows

## Textfenster-Technik unter CP/M 3.0

Angeregt durch den Beitrag »Graphik unter CP/M 3.0 und Turbo-Pascal« im Sonderheft Nr. 4 habe ich eine Bibliothek mit Routinen zur Bearbeitung von Textfenstern zusammengestellt. In den Inlineprozeduren wird ebenfalls die Funktion USERF im Bios nach Übergabe der benötigten Parameter angesprungen. Die Bibliothek läßt sich beliebig erweitern. Zum Abschluß ein kleiner Tip für all diejenigen, die Turbo-Pascal unter CP/M 2.2 benutzen. Ersetzen Sie die Adresse &FC5A durch &BE9B, dann können Sie die Prozeduren auch in Ihre Turbo-Pascal Programme unter CP/M 2.2. einbauen.

**Literaturhinweise:**

Janneck/Mossakowski,
ROM-Listing CPC 464/664/6128, 1986 M&T Verlag
Brückmann/Englisch/Gerits/Steiger
CPC 664/6128 INTERN, 1985 Data Becker

(Hans-Werner Fromme)

### für 6128

```
(*****************************************************************)
(*                                                               *)
(*   W I N D O W . I N C                                         *)
(*                                                               *)
(*   Bibliothek zur Text-Window Bearbeitung fuer SCHNEIDER CPC6128   *)
(*   unter CP/M 3.0 (Plus).                                      *)
(*                                                               *)
(*   zusammengestellt von   Hans-Werner Fromme   im Maerz 1987   *)
(*                                                               *)
(*---------------------------------------------------------------*)
(*       CPC-Firmware-Routinen werden ueber die Adresse der      *)
(*       BIOS-Funktion Nr. 30 (USERF) = &FC5A initialisiert.     *)
(*---------------------------------------------------------------*)
(*   WINDOW.INC beinhaltet folgende Prozeduren:                  *)
(*                                                               *)
(*   TextInit;                       Initialisierung des Text-Packs  *)
(*   TextReset;                      Ruecksetzen des Text-Packs      *)
(*                                                               *)
(*   WinSelect(WindowNr:byte);       Wahl des aktuellen Fensters     *)
(*   WinEnable(S1,Z1,S2,Z2:byte);    Festlegung der Fenstergrenzen   *)
(*   WinFrameEnable                                              *)
(*            (S1,Z1,S2,Z2:byte);    wie WinEnable, jedoch mit Rahmen *)
(*   WinSwap(WinNr1,WinNr2:byte);    Parameteraustausch zweier Windows *)
(*   WinClear;                       Aktuelles Fenster loeschen      *)
(*                                                               *)
(*   WinSetCursor(X,Y:byte);         Cursor im akt. Fenster positionieren *)
(*   WinGetCursor(VAR X,Y:byte);     Position des Cursors abfragen   *)
(*   WinCursorOn;                    Cursor einschalten              *)
(*   WinCursorOff;                   Cursor abschalten               *)
(*                                                               *)
(*   WinSetPen(PenNr:byte);          Schreibfarbe f. akt. Fenster setzen *)
(*   WinSetPaper(PaperNr:byte);      Hintergrundfarbe f. akt. F. setzen *)
(*   WinInvers;                      Pen- und Paper-Farbe tauschen   *)
(*   WinSetBack(Modus:byte);         Wahl des Hintergrund-Modus      *)
(*                                                               *)
(*   WinWrite(ASCII:char);           Zeichenausgabe im akt. Fenster  *)
(*   WinRead(VAR ASCII:char);        Zeichen aus dem akt. Fenster lesen *)
(*                                                               *)
(*   GetWin(VAR S1,Z1,S2,Z2:byte);   Abfrage der akt. Fenstergrenzen *)
(*                                                               *)
(*****************************************************************)


PROCEDURE TextInit;
  BEGIN
    inline($cd/$5a/$fc/$4e/$bb)
  END;

PROCEDURE TextReset;
  BEGIN
    inline($cd/$5a/$fc/$51/$bb)
  END;

PROCEDURE WinSelect(WindowNr:byte);
  BEGIN
    inline($3a/WindowNr/$cd/$5a/$fc/$b4/$bb)
  END;

PROCEDURE WinEnable(S1,Z1,S2,Z2:byte);
  BEGIN
    inline($3a/S1/$67/$3a/S2/$57/
           $3a/Z1/$6f/$3a/Z2/$5f/
           $cd/$5a/$fc/$66/$bb)
  END;

PROCEDURE WinSwap(WinNr1,WinNr2:byte);
  BEGIN
    inline($3a/WinNr1/$47/$3a/WinNr2/$4f/
           $cd/$5a/$fc/$b7/$bb)
  END;

PROCEDURE WinClear;
  BEGIN
    inline($cd/$5a/$fc/$6c/$bb)
  END;

PROCEDURE WinSetCursor(Zeile,Spalte:byte);
  BEGIN
    inline($3a/Zeile/$6F/$3a/Spalte/$67/
           $cd/$5a/$fc/$75/$bb)
  END;

PROCEDURE WinGetCursor(VAR Zeile,Spalte:byte);
  BEGIN
    inline($cd/$5a/$fc/$78/$bb/
           $7c/$32/Spalte/$7d/$32/Zeile)
  END;
```

```
(*****************************************************************)
(*                                                               *)
(*   WINDEMO.PAS                                                 *)
(*     Demonstrationsprogramm zur Textfestertechnik              *)
(*     CPC 6128  -  Turbo-Pascal  - CP/M 3.0                     *)
(*---------------------------------------------------------------*)
(*     benoetigt wird WINDOW.INC                                 *)
(*---------------------------------------------------------------*)
(*     zusammengestellt von  Hans-Werner Fromme  im Maerz  1987  *)
(*                                                               *)
(*****************************************************************)


PROGRAM windemo;

CONST text1 ='*** CPC6128  -  TURBO PASCAL  -  Fenster-Demonstration ***';
      text2 ='*** Bitte Taste druecken ***';
      text3 ='*** Ende der Vorstellung ***';

VAR chr:char;
      k,i:integer;

{$I window.inc }

BEGIN
  ClrScr;
  TextInit;

{***************************** Textfenster #0 ************************}
  WinSelect(0);
  WinFrameEnable(2,2,78,2);
  WinInvers;
  WinClear;
  WinSetCursor(1,1);
  FOR i:=1 TO length(text1) DO WinWrite(ord(copy(text1,i,1)));
  WinCursorOff;

{***************************** Textfenster #1 ************************}
  WinSelect(1);
  WinFrameEnable(2,21,78,21);
  WinInvers;
  WinClear;
  WinSetCursor(1,1);
  FOR i:=1 TO length(text2) DO WinWrite(ord(copy(text2,i,1)));
  WinCursorOff;
  read(kbd,chr);

{***************************** Textfenster #2 ************************}
  WinSelect(2);
  WinFrameEnable(2,5,39,18);
  WinClear;
  WinSetCursor(1,1);
  WinCursorOff;
  REPEAT
    FOR i:=33 to 255 DO WinWrite(i);
    i:=1;
  UNTIL Keypressed;
  WinSelect(1);
  delay(1000);
  WinClear;
  delay(1000);
  WinSelect(2);
  WinClear;

{***************************** Textfenster #3 ************************}
  WinSelect(3);
  WinEnable(44,5,76,18);
  WinInvers;
  WinClear;
  WinSetCursor(7,3);
  FOR i:=1 to length(text3) DO WinWrite(ord(copy(text3,i,1)));
  WinCursorOff;
  delay(3000);

  TextInit;
  ClrScr
END.
```

# CPC DATABOX

## — Das ist die Software zum CPC Magazin —

**DATABOX:**

— mehr als der übliche Softwareservice

— bringt alle sechzehn Programme des Sonderheftes auf Kassette oder 3''-Diskette

— Die Programme sind, durch den auf den Datenträger enthaltenen CPC-Emulator, auf allen drei CPC-Rechnern lauffähig.

— trägt das Titelbild des gleichzeitig erscheinenden Sonderheftes.

— über 500 KByte Software für Ihren CPC!




**Für alle CPC's als Kassette und 3''-Diskette.**

**Inhalt der Databox zu Sonderheft 5/87**

**Diskette/Kassette 1**

| | |
|---|---|
| PSG | Gbasic |
| Vier gewinnt | Poker |
| Convert | Disktool |
| Erweitertes DOS | Wertpapierbilanz |
| Envpro | Terminalprogramm |
| Secret Cavern | Tennis-Manager |
| Colfunkt | Morse-Trainer |
| Unilab | Mensch ärgere den CPC nicht |
| Zeitsignal | Slime (DEVPAC) |
| Erasto (DEVPAC) | Timer (DEVPAC) |
| Emulator | |

**Diskette/Kassette 2**

| | |
|---|---|
| Sounddemos | Schlumpi |
| Runner | Defprint |
| Crazystamps | Drucken mit dem 8.Bit |
| Tonjustage | Bundeslisga |
| Lister | Windows |
| Deluxe Data System | Timeout |
| Tilgung | Guazuela |
| Memory | Trace |
| Fehlereditor | Checksummer |

Einzelbezugspreise für DATABOX:
je Diskette 3'' 24,– DM zuzüglich 3,– DM Porto/Verpackung (im Ausland zuzüglich 5,– DM Porto/Verpackung).
Kombi-Pack (3" -Diskette I +II) nur 38,– DM

je Kassette 14,– DM zuzüglich 3,– DMPorto/Verpackung (im Ausland zuzüglich 5,– DM Porto/Verpackung).
Kombi-Pack (Kassette I +II) nur 25,– DM

**Bitte benutzen Sie den Bestellcoupon!**

DMV

**Daten & Medien Verlagsges. mbH**

**Postfach 250**

**3440 Eschwege**

Name: .....

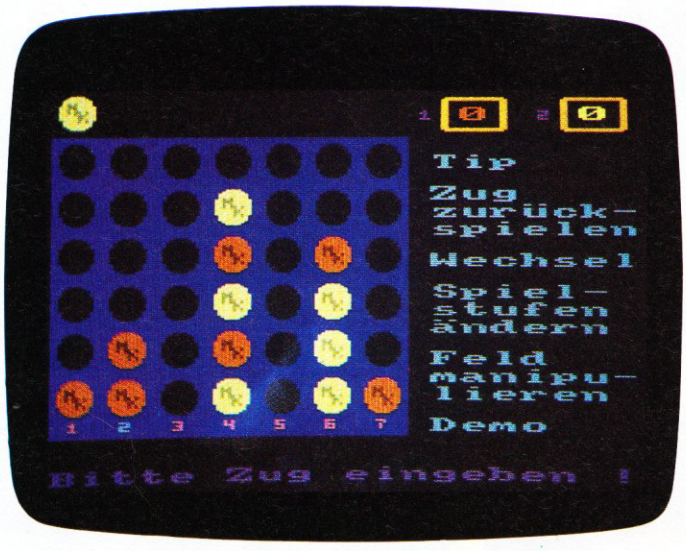
Anschrift: .....

Hiermit bestelle ich die Databox zum Sonderheft 5/87
Diskette I 24,– DM ☐ Diskette II 24,– DM
(beidseitig) (beidseitig)
Kombi-Pack Diskette I + II 38,– DM

Kassette I 14,– DM ☐ Kassette II 14,– DM
Kompi-Pack Kassette I + II 25,– DM

zuzüglich 3,– DM Porto/Verpackung
(im Ausland zuzüglich 5,– DM Porto/Verpackung)

bitte liefern Sie per NN
V-Scheck liegt bei




# VIER gewinnt

## *Sind Sie ein Liebhaber von Strategiespielen?*

Und suchen vielleicht einmal eine Abwechslung zum Schachcomputer? Dann ist VIER GEWINNT das ideale Programm für Sie, da es sich nicht nur durch eine hervorragende Grafik, sondern vor allem durch eine außerordentlich hohe Spielstärke auszeichnet (CPC 464/6128).

Vier Gewinnt ist ein beliebtes Strategiespiel mit einfachen Spielregeln: Ein 7 × 6 Felder großes Spielfeld wird senkrecht zwischen zwei Spielern aufgestellt. Der eine Spieler erhält rote, der andere gelbe Spielsteine. Sie werfen nun abwechselnd je einen ihrer Steine in einen der sieben Schächte des Spielfeldes (Aussetzen ist nicht erlaubt). Dieser Stein fällt dann bis ganz unten durch. Auf diese Weise werden die Schächte nach und nach aufgefüllt. Sieger ist, wer als erstes vier Steine in eine Reihe bekommt, senkrecht, waagrecht, oder diagonal. Dieser Spielgedanke eignet sich sehr gut zur Umsetzung in ein Computerprogramm, so daß solche auch schon oft vorgestellt wurden. Bisher spielten die Computer aber meist ziemlich schwach und waren so dem Menschen fast immer unterlegen. Es lag also nahe, einmal ein Programm zu schreiben, das den Rechner als ernstzunehmenden Gegner auftreten läßt und außerdem noch einige zusätzliche Funktionen bietet (z.B. Züge zurücknehmen oder Seitenwechsel), um so das Spiel interessanter zu gestalten. Um die große Anzahl der erforderlichen Rechenoperationen in absehbarer Zeit durchführen zu können, mußte die gesamte Strategie des Computers in Maschinensprache geschrieben werden. Das Programm arbeitet dabei nach dem Minimax-Prinzip: Zunächst werden alle Stellungen, die innerhalb einer gegebenen Anzahl von Zügen erreicht werden können, ermittelt und bewertet. Mit Hilfe der so gewonnenen Informationen kann dann der beste Zug herausgefunden werden. Besondere Sorgfalt wurde dabei auf den alles entscheidenden Stellungsbewerter gelegt, was allerdings auch zur Folge hatte, daß der Computer nicht wesentlich besser spielt, wenn er zwei Züge vorausberechnet (Spielstufe 1), als wenn er nur bis zum nächsten Zug weiter »denkt«. Erst bei drei Zügen (Spielstufe 2) tritt eine deutliche Verbesserung der Strategie ein. Neben der hohen Spielstärke sollte das Programm aber auch über eine anspruchsvolle Grafik verfügen. Besondere Schwierigkeiten bereitete dabei die Animation eines fallenden Spielsteins. Auch dafür war ein kleines Maschinenprogramm erforderlich, welches einen Stein auf dem Bildschirm darstellt, ohne dabei das Spielfeld zu überschreiben. Mit Hilfe dieses Programms war es dann möglich, den Eindruck zu erwecken, als würde ein Stein hinter dem Feld hinunterfallen und wäre nur durch die Löcher hindurch sichtbar. Doch überzeugen Sie sich selbst von der Spielweise des Computers. Werden Sie ihn schlagen?

### PROGRAMMABLAUF

Nach dem Start des Basic-Programms wird zunächst das Maschinenprogramm aus DATA-Zeilen in den Speicher gelesen. Mögliche Tippfehler werden durch ein doppeltes Prüfsystem mit ziemlicher Sicherheit erkannt – und die fehlerhafte Zeile angezeigt. Hat das Einlesen geklappt, erscheint das Titelbild auf dem Bildschirm, und Sie können wählen, mit welchen Steinen Sie spielen möchten, und ob Sie den ersten Zug machen wollen. Danach wird das Spielfeld angezeigt. Rechts am Bildschirm befindet sich eine Liste aller zusätzlichen Optionen, die Ihnen während des Spiels zur Verfügung stehen: TIP, ZUG ZURÜCK-SPIELEN, SEITEN-WECHSEL, SPIELSTUFEN ÄNDERN, FELD MANIPULIEREN und DEMO-SPIEL. Darüber können Sie die Spielstufen ablesen, in denen der Computer spielt. Die linke Stufe (Nr. 1) gibt dabei an, wie stark der Computer gegen Sie spielt, während die rechte (Nr. 2) die Spielstärke des zweiten (Computer-)Spielers im Demo-Spiel darstellt. Sie können also beim Demo-Spiel die Spieler mit zwei unterschiedlichen Spielstufen gegeneinander antreten lassen. In den untersten beiden Zeilen werden Ihnen alle im Verlauf eines Spiels wichtigen Informationen angezeigt. Jedesmal wenn Sie am Zug sind, erscheint ein Spielstein Ihrer Farbe über Schacht 1. Durch Drücken einer Zahl von 1 bis 7 können Sie ihn beliebig positionieren. Wenn Sie dann ENTER drücken, fällt der Stein in den zuletzt angewählten Schacht, und der Computer ist an der Reihe. Der zuletzt gemachte Zug wird jeweils dadurch gekennzeichet, daß die Zahl unter dem entsprechenden Schacht blau aufleuchtet.

Um die zusätzlichen Optionen aufzurufen, geben Sie, wenn Sie am Zug sind, anstatt einer Zahl den Anfangsbuchstaben des gewünschten Menüpunktes und anschließend ENTER ein.

**TIP**
Der Computer sagt Ihnen, welchen Zug er an Ihrer Stelle tun würde. Dabei wird die rechte Spielstufe (Nr. 2) verwendet. Betrachtet er mehrere Züge als gleichwertig, so zeigt er diese alle an. Ihr Spielstein wird gleich über dem angezeigten Schacht positioniert, Sie brauchen nur noch ENTER zu drücken (natürlich können Sie auch einen anderen Zug machen).

**Zug zurückspielen**
Der letzte Zug (blau markiert) wird zurückgenommen. Dazu wird der entsprechende Stein wieder vom Spielfeld entfernt. Diese Option kann sooft aufgerufen werden, bis sich das Spiel wieder in der Ausgangsstellung (in der Regel das leere Feld) befindet.

### Wechseln

Sie übernehmen die Steine des Computers, während er mit Ihren Steinen weiterspielt. Wenn nach dem Aufruf der beiden zuletzt genannten Menüpunkte nicht mehr Sie, sondern der Computer an der Reihe ist, wird dieser Fall angezeigt. Sie können dann zwar noch mehr Menüpunkte aufrufen (eventuell noch weitere Züge zurücknehmen, so daß wieder Sie an der Reihe sind), eine Zugeingabe wird jedoch ignoriert, an Ihrer Stelle macht dann der Computer seinen Zug.

### Spielstufen ändern

Sie können neue Werte für die oben angezeigten Spielstufen bestimmen. Geben Sie dazu jeweils eine Zahl von 0 bis 4, gefolgt von ENTER, ein. Die Spielstufe gibt, wie bereits erwähnt, an, wieviele Züge der Computer vorausdenkt.

### Feld manipulieren.

Diese Option erlaubt es Ihnen, während eines Spiels das Spielfeld beliebig »umzubauen«. Dazu erscheint unten links im Feld ein blinkendes Kreuz als Cursor, das Sie mit den Cursortasten bewegen können. Mit »R« und »G« wird auf Cursorposition ein roter bzw. gelber Stein gesetzt, mit »DEL« wieder weggenommen. »CLR« löscht das gesamte Spielfeld und setzt den Cursor wieder in Ausgangsposition. Mit »ENTER« können Sie Ihre Eingabe abschließen. Alle falsch positionierten Spielsteine werden dabei wieder entfernt. Das Spiel wird dann ganz normal fortgesetzt. Beachten Sie aber, daß das von Ihnen konstruierte Spielfeld immer eine Ausgangsstellung darstellt, von der aus keine Züge zurückgenommen werden können.

### Demo

Der (Computer-) Spieler 2 (Spielstufe Nr. 2) spielt nun an Ihrer Stelle weiter. Das Demo kann abgebrochen werden, indem Sie irgend eine Taste drücken, solange der entsprechende Text im unteren Fenster erscheint. Sie spielen dann mit den Steinen des Spielers weiter, der als nächstes am Zug gewesen wäre.

### Spielende

Ein Spiel ist beendet, wenn es einem Spieler gelingt, vier seiner Steine in eine Reihe zu bekommen, oder das ganze Spielfeld voll ist (in diesem Fall geht das Spiel unentschieden aus). Das Spielende wird auf dem Bildschirm angezeigt. Drücken Sie dann irgend eine Taste, um wieder zum Titelbild zu gelangen. Sie können ein laufendes Spiel auch abbrechen, in dem Sie zweimal die ESC-Taste betätigen. Auch dann kehren Sie wieder zum Titelbild zurück. Dieser Abbruch ist allerdings nicht möglich, solange der Text »Ich überlege...« angezeigt wird.

### VARIABLENLISTE

| | |
|---|---|
| a,b | Hilfsvariablen für unterschiedliche Zwecke |
| anf | Nummer des Spielers, der den ersten Zug machte |
| c | Kontrollvariable beim Einlesen des Maschinenprogramms / Hilfsvariable bei Zugeingabe |
| dx, dy | Richtungsangaben für x- u. y-Koordinaten |
| fd | Flag für DEMO-Spiel |
| fl | gibt an, ob ein bzw. welcher Spieler gewonnen hat / z.T. auch Flag für verschiedene Zwecke |
| fm | Flag, ob Feld manipuliert wurde |
| h | Hintergrundfarbe bei Ausgabe eines Menuepunktes (vgl. m,v) |
| i | Schleifenvariable / Hilfsvariable für Tastaturabfr. |
| j | Schleifenvariable |
| k | Kontrollsumme (richtiger Wert) beim Einlesen des Maschinenprogramms (vgl. s) |
| m | gibt an, welcher Menuepunkt (=Option) ausgegeben werden soll (vgl. h,v) |
| o | Adressvariable beim Einlesen des Maschinenprogramms |
| q | Nummer des Spielers, der gerade an der Reihe ist |
| s | Kontrollsumme (tatsächlicher Wert) beim Einlesen des Maschinenprgs. (vgl. k) |
| t! | Hilfsvariable für Warteschleifen |
| v | Vordergrundfarbe bei Ausgabe eines Menuepunktes (vgl. h,m) |
| x | x-Koordinate / Bit-Maske bei Übergabe des Spielfeldes an das Maschinenprg. |
| y | y-Koordinate |
| z | Zug (Schachtnr.) / Anzahl gültiger Steine nach »Feld manipulieren« / Zeilernnr. bei Maschinenprg. einlesen |
| zu | Zugzähler (vgl. z(x)) |
| a$ | Hilfsvariable für unterschiedliche Zwecke |
| i$ | Hilfsvariable für Tastaturabfrage |
| s$ | Hilfsvariable zum Einlesen der SYMBOL-Daten |
| c(x) | Farbe des Spielers Nr. x |
| f(x,y) | Spielfeld |
| h(x) | Anzahl Steine im Schacht x |
| z(x) | Züge seit Ausgangsstellung, wichtig für »Zug zurücknehmen« (vgl. zu) |
| m$(x,y)y. | Zeile des x. Menüpunktes |

(Michael Knaup)

## für 464-664-6128



```
10 REM %%%%
```

```
370 a=INSTR(a$,"["):IF a THEN a$=LEFT$(a$, [4245]
a-1)+CHR$(253)+MID$(a$,a+1)
380 a=INSTR(a$,"]"):IF a THEN a$=LEFT$(a$, [3986]
a-1)+CHR$(255)+MID$(a$,a+1)
390 m$(i,j)=a$:NEXT j,i                    [1364]
400 s(1)=PEEK(&A011):s(2)=PEEK(&A012)      [2237]
410 IF PEEK(&A010) THEN RESTORE 3160 ELSE  [2485]
RESTORE 3170
420 MODE 1:FOR j=0 TO 3:READ a:INK j,a:NEX [3613]
T:BORDER 0
430 PAPER 0:PEN 2:LOCATE 8,3:PRINT"V I E R [3666]
   G E W I N N T"
440 PEN 1:LOCATE 17,5:PRINT CHR$(164);1986 [1772]
450 PEN 3:LOCATE 12,7:PRINT"by Michael Kna [3264]
up"
460 PEN 1:LOCATE 1,14:PRINT"Wollen Sie mit [9084]
 den ";:PEN 2:PRINT"roten";:PEN 1:PRINT" o
der ";:PEN 3:PRINT"gelben":PEN 1:PRINT"Ste
inen spielen ?"
470 PRINT CHR$(2);CHR$(10);"[r/g] ?  ";CHR [4447]
$(143);CHR$(8);
480 i$="":WHILE i$<>"r" AND i$<>"g":i$=LOW [4658]
ER$(INKEY$):WEND
490 IF i$="r" THEN c(1)=4:c(2)=2:PRINT"rot [4155]
" ELSE c(1)=2:c(2)=4:PRINT"gelb"
500 PEN 1:LOCATE 1,21:PRINT"Wollen Sie den [4138]
 ersten Zug machen ?"
510 PRINT:PRINT"[j/n] ?  ";CHR$(143);CHR$( [2063]
8);
520 i$="":WHILE i$<>"j" AND i$<>"n":i$=LOW [2205]
ER$(INKEY$):WEND
530 IF i$="j" THEN PRINT"ja":anf=2 ELSE PR [2758]
INT"nein":anf=1
540 t!=TIME+300:WHILE t!>TIME:WEND         [2517]
550 :                                      [174]
560 REM %%%%%%%%%%%%%                     [955]
570 REM % Spielfeld %                      [1792]
580 REM %%%%%%%%%%%%%                     [955]
590 :                                      [174]
600 MODE 0:FOR j=0 TO 15                   [1828]
610 READ a:IF a<0 THEN a=-a:READ b ELSE b= [2326]
a
620 INK j,a,b:NEXT                         [1752]
630 ORIGIN 0,0,0,388,48,350:CLG 15:ORIGIN  [2400]
0,0,0,640,0,400
640 FOR i=6 TO 1 STEP-1:FOR j=1 TO 7       [1248]
650 CALL &9804,FNx(j),FNy(i),256           [1652]
660 NEXT j,i                               [396]
670 z=0:GOSUB 2550                         [1027]
680 a$=CHR$(150)+CHR$(154)+CHR$(156):PEN 1 [3939]
0:LOCATE 14,1:PRINT a$;" ";a$
690 a$=CHR$(149):LOCATE 14,2:PRINT a$;" "; [4067]
a$;" ";a$;" ";a$
700 a$=CHR$(147)+CHR$(154)+CHR$(153):LOCAT [3381]
E 14,3:PRINT a$;" ";a$
710 PEN 11:LOCATE 13,2:PRINT CHR$(246);:LO [4482]
CATE 17,2:PRINT CHR$(247);
720 WINDOW 1,20,24,25:PRINT CHR$(2);       [2427]
730 WINDOW#1,15,15,2,2:PAPER#1,0:PEN#1,c(1 [3595]
):PRINT#1,HEX$(s(1));
740 WINDOW#2,19,19,2,2:PAPER#2,0:PEN#2,c(2 [2520]
):PRINT#2,HEX$(s(2));
750 PAPER#3,0:PEN#3,1                      [1775]
760 h=0:v=1:FOR m=1 TO 6:GOSUB 2670:NEXT   [2461]
770 :                                      [174]
780 REM %%%%%%%%%%%%%%%%%                 [1043]
790 REM % Hauptprogramm %                  [1350]
800 REM %%%%%%%%%%%%%%%%%                 [1043]
810 :                                      [174]
820 q=anf:zu=0:x=0:y=0:dx=0:dy=0:fl=-1     [2223]
830 WHILE fl=-1                            [1194]
840 ON q GOSUB 2160,1010                   [906]
850 GOSUB 880                              [1060]
860 q=q XOR 3:WEND                         [1264]
870 GOTO 2400                              [347]
880 :                                      [174]
890 REM %%%%%%%%%%%%%%%%%                 [1169]
900 REM % Zug ausfuehren %                 [2036]
910 REM %%%%%%%%%%%%%%%%%                 [1169]
920 :                                      [174]
930 h(z)=h(z)+1:f(z,h(z))=q:zu=zu+1:z(zu)= [4633]
z
940 FOR j=FNy(7) TO FNy(h(z)) STEP-2       [2892]
950 CALL &9804,FNx(z),j,c(q)               [1419]
960 NEXT:GOSUB 2550                        [1190]
970 z=&A100:FOR i=1 TO 7:FOR j=1 TO 8      [1681]
980 POKE z,f(i,j):z=z+1:NEXT j,i           [1293]
990 CALL &9802,@fl,@x,@y,@dx,@dy           [1460]
1000 RETURN                                [555]
1010 :                                     [174]
1020 REM %%%%%%%%%%%%%%%                  [947]
1030 REM % Spieler Zug %                   [2169]
1040 REM %%%%%%%%%%%%%%%                  [947]
1050 :                                     [174]
1060 c=1:fd=0:CLS:PEN 11:PRINT CHR$(7);CHR [6777]
$(10);"Bitte Zug eingeben !";
1070 IF c<8 THEN CALL &9804,FNx(c),FNy(7), [4661]
c(2) ELSE m=c-7:h=1:v=3:GOSUB 2670
1080 i$="":WHILE i$="":i$=UPPER$(INKEY$):W [1626]
END
1090 IF c<8 THEN CALL &9804,FNx(c),FNy(7), [5860]
0 ELSE IF i$<>CHR$(13) THEN m=c-7:h=0:v=1:
GOSUB 2670
1100 IF i$=CHR$(13) THEN 1190              [1419]
1110 IF i$>"0" AND i$<"8" THEN c=VAL(i$)   [2461]
1120 IF i$="T" THEN c=8                    [1781]
1130 IF i$="Z" THEN c=9                    [1719]
1140 IF i$="W" THEN c=10                   [704]
1150 IF i$="S" THEN c=11                   [1066]
1160 IF i$="F" THEN c=12                   [580]
1170 IF i$="D" THEN c=13                   [1255]
1180 GOTO 1070                             [317]
1190 IF c>7 THEN ON c-7 GOTO 1230,1350,149 [2264]
0,1620,1890,1750
1200 IF q=1 THEN 2160                      [1188]
1210 IF h(c)=6 THEN PEN 2:CLS:PRINT CHR$(7 [5197]
);CHR$(10);"Unerlaubter Zug.":GOTO 1070
1220 z=c:RETURN                            [1360]
1230 :                                     [174]
1240 REM  ***                              [413]
1250 REM  *** Tip                          [573]
1260 REM  ***                              [413]
1270 :                                     [174]
1280 IF q=1 THEN PRINT CHR$(30);CHR$(7);:G [3043]
OTO 1080
1290 GOSUB 2260                            [871]
1300 CLS:PEN 11:PRINT CHR$(7);CHR$(10);"Me [1434]
in Tip: ";
1310 PEN c(2):FOR j=1 TO PEEK(&A001)       [1695]
1320 PRINT HEX$(PEEK(&A001+j));:IF j<>PEEK [3610]
(&A001) THEN PRINT",";
1330 NEXT:i$=CHR$(z+48)                    [855]
1340 GOTO 1090                             [393]
1350 :                                     [174]
1360 REM  ***                              [413]
1370 REM  *** Zug zurueckspielen           [651]
1380 REM  ***                              [413]
1390 :                                     [174]
1400 CLS:PEN 2                             [491]
1410 IF zu=0 THEN PRINT CHR$(7);"Spiel ber [5772]
eits in    Ausgangsstellung.":GOTO 1080
1420 z=z(zu):zu=zu-1                       [2088]
1430 FOR j=FNy(h(z)) TO FNy(7) STEP 2      [2104]
1440 CALL &9804,FNx(z),j,c(f(z,h(z)))      [1828]
1450 NEXT:t!=TIME+150:WHILE t!>TIME:WEND   [4108]
1460 CALL &9804,FNx(z),FNy(7),0            [1853]
```

```
1470 f(z,h(z))=0:h(z)=h(z)-1:q=q XOR 3      [2409]
1480 z=z(zu):GOSUB 2550:GOTO 1580           [2168]
1490 :                                      [174]
1500 REM  ***                               [413]
1510 REM  *** Wechsel                       [658]
1520 REM  ***                               [413]
1530 :                                      [174]
1540 FOR i=1 TO 6:FOR j=1 TO 7:f(j,i)=(f(j  [4433]
,i) XOR 3)*SGN(f(j,i)):NEXT j,i
1550 a=c(1):c(1)=c(2):c(2)=a:anf=anf XOR 3  [2943]
:q=q XOR 3
1560 PEN#1,c(1):PRINT#1,HEX$(s(1));         [1815]
1570 PEN#2,c(2):PRINT#2,HEX$(s(2));         [1602]
1580 CLS:IF q=2 THEN IF fd THEN 1010 ELSE   [2130]
1730
1590 PEN 2:PRINT CHR$(7);"Achtung:"         [1982]
1600 PEN 11:PRINT"Ich bin am Zug."          [2757]
1610 GOTO 1080                              [387]
1620 :                                      [174]
1630 REM  ***                               [413]
1640 REM  *** Spielstufen aendern           [999]
1650 REM  ***                               [413]
1660 :                                      [174]
1670 FOR j=1 TO 2                           [615]
1680 CLS:PEN c(j):PRINT CHR$(10);"Stufe";j  [4905]
;"[0-4]:  ";:PEN 11:PRINT HEX$(s(j));CHR$(
8);
1690 i$="":WHILE (i$<"0" OR i$>"4") AND i$  [3812]
<>CHR$(13):i$=INKEY$:WEND
1700 IF i$<>CHR$(13) THEN PRINT i$;CHR$(8)  [3481]
;:s(j)=VAL(i$):GOTO 1690
1710 POKE &A010+j,s(j):PRINT#j,HEX$(s(j));  [1984]
1720 NEXT                                   [350]
1730 CLS:PEN 10:PRINT"O.k.":PEN 11:PRINT"B  [3972]
itte Zug eingeben !";
1740 GOTO 1080                              [387]
1750 :                                      [174]
1760 REM  ***                               [413]
1770 REM  *** Demo                          [757]
1780 REM  ***                               [413]
1790 :                                      [174]
1800 fd=-1                                  [399]
1810 IF zu<2 AND fm=0 THEN z=INT(RND*7)+1   [3555]
ELSE GOSUB 2260
1820 CLS:PEN 2:PRINT"Taste":PEN 11:PRINT C  [3847]
HR$(243);" Abbruch Demo";
1830 GOSUB 880:CLS                          [1547]
1840 IF fl>-1 THEN 2400                     [921]
1850 q=q XOR 3:IF INKEY$="" THEN 1810       [1790]
1860 m=6:h=0:v=1:GOSUB 2670                 [2245]
1870 IF q=1 THEN 1490                       [1146]
1880 GOTO 1010                              [345]
1890 :                                      [174]
1900 REM  ***                               [413]
1910 REM  *** Feld manipulieren             [2316]
1920 REM  ***                               [413]
1930 :                                      [174]
1940 x=1:y=1:z=0:GOSUB 2550:CLS:TAG:PLOT 0  [9227]
,36,11:CALL &BBE4:a$="":FOR j=240 TO 243:a
$=a$+CHR$(j)+",":NEXT:PRINT LEFT$(a$,7);:T
AGOFF:PEN 11:PRINT CHR$(10);"R,G,DEL,CLR,E
NTER";
1950 MOVE x*56-32,y*48+46:DRAWR 0,-14,14:M  [6419]
OVER 4,14:DRAWR 0,-14:MOVE x*56-44,y*48+38
:DRAWR 30,0:MOVER -30,2:DRAWR 30,0
1960 i=0:WHILE (i<240 OR i>243) AND i<>71   [7357]
AND i<>82 AND i<>127 AND i<>13 AND i<>16:i
$="":WHILE i$="":i$=UPPER$(INKEY$):WEND:i=
ASC(i$):WEND
1970 IF i=71 THEN IF c(1)=4 THEN f(x,y)=1   [1805]
ELSE f(x,y)=2
1980 IF i=82 THEN IF c(1)=2 THEN f(x,y)=1   [2319]
ELSE f(x,y)=2
1990 IF i=127 THEN f(x,y)=0                 [1687]
2000 CALL &9804,FNx(x),FNy(y),c(f(x,y))     [1630]
2010 IF i=240 THEN y=y+1:IF y>6 THEN y=1    [2821]
2020 IF i=241 THEN y=y-1:IF y<1 THEN y=6    [2255]
2030 IF i=242 THEN x=x-1:IF x<1 THEN x=7    [1284]
2040 IF i=243 THEN x=x+1:IF x>7 THEN x=1    [1024]
2050 IF i=16 THEN FOR i=6 TO 1 STEP-1:FOR   [6060]
j=1 TO 7:f(j,i)=0:CALL &9804,FNx(j),FNy(i)
,0:NEXT j,i:x=1:y=1
2060 IF i<>13 THEN 1950                     [908]
2070 FOR i=1 TO 7:fl=0:h(i)=0:z=0:FOR j=1   [2628]
TO 6
2080 IF fl THEN f(i,j)=0                    [335]
2090 IF f(i,j)=0 THEN fl=-1 ELSE z=z+1:h(i  [2523]
)=j
2100 CALL &9804,FNx(i),FNy(j),c(f(i,j))     [2443]
2110 NEXT j,i                               [396]
2120 q=2:zu=0:anf=2-(z MOD 2)               [2688]
2130 TAGOFF:m=5:h=0:v=1:GOSUB 2670:fm=-1    [2103]
2140 LOCATE#3,1,23:PRINT#3,SPACE$(10);      [2377]
2150 GOTO 1010                              [345]
2160 :                                      [174]
2170 REM %%%%%%%%%%%%%%%%                  [850]
2180 REM % Computer Zug %                   [516]
2190 REM %%%%%%%%%%%%%%%%                  [850]
2200 :                                      [174]
2210 IF zu=0 AND fm=0 THEN z=INT(RND*7)+1   [3555]
ELSE GOSUB 2260
2220 IF s(1)<2 THEN RETURN                  [930]
2230 PRINT CHR$(7);:CALL &9804,FNx(z),FNy(  [2852]
7),c(1)+4
2240 t!=TIME+600:WHILE t!>TIME:WEND         [2453]
2250 RETURN                                 [555]
2260 :                                      [174]
2270 REM %%%%%%%%%%%%%%%%%%%%%%%%          [1394]
2280 REM % besten Zug ermitteln %           [1322]
2290 REM %%%%%%%%%%%%%%%%%%%%%%%%          [1394]
2300 :                                      [174]
2310 IF q=1 THEN x=0 ELSE x=3               [1524]
2320 z=&A100:FOR i=1 TO 7:FOR j=1 TO 8      [1681]
2330 POKE z,(f(i,j) XOR x)*SGN(f(i,j))      [775]
2340 z=z+1:NEXT j,i                         [1047]
2350 POKE &A000,s(q):POKE &A138,anf XOR x   [1761]
2360 CLS:PEN c(q):PRINT CHR$(10);"Ich ";CH  [4111]
R$(255);"berlege...";
2370 CALL &9800                             [477]
2380 z=PEEK(&A002+INT(RND*PEEK(&A001)))     [3154]
2390 CLS:RETURN                             [832]
2400 :                                      [174]
2410 REM %%%%%%%%                          [701]
2420 REM % Ende %                           [551]
2430 REM %%%%%%%%                          [701]
2440 :                                      [174]
2450 CLS:PRINT CHR$(7);:t!=TIME+90:WHILE t  [3647]
!>TIME:WEND:PRINT CHR$(7);
2460 IF fl=0 THEN PEN 10:PRINT"Unentschied  [4147]
en.":GOTO 2530
2470 PEN c(fl):IF fd THEN PRINT HEX$(fl);"  [4877]
 hat gewonnen.":GOTO 2500
2480 IF fl=1 THEN PRINT"Ich habe gewonnen.  [2798]
"
2490 IF fl=2 THEN PRINT"Sie haben gewonnen  [1764]
."
2500 FOR j=1 TO 4                           [621]
2510 CALL &9804,FNx(x),FNy(y),c(fl)+4       [2075]
2520 x=x+dx:y=y+dy:NEXT                     [1390]
2530 WHILE INKEY$<>"":WEND:WHILE INKEY$=""  [3680]
:WEND
2540 GOTO 270                               [415]
2550 :                                      [174]
2560 REM %%%%%%%%%%%%%%%%%%                [1169]
```

```
2570 REM % Unterprogramme %                [1739]
2580 REM %%%%%%%%%%%%%%%%%%               [1169]
2590 :                                     [174]
2600 ORIGIN 0,0,0,388,48,48:CLG 15:ORIGIN  [2396]
0,0,0,640,0,400
2610 TAG:FOR j=0 TO 6                      [440]
2620 IF z=j+1 THEN PLOT -10,0,13 ELSE PLOT [2160]
 -10,0,12
2630 MOVE j*56,66:PRINT CHR$(246+j);       [1324]
2640 NEXT:TAGOFF                           [893]
2650 RETURN                                [555]
2660 :                                     [174]
2670 RESTORE 3190:FOR j=1 TO m:READ a,b:NE [2518]
XT
2680 IF h=0 THEN CALL &BBE4 ELSE CALL &BBE [2660]
4,0
2690 LOCATE 1,1:TAG:FOR j=1 TO a           [876]
2700 MOVE 412,b-(j-1)*18:DRAWR MAX(LEN(m$( [4367]
m,j)),LEN(m$(m,j-1)))*32,0,h
2710 PLOT 416,b-(j-1)*18-2,v:PRINT m$(m,j) [2959]
;
2720 MOVE 412,b-(j-1)*18-2:IF j=a THEN DRA [2439]
WR 0,-16,h ELSE DRAWR 0,-18,h
2730 NEXT:TAGOFF                           [893]
2740 MOVE 416,b-a*18:DRAWR LEN(m$(m,a))*32 [3103]
-4,0,h
2750 RETURN                                [555]
2760 :                                     [174]
2770 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   [1672]
2780 REM % Maschinenprogramm einlesen %    [1748]
2790 REM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   [1672]
2800 :                                     [174]
2810 RESTORE 3200:DEFINT a-z               [1432]
2820 PRINT:PRINT"Bitte warten !"           [1820]
2830 PRINT"Maschinenprogramm wird eingeles [3158]
en."
2840 z=9020:FOR i=0 TO 4:FOR j=0 TO 15     [1699]
2850 READ a$:a=VAL("&"+a$):s=s+a           [896]
2860 POKE &9700+16*i+j,a:NEXT              [2328]
2870 READ k:IF s<>k THEN 2990              [1695]
2880 NEXT                                  [350]
2890 z=9110:o=&9800:c=0                    [910]
2900 FOR i=0 TO 78:s=0:FOR j=0 TO 15       [1057]
2910 READ a$:a=VAL("&"+a$):s=s+a           [896]
2920 POKE o+j,a:NEXT                       [968]
2930 CALL &9700,o,16,@c                    [1162]
2940 READ a$:IF VAL("&"+a$)<>c THEN 2990   [1705]
2950 READ k:IF s<>k THEN 2990              [1695]
2960 o=o+16:NEXT                           [908]
2970 PRINT" O.k.":RESTORE                  [985]
2980 RETURN                                [555]
2990 z=z+10*i                              [684]
3000 PRINT:PRINT"Fehler in Zeile";z        [2519]
3010 PRINT"Bitte korrigieren !"            [2983]
3020 PRINT:CLEAR                           [557]
3030 END                                   [110]
3040 :                                     [174]
3050 REM %%%%%%%%%                        [598]
3060 REM % Daten %                         [708]
3070 REM %%%%%%%%%                        [598]
3080 :                                     [174]
3090 DATA 010,110,010,010,111,111,001,111, [2315]
100,111,111,001,011,001,111
3100 DATA 101,101,111,001,001,111,100,111, [2780]
001,111,111,100,111,101,111
3110 DATA 111,001,010,010,010              [1094]
3120 :                                     [174]
3130 DATA Tip,,,Zug,zur]ck-,spielen,Wechse [2823]
l,,,Spiel-,stufen,[ndern
3140 DATA Feld,manipu-,lieren,Demo,,        [965]
3150 :                                     [174]
3160 DATA 0,13,6,24,0,13,6,3,24,12,-6,0,-3 [3664]
,0,-24,0,-12,0,15,4,7,14,-26,0,1
3170 DATA 0,18,9,26,0,20,9,5,24,19,-9,0,-5 [3112]
,0,-24,0,-19,0,15,11,26,6,-26,0,14
3180 :                                     [174]
3190 DATA 1,334,3,302,1,234,3,202,3,134,1, [1903]
66
3200 :                                     [174]
3210 REM %%%%%%%%%%%%%%%%%%%%%            [1239]
3220 REM % Maschinenprogramm %             [1202]
3230 REM %%%%%%%%%%%%%%%%%%%%%            [1239]
3240 :                                     [174]
3250 REM prc                               [363]
3260 DATA FE,03,C0,DD,6E,00,DD,66,01,DD,4E [1755]
,02,DD,46,03,DD,1920
3270 DATA 5E,04,DD,56,05,7E,23,66,6F,C5,1A [3205]
,4F,06,08,CB,11,3240
3280 DATA 1F,AC,E6,80,28,09,7C,EE,08,67,7D [2488]
,EE,10,6F,37,ED,5105
3290 DATA 6A,10,EB,13,C1,0B,79,B0,20,DF,DD [3808]
,5E,00,DD,56,01,6860
3300 DATA EB,73,23,72,C9,00,00,00,00,00,00 [1189]
,00,00,00,00,00,7560
3310 :                                     [174]
3320 REM $9800-$9CEF                       [771]
3330 DATA 18,0A,18,03,C3,C1,9B,21,1F,98,18 [4155]
,03,21,79,98,E5,337C,1382
3340 DATA F3,08,D9,E1,C5,F5,CD,53,9C,F1,C1 [2228]
,D9,08,FB,C9,FD,9050,2943
3350 DATA 21,00,A1,DD,E5,CD,B9,99,DD,E1,DD [3844]
,6E,08,DD,66,09,7F45,2304
3360 DATA 77,23,77,3D,F8,36,00,79,CD,6C,98 [3075]
,14,1C,DD,6E,04,BB38,1605
3370 DATA DD,66,05,7B,CD,66,98,DD,6E,06,DD [3454]
,66,07,7A,CD,66,B84F,2006
3380 DATA 98,78,CD,6C,98,DD,6E,00,DD,66,01 [3366]
,7B,CD,66,98,DD,2C58,2195
3390 DATA 6E,02,DD,66,03,7A,77,23,07,9F,77 [3376]
,C9,F5,E6,07,5F,DC99,1777
3400 DATA F1,CB,2F,CB,2F,CB,2F,57,C9,FD,21 [2137]
,00,A1,FD,36,39,1182,2090
3410 DATA 01,AF,32,01,A0,21,CB,98,CD,48,9B [3540]
,3A,01,A0,B7,C0,9B6B,1801
3420 DATA CD,EC,98,21,00,00,E5,21,40,A1,01 [3482]
,01,07,FD,CB,3C,D969,1638
3430 DATA 06,30,21,5E,23,56,2B,E3,EB,CD,4C [4047]
,9C,EB,E3,38,14,7537,1782
3440 DATA 3A,01,A0,28,01,AF,E5,3C,32,01,A0 [3251]
,3C,6F,26,A0,71,46F7,1417
3450 DATA E1,EB,E3,EB,23,23,0C,10,D4,E1,C9 [4726]
,FE,03,C0,7B,CD,4DBD,2435
3460 DATA 37,9C,D8,3E,F8,A3,0F,0F,0F,3C,4F [2479]
,3A,01,A0,B7,28,1B5B,1526
3470 DATA 03,7C,B7,C8,21,01,A0,36,01,23,71 [3356]
,C9,CD,B9,99,B7,7C15,1834
3480 DATA FA,03,99,FD,36,3C,00,CA,DF,99,D6 [3269]
,02,FD,77,3A,FD,8523,2250
3490 DATA 77,3B,C9,FD,E5,E1,11,40,00,19,E5 [1424]
,1E,10,19,FD,56,3133,1831
3500 DATA 39,1E,00,06,07,C5,D5,FD,E5,E3,5D [2098]
,54,01,50,00,FD,9B94,1730
3510 DATA 7E,39,EE,03,09,FD,09,EB,0E,39,ED [4593]
,B0,12,E1,D1,0E,3955,1880
3520 DATA 07,7E,23,0D,28,23,B7,20,F8,2B,72 [2905]
,23,0D,20,FC,23,4054,1243
3530 DATA 23,37,CB,13,3A,00,A0,E5,D5,B7,20 [2834]
,12,CD,DF,99,D1,4693,1995
3540 DATA E1,FD,4E,3A,FD,46,3B,18,45,CB,03 [3727]
,23,18,40,3D,32,FEDE,1529
3550 DATA 00,A0,CD,EC,98,21,00,A0,34,FD,7E [3499]
,3C,B7,28,E0,57,7287,1971
3560 DATA FD,7E,39,4F,D6,02,2F,67,6F,7A,06 [2912]
,07,FD,E5,E3,11,AB72,1853
```

```
3570 DATA 40,00,19,5E,23,56,23,07,30,0F,E3 [2197]
,08,CD,4C,9C,CB,FFA6,1284
3580 DATA 41,20,01,3F,30,01,EB,E3,08,10,E8 [2337]
,C1,D1,E1,DD,E1,F1FA,2001
3590 DATA E3,71,23,70,23,E3,01,B0,FF,FD,09 [3585]
,DD,E5,C1,05,C2,F270,2285
3600 DATA 15,99,CB,03,FD,73,3C,E1,C9,AF,08 [4290]
,21,D2,99,CD,48,B3F7,2090
3610 DATA 9B,08,D9,B7,C0,FD,E5,E1,06,38,86 [3025]
,23,10,FC,FE,3F,EB9A,2278
3620 DATA 9F,C9,FE,04,C0,7A,08,D1,E1,C1,C5 [2519]
,E5,D5,D9,C9,21,9B91,2657
3630 DATA 00,80,11,54,9C,AF,F5,CD,43,9C,28 [2531]
,11,47,1A,E6,0F,348C,1632
3640 DATA 07,D5,16,00,05,28,03,ED,44,15,5F [3995]
,19,D1,F1,13,C6,4868,1403
3650 DATA 08,FE,36,38,E1,E6,07,3C,FE,06,38 [3359]
,DA,08,D9,21,C3,722A,1881
3660 DATA 9A,CD,48,9B,FD,E5,E1,16,00,06,07 [2313]
,C5,3E,20,A2,57,D39C,1868
3670 DATA 0E,00,7E,E6,03,28,05,23,0C,14,18 [3253]
,F6,79,FE,06,28,E496,1176
3680 DATA 1A,7E,B7,28,10,06,02,07,F5,DC,64 [2875]
,9A,F1,05,07,DC,B900,1598
3690 DATA 64,9A,7E,B2,57,23,0C,CB,DA,18,E1 [4284]
,23,23,C1,10,CB,B4D0,1844
3700 DATA FD,E5,E1,06,38,7E,E6,03,77,23,10 [3210]
,F9,D9,FD,75,3A,4B39,2192
3710 DATA FD,74,3B,C9,CB,5A,20,12,FD,7E,39 [3540]
,B8,1E,04,28,42,BB71,1732
3720 DATA CB,6A,20,3E,1E,00,CB,EA,18,38,2B [3125]
,7E,23,07,07,A0,6B28,1328
3730 DATA 28,1F,7A,CB,40,20,01,07,07,38,16 [3025]
,1E,03,CB,40,28,94CA,925
3740 DATA 21,7A,E6,07,28,1C,D9,11,E8,03,19 [3338]
,3D,20,FC,D9,18,FA8B,1540
3750 DATA 11,CB,62,1E,01,20,0B,FD,7E,38,A8 [2901]
,A9,EE,01,E6,01,EA90,1634
3760 DATA 3C,5F,7B,FE,02,38,02,CB,E2,CB,23 [3005]
,78,1F,83,C6,0E,B713,1753
3770 DATA 5F,18,15,FE,03,28,75,30,02,05,C8 [3022]
,FE,02,28,1B,E6,D67D,1362
3780 DATA 04,0F,5F,CB,42,20,01,1C,7B,D9,EB [3170]
,07,21,7E,9C,4F,B3B4,1420
3790 DATA 06,00,09,4E,23,46,EB,09,D9,C9,06 [2653]
,01,3A,20,A0,CD,9DB7,1322
3800 DATA 37,9C,38,05,04,3A,21,A0,5F,3A,21 [3149]
,A0,CD,37,9C,38,CAAB,1345
3810 DATA 05,04,3A,20,A0,5F,4B,5A,05,20,06 [3428]
,3E,05,83,5F,18,0F8D,879
3820 DATA C7,05,20,10,FD,7E,38,A9,AB,EE,01 [3881]
,E6,01,07,C6,07,7577,1709
3830 DATA 83,5F,18,B4,7B,FD,BE,39,28,09,08 [2365]
,3F,3F,08,C6,03,0F46,1445
3840 DATA 5F,18,A5,08,30,F6,08,C6,0B,5F,18 [3403]
,9C,7A,0F,0F,FD,33E1,1483
3850 DATA E5,E1,16,00,19,B6,77,C9,11,54,9C [2401]
,0E,00,1A,06,F9,E8CB,1555
3860 DATA 07,DC,78,9B,06,01,07,DC,78,9B,06 [3213]
,09,07,DC,78,9B,1659,1528
3870 DATA 06,08,07,DC,78,9B,13,3E,08,81,4F [2765]
,FE,36,38,DE,E6,CB8C,1629
3880 DATA 07,3C,4F,FE,06,38,D6,C9,F5,D5,C5 [3124]
,E5,21,00,00,DD,F430,2015
3890 DATA 21,20,A0,1E,04,79,CD,43,9C,FE,01 [2520]
,30,06,51,DD,71,FF64,1532
3900 DATA 00,DD,23,20,02,24,37,38,01,2C,79 [3350]
,80,4F,1D,20,E5,7EEA,1100
3910 DATA 5A,7C,FE,01,9F,57,7D,FE,01,9F,AA [2494]
,28,0F,16,01,AF,6052,1677
3920 DATA B4,20,01,14,B5,DD,E1,DD,E5,CD,52 [2225]
,9C,E1,C1,D1,F1,5019,2621
```



```
3930 DATA C9,DD,7E,00,CD,2C,BC,E6,55,32,20 [2157]
,A0,DD,6E,02,DD,A7E8,2096
3940 DATA 66,03,DD,5E,04,DD,56,05,CD,1D,BC [3382]
,11,AE,9C,06,16,0B12,1533
3950 DATA C5,E5,CD,ED,9B,E1,C1,CD,26,BC,10 [3362]
,F4,C9,01,C0,05,A554,2531
3960 DATA 7E,E5,F5,0F,CD,05,9C,07,6F,F1,CD [3100]
,05,9C,B5,E1,77,7BE3,2231
3970 DATA 23,10,ED,13,C9,E6,55,DD,CB,01,46 [2768]
,20,1C,FE,55,28,E71C,1757
3980 DATA 0D,1A,A1,3A,20,A0,28,06,CB,F7,EA [3073]
,1E,9C,AF,FE,40,6704,1859
3990 DATA 28,FB,CB,09,CB,09,D0,13,C9,E5,6F [2630]
,1A,A1,7D,E1,E2,2FC1,2246
4000 DATA 1E,9C,3A,20,A0,18,E7,4F,E6,07,C8 [2600]
,79,3D,CD,43,9C,13F0,1817
4010 DATA C0,37,C9,32,48,9C,FD,7E,00,E6,03 [3482]
,C9,7C,92,C0,7D,D845,2126
4020 DATA 93,C9,DD,E9,72,73,74,F6,C4,C3,C2 [3625]
,72,74,76,F8,C6,BCCA,2772
4030 DATA C4,C2,72,75,78,FA,C8,C5,C2,12,15 [2392]
,18,1A,08,05,02,1529,1686
4040 DATA 12,14,16,18,06,04,02,12,13,14,16 [3037]
,04,03,02,03,00,D5EE,187
4050 DATA FD,FF,4C,1D,B4,E2,0A,00,F6,FF,32 [1860]
,00,CE,FF,4B,00,B914,2116
4060 DATA B5,FF,6E,00,92,FF,DC,00,24,FF,00 [2588]
,00,00,00,64,00,3108,1558
4070 DATA 9C,FF,F4,01,0C,FE,D0,07,30,F8,AC [3944]
,0D,54,F2,55,55,01AC,2114
4080 DATA 50,55,55,50,54,01,50,50,00,50,40 [2808]
,00,10,40,00,10,4548,815
4090 DATA 4C,C0,10,0F,C0,00,0F,C0,00,0C,C0 [3880]
,00,0C,F3,00,0C,D48F,1169
4100 DATA F3,00,00,3C,00,00,3C,00,00,33,00 [2819]
,40,33,10,40,00,3378,609
4110 DATA 10,40,00,10,50,00,50,54,01,50,55 [2689]
,55,50,55,55,50,136C,921
4120 :                                     [174]
```

# POKER



*Hier »Zocken« die Großmeister, denn Geld spielt keine Rolle*

**Setzen Sie sich zusammen mit Sam Einauge, Whiskey Kid, Postkutschen Ede und Revolver Joe in den Saloon und verspielen Ihr ganzes erspartes Vermögen – oder werden an einem Abend Millionär!**

Alles, was Sie dazu brauchen sind starke Nerven, 20.000 Dollar in bar, einen CPC 464/6128 und natürlich ein steinhartes Pokerface.

## SPIELREGELN

Im Saloon gelten folgende dem Original Draw-Poker etwas abgewandelte Regeln: Gespielt wird mit einem französischen Kartenspiel aus 52 Karten plus einem Joker. Jeder Spieler zahlt zunächst 100$ in den Pott und erhält dafür 5 Karten. Nun beginnt die erste Wettrunde.

Spieler 1 (links vom Kartengeber) macht einen beliebigen Einsatz in den Pott. Hat er sehr schlechte Karten, kann er auch passen und scheidet damit für dieses Spiel aus. In diesem Fall macht dann der nächste Spieler den Ersteinsatz, usw. Für die nachfolgenden Spieler gibt es dann jeweils drei Möglichkeiten:

– Passen:
Man zahlt nichts (mehr) in den Pott und scheidet für dieses Spiel aus.

– Halten:
Man zahlt den zuletzt gebotenen Betrag in den Pott und bleibt damit weiter im Spiel.

– Erhöhen:
Man zahlt mehr als den zuletzt gebotenen Betrag in den Pott. Alle folgenden Spieler müssen nun, um im Spiel zu bleiben, mindestens genausoviel zahlen.

Eine Wettrunde dauert so lange bis ein Spieler erhöht, und daraufhin alle anderen Spieler entweder passen oder halten, aber nicht mehr weiter erhöhen. Anschließend an die erste Wettrunde darf jeder noch im Spiel befindliche Spieler beliebig viele seiner Karten (0–5) umtauschen. Pro umgetauschter Karte muß er 50$ in den Pott zahlen. Er erhält damit die Möglichkeit sein Blatt zu verbessern. Jetzt beginnt die zweite Wettrunde, die genauso verläuft wie die erste. Danach vergleichen alle noch übriggebliebenen Spieler ihre Karten. Der Spieler mit dem besten Blatt gewinnt den gesamten Inhalt des Potts. Die Blätter (aus jeweils 5 Karten) werden dabei jeweils einer der folgenden Gattungen zugeordnet, wobei jede Gattung die jeweils darunterliegenden schlägt:

1. Five of a Kind: fünf gleiche Karten (z.B. vier Buben und der Joker)
2. Straight Flush: Sequenz in einer Farbe (z.B. Pik A-K-D-B-10)
3. Four of a Kind: vier gleiche Karten (z.B. vier Asse)
4. Full House: drei und zwei gleiche Karten (z.B. drei Damen und zwei Achter)
5. Flush: Alle Karten in der gleichen Farbe (z.B fünf Herzkarten)
6. Straight: gemischte Sequenz (z.B. 8-7-6-5-4 in unterschiedlichen Farben)
7. Skip Straight: wie Straight, jedoch in Zweierabständen (z.B. D-10-8-6-4)
8. Round the Corner Straight: wie Straight, jedoch As in der Mitte (z.B. 2-A-K-D-B)
9. Three of a Kind: drei gleiche Karten (z.B. drei Dreier)
10. Blaze: Nur »Bilder« (Buben, Damen und Könige)
11. Two Pairs: Zwei Paare (z.B. zwei Siebener und zwei Zweier)
12. Four Flush: Vier Karten in der gleichen Farbe (z.B. vier Karo, eine Kreuzkarte)
13. One Pair: Ein Paar (z.B. zwei Könige)
14. No Pair: Keine der genannten Kombinationen

Der Joker kann dabei an die Stelle einer beliebigen Karte treten, wodurch eventuell eine höhere Kartenkombination erreicht werden kann. Haben beim Vergleich mehrere Spieler Blätter derselben Gattung, können diese folgendermaßen in der angegebenen Hierarchie (von oben nach unten) weiter unterschieden werden:

1. Kombinationen mit Joker gelten stets weniger als gleichwertige ohne Joker.
2. Der höchste Kartenwert innerhalb der Kombination (z.B. des Paares bei One Pair) gewinnt. Ist auch dieser gleich, gewinnt der zweithöchste, usw.
3. Sind alle Karten innerhalb der Kombination gleichwertig, so gewinnt die höchste/zweithöchste/ usw. Einzelkarte (außerhalb der Kombination)
4. Sind auch alle Einzelkarten gleich, entscheidet schließlich die Farbe der höchsten Karte im Blatt. Reihenfolge der Kar-

tenwerte (von links nach rechts abnehmend):
A-K-D-B-10-9-8-7-6-5-4-3-2
Reihenfolge der Farben (von links nach rechts abnehmend):
Pik – Herz – Karo – Kreuz

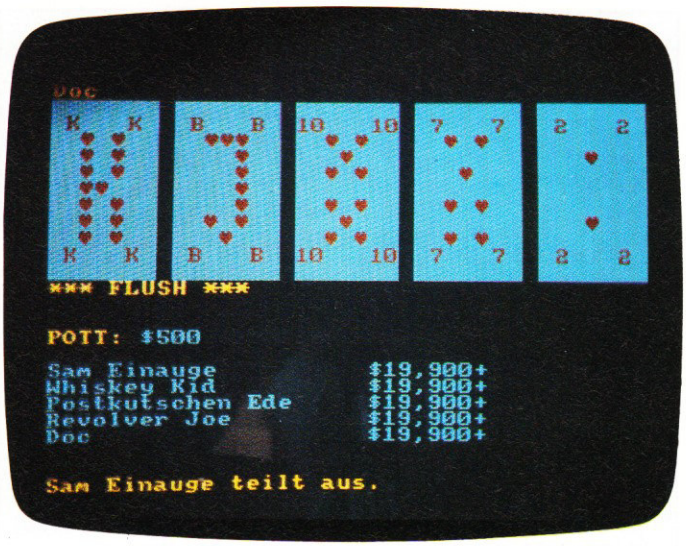
**Sonderregel:**
Bei Flush (nicht bei Four Flush) entscheidet zunächst die Farbe, dann erst (bei gleicher Farbe) die höchste Karte.

## PROGRAMMABLAUF

Zunächst werden Ihnen Ihre vier Mitspieler vorgestellt. Geben dann auch Sie Ihren Namen ein (max. 16 Buchstaben). Nun kann das Spiel beginnen... Alle wichtigen Informationen über den Spielverlauf erscheinen in einem Fenster im unteren Bildschirmbereich. Darüber steht jeweils der augenblickliche Inhalt des Potts, sowie während einer Wettrunde der zuletzt gebotene Einsatz. In der oberen Bildschirmhälfte werden Ihre Karten angezeigt. Wenn ein tiefer Ton ertönt, wartet das Programm so lange, bis Sie die Leertaste drücken. Ein hoher Ton signalisiert einen Eingabefehler. Wiederholen Sie dann Ihre letzte Eingabe. Um abzuheben, geben Sie einfach ein, wieviele Karten des Stoßes Sie hochheben wollen (0–52). Bei der Frage nach Ihrem Einsatz können Sie entweder passen, indem Sie weniger als den zuletzt gebotenen Betrag setzen (sinnvollerweise 0$), oder halten, indem Sie genausoviel zahlen, oder erhöhen, indem Sie mehr bieten. Wenn Sie Karten umtauschen wollen, müssen Sie eine Ziffernfolge aus den Nummern der gewünschten Karten eingeben, wobei die Karten von links nach rechts von 1 bis 5 durchnumeriert zu denken sind. Schreiben Sie die einzelnen Ziffern direkt hintereinander (ohne Trennzeichen) und drücken die ENTER-Taste zum Abschluß. Beispiel: Bei der Eingabe »125« würden die erste, zweite und letzte Karte umgetauscht werden. Wenn Sie nicht umtauschen wollen, machen Sie einfach eine leere Eingabe.
Zum Schluß eines Spiels müssen alle übriggebliebenen Spieler ihre Karten aufdecken. Dabei werden jeweils die Blätter zweier Spieler miteinander verglichen, bis schließlich der Sieger feststeht. Jeder Spieler bringt von zu Hause 20.000$ in bar mit, kann aber während des Spiels noch Schulden machen. Wenn jemand allerdings zu viel verliert, kann es passieren, daß seine Mitspieler Ihm nicht mehr trauen (kann er seine Schulden denn zurückzahlen?) und deshalb nicht weiter mitspielen lassen. Selbstverständlich können Sie jederzeit aus dem Spiel aussteigen, wenn Sie glauben, nun genug gewonnen oder verloren zu haben. Dazu müssen Sie nur zweimal die ESC-Taste drücken.
Und nun viel Spaß, und
Don't loose your wife in Poker-Game!

## VARIABLENLISTE

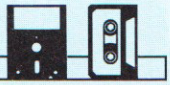
| | |
|---|---|
| a,b | Hilfsvariablen für unterschiedliche Zwecke |
| as | Anzahl der Spieler während einer Wettrunde |
| brk | Belegung der ESC-Taste während INPUT (&FF bei CPC 464, &FC bei CPC 6128, Abfrage Computertyp automatisch) Abstand zweier Kartenwerte bei Prüfung auf Straight (1 für Straight, 2 für Skip-Straight) |
| e | Hilfsvariable für INPUT und »Karten sortieren« |
| f | Art des Blatts (0: No Pair, 1: One Pair, usw.) / Farbe bei »Karten anzeigen« |
| fe | gibt an, wann zuletzt erhöht wurde |
| fj | Flag für Joker |




| | |
|---|---|
| fl | allgem. Flag |
| fs | Flag für Scroll (beim Kartenvergleich) / Flag für Straight (bei »Karten sortieren«) |
| fu,ful | Flags, ob durch Umtausch einer Karte eventuell Straight zu erreichen ist |
| i,j | Schleifenvariablen |
| k | Schleifen- und Zählvariable |
| ka | Zählvariable beim Kartenverteilen, Anfangswert: Anzahl abgehobener Karten |
| l | Zählvariable / Hilfsvariable bei »Karten anzeigen« |
| m | Geld im Pott |
| ob | Flag, ob es sich beim Kartenvergleich um das erste Blatt handelt |
| p | Wert eines Blattes |
| q | Stelle, an der sich im Blatt ein Paar oder Drilling befindet |
| r,r0 | Hilfsvariablen für RND |
| s | Spielernummer |
| sp | Spieler, der Wettrunde eröffnet |
| sp0 | Kartengeber |
| t | Hilfsvariable für Warteschleifen |
| u | Karte, die umgetauscht werden soll |
| ul | Hilfsvariable in Verbindung mit u |
| uz | Anzahl der Karten, die für mögl. Straight umgetauscht werden müßten |
| w | Hilfsvariable für WINDOW |
| wr | Nr. der Wettrunde (1 oder 2)x Hilfsvariable für Cursorpositionierung |
| y | Zählvariable beim Kartengeben / Einsatz eines Spielers |
| z | Wert des höchsten Blattes beim Kartenvergleich / gebotener Betrag bei Wettrunde |
| zs | Spieler mit dem besten Blatt |
| a$,b$, c$,i$ | Hilfsvariablen für unter- schiedliche Zwecke |
| a(x), b(x) | Arbeitsspeicher für je ein Blatt aus 5 Karten |
| k(x) | Kartenstoß |
| f(x) | Strategie des Spielers x |
| l(x) | Höchsteinsatz des Spielers x |
| m(x) | Geld des Spielers x |
| p(x) | Wert d. Karten des Spielers x |
| s(x,y) | Karten des Spielers x |
| u(x) | umzut. Karte des Spielers x |
| n$(x) | Name des Spielers x |

(Michael Knaup)

## für 464-664-6128



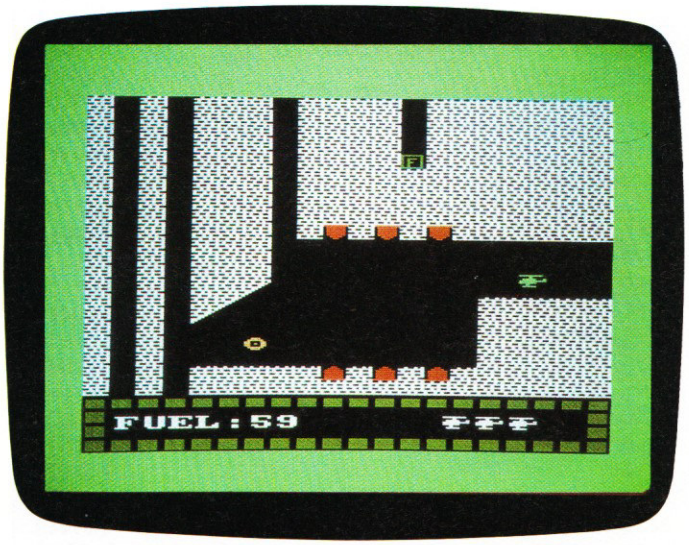
```
1 REM %%%%%%%%%%%%%%%%%%%%          [1239]
2 REM %                    %          [378]
3 REM %   *** POKER ***    %          [704]
4 REM %                    %          [378]
5 REM %  September  1986   %          [1439]
6 REM %                    %          [378]
7 REM %%%%%%%%%%%%%%%%%%%%          [1239]
8 :                                  [174]
10 REM  Version: keine Steuercodes   [1386]
11 :                                 [174]
20 PRINT:PRINT"Farbmonitor [j/n] ?  ";CHR$ [3544]
(143);CHR$(8);
30 WHILE i$()"j" AND i$()"n":i$=LOWER$(INK [3976]
EY$):WEND
40 IF i$="j" THEN PRINT"ja";:RESTORE 10200 [2093]
 ELSE PRINT"nein":RESTORE 10210
50 SYMBOL AFTER 252:RANDOMIZE TIME:GOSUB 8 [1911]
600
60 DIM s(4,5),a(5),b(5),k(52),n$(4),m(4),f [2397]
(4),l(4),u(4),p(4)
70 SYMBOL 252,&7C,&C6,&C6,&DC,&C6,&C6,&D8, [2153]
&C0
80 SYMBOL 253,&66,&0,&78,&C,&7C,&CC,&76,&0 [2423]
90 SYMBOL 254,&66,&0,&3C,&66,&66,&66,&3C,& [2711]
0
100 SYMBOL 255,&66,&0,&66,&66,&66,&66,&3E, [1989]
&0
110 MODE 1:INK 0,0:INK 1,0:INK 2,0:INK 3,0 [4361]
:BORDER 0
120 PAPER 0:CLS:WINDOW#1,13,27,4,8:PAPER#1 [4419]
,2:PEN#1,3:CLS#1
130 LOCATE#1,4,3:PRINT#1,"P O K E R"       [1448]
140 ORIGIN 192,270                         [385]
150 FOR j=1 TO 100                         [769]
160 a=INT(RND*238):b=INT(RND*78)           [2506]
170 PLOT a,b,0:PLOT a+2,b:PLOT a,b+2:PLOT  [3206]
a+2,b+2
180 NEXT:FOR j=1 TO 3:READ a:INK j,a:NEXT  [4146]
190 PEN 1:LOCATE 9,25:PRINT CHR$(164);" 19 [4537]
86 by Michael Knaup"
200 PEN 3:LOCATE 1,12:PRINT"Es spielen:":P [3393]
EN 1:PRINT
210 RESTORE:FOR j=1 TO 4                   [1107]
220 READ n$(j):PRINT n$(j):m(j)=20000      [1637]
230 NEXT:PRINT:PEN 3:PRINT"und";           [2162]
240 PEN 1:INPUT" ",n$:n$=" "+LOWER$(LEFT$( [3144]
n$,16))
250 FOR j=1 TO LEN(n$)                     [852]
260 IF MID$(n$,j,1)=" " THEN n$=LEFT$(n$,j [5248]
)+UPPER$(MID$(n$,j+1,1))+MID$(n$,j+2)
270 NEXT:n$=MID$(n$,2)                     [1589]
280 LOCATE 5,19:PRINT CHR$(18);n$          [2432]
290 PEN 2:PRINT:PRINT"Tiefer Ton = Leertas [4571]
te dr";CHR$(255);"cken."
300 PRINT" Hoher Ton = Eingabe wiederholen [4150]
."
310 n$(0)=n$:m(0)=20000                    [923]
320 GOSUB 8000:sp=1:bl=0:ZONE 18           [3394]
330 WINDOW#1,1,40,18,22:WINDOW#2,1,40,24,2 [3100]
5
340 WINDOW#3,1,20,16,16:WINDOW#4,21,40,16, [2516]
16
350 ON BREAK GOSUB 8700                    [561]
1000 :                                     [174]
1001 REM %%%%%%%%%%%%%%%%%%               [1043]
1002 REM % Hauptprogramm %                 [1350]
1003 REM %%%%%%%%%%%%%%%%%%               [1043]
1004 :                                     [174]
1010 FOR j=0 TO 4:PAPER#j,0:CLS#j:NEXT     [1997]
1020 WINDOW SWAP 1,0:PEN 1:sp0=sp:wr=0     [2622]
1030 FOR j=1 TO 5                          [624]
1040 s=-j*(j(5):IF m(s)=-1 THEN 1070       [1966]
1050 m(s)=m(s)-100:m=m+100                 [928]
1060 PRINT n$(s),USING"$$#####,###+";m(s)  [1495]
1070 NEXT:WINDOW SWAP 1,0                  [1765]
1080 GOSUB 8100                            [820]
1090 LOCATE#2,1,2:PEN#2,3                  [1406]
1100 PRINT#2,n$(sp);" mischt.";            [1789]
1110 k(0)=0:FOR j=1 TO 52                  [1664]
1120 k(j)=j+3:NEXT                         [794]
1130 FOR j=0 TO 52                         [573]
1140 r=INT(RND*53):a=k(j):k(j)=k(r):k(r)=a [1968]
1150 NEXT:t=TIME+300:WHILE t)TIME:WEND     [2738]
1160 a=sp:GOSUB 8250                       [1503]
1170 LOCATE#2,1,2:PRINT#2,CHR$(18);n$(a);" [3567]
 hebt ab";
1180 ON SGN(a)+1 GOSUB 2000,3000           [687]
1190 LOCATE#2,1,2:PRINT#2,CHR$(18);n$(sp); [4624]
" teilt aus.";
1200 a=sp:FOR j=1 TO 5                     [1634]
1210 GOSUB 8200                            [1040]
1220 s(a,j)=k(ka):ka=ka+1:IF ka)52 THEN ka [1937]
=0
1230 IF a()sp THEN 1210                    [542]
1240 NEXT                                  [350]
1250 GOSUB 8200                            [1040]
1260 ON SGN(a)+1 GOSUB 2100,3100           [807]
1270 IF a()sp THEN 1250                    [502]
1280 GOSUB 5000                            [883]
1290 IF as=0 THEN 1800                     [388]
1300 a=sp:LOCATE#2,1,2:PRINT#2,n$(sp0);" t [6278]
eilt aus."
1310 GOSUB 8200:IF f(a)=-1 THEN 1390       [1727]
1320 ON SGN(a)+1 GOSUB 2300,3300           [596]
1330 y=0:FOR j=1 TO 5                      [767]
1340 IF b(j)=1 THEN s(a,j)=k(ka):ka=ka+1:y [2596]
=y+1:IF ka)52 THEN ka=0
1350 NEXT                                  [350]
1360 PEN#1,1:PRINT#1,n$(a);" kauft";y;CHR$ [2742]
(8);"."
1370 m(a)=m(a)-50*y:m=m+50*y:GOSUB 8100    [1331]
1380 ON SGN(a)+1 GOSUB 2100,3500           [791]
1390 IF a()sp THEN 1310                    [546]
1400 GOSUB 5000                            [883]
1410 IF as=0 THEN 1800                     [388]
1500 CLS                                   [91]
1510 ob=1:FOR i=0 TO 4                     [586]
1520 IF m(i)=-1 OR f(i)=-1 THEN 1740       [2570]
1530 IF ob=1 THEN ob=0:x=1:GOSUB 8500:z=p: [1571]
zs=i:GOTO 1740
1540 x=13:GOSUB 8500:fs=0                  [1957]
1550 IF p(z THEN 1700                      [582]
1560 IF p)z THEN 1690                      [1191]
1570 FOR j=1 TO 5:a(j)=s(zs,j):b(j)=s(i,j) [2382]
:NEXT
1580 FOR j=1 TO 4:FOR k=j+1 TO 5           [1435]
1590 IF a(j)(a(k) THEN a=a(j):a(j)=a(k):a( [1476]
k)=a
1600 IF b(j)(b(k) THEN a=b(j):b(j)=b(k):b( [1093]
k)=a
1610 NEXT k,j                              [371]
1620 fl=0:FOR j=1 TO 5                     [776]
1630 IF INT(a(j)/4))INT(b(j)/4) THEN fl=1: [1934]
j=5
```

```
1640 IF INT(a(j)/4)<INT(b(j)/4) THEN f1=-1 [1731]
:j=5
1650 NEXT                                  [350]
1660 IF f1=1 THEN 1700                     [471]
1670 IF f1=-1 THEN 1690                    [594]
1680 IF a(1)>b(1) THEN 1700                [1184]
1690 z=p:zs=i:fs=1                         [895]
1700 PRINT n$(zs);" gewinnt."              [2559]
1710 GOSUB 8000                            [824]
1720 IF fs=1 THEN PRINT STRING$(12,10);    [1822]
1730 LOCATE 1,13:PRINT CHR$(20);           [1740]
1740 NEXT:sp=zs                            [676]
1800 a$=FNd$(m,"$$#####,###")              [888]
1810 a$=MID$(a$,INSTR(a$,"$"))             [1229]
1820 LOCATE 1,16:PEN 3:PRINT CHR$(20);n$(s [3995]
p);" gewinnt ";:PEN 1:PRINT a$;"."
1830 m(sp)=m(sp)+m:m=0                     [801]
1840 IF m(0)<-10000-2000*RND THEN 1950     [1200]
1850 PEN#1,1:FOR j=1 TO 4                  [603]
1860 IF m(j)<-10000-2000*RND THEN m(j)=-1: [3315]
PRINT#1,n$(j);" steigt aus."
1870 NEXT                                  [350]
1880 a=sp0:GOSUB 8200:sp=a:GOSUB 8000      [2626]
1890 IF m(1)<>-1 OR m(2)<>-1 OR m(3)<>-1 O [2549]
R m(4)<>-1 THEN 1000
1900 CLS:PEN 3:PRINT"Herzlichen Gl";CHR$(2 [2987]
55);"ckwunsch !"
1910 a$=FNd$(m(0)-20000,"$$#####,###"):a$=M [2953]
ID$(a$,INSTR(a$,"$"))
1920 PEN 1:PRINT CHR$(10);"Du hast heute " [4790]
;a$;" gewonnen."
1930 PRINT"(Schade, da";CHR$(252);" das Ge [4285]
ld gef";CHR$(253);"lscht ist)"
1940 GOTO 1970                             [337]
1950 GOSUB 8000:CLS:PEN 1                  [2291]
1960 PRINT"Du hast nun alles verspielt,":P [8740]
RINT CHR$(10);"Haus und Hof, Hund und Pfer
d."
1970 PRINT:PRINT"Wir bedanken uns f";CHR$( [3219]
255);"r's mitspielen !"
1980 PRINT:PRINT:PRINT:SYMBOL AFTER 256:CL [3134]
EAR
1990 END                                   [110]
2000 :                                     [174]
2001 REM %%%%%%%%%%%                       [897]
2002 REM % Spieler %                       [1355]
2003 REM %%%%%%%%%%%                       [897]
2004 :                                     [174]
2010 WINDOW SWAP 0,2:x=POS(#0)             [1962]
2020 LOCATE x,2:PRINT" [0-52]: ";:GOSUB 88 [3080]
00:ka=e
2030 IF ka<0 OR ka>52 OR ka>INT(ka) THEN P [2834]
RINT CHR$(7);:GOTO 2020
2040 WINDOW SWAP 0,2                       [1026]
2050 RETURN                                [555]
2060 :                                     [174]
2100 s=0:GOSUB 6000:p(0)=p                 [1744]
2110 LOCATE 1,1:PEN 3:PRINT n$(0)          [810]
2120 x=2:GOSUB 7000                        [1107]
2130 RESTORE 10300                         [707]
2140 FOR j=0 TO INT(p/1000):READ a$:NEXT:L [2445]
OCATE 1,13:PRINT"*** ";a$;" ***";CHR$(18)
2150 a=0:f(0)=0:RETURN                     [1684]
2160 :                                     [174]
2200 x=VPOS(#0)                            [1712]
2210 LOCATE 1,x:PEN 3:PRINT CHR$(18);"Dein [3764]
 Einsatz ? ";
2220 PEN 1:PRINT"$";:GOSUB 8800:y=e        [1863]
2230 IF y<>ROUND(y,-2) THEN PRINT#2,CHR$(7 [5569]
);"Kein Kleingeld,":PRINT#2,"wir spielen u
m 100$-Scheine !";:GOTO 2210
```

```
2240 IF y<z THEN y=0                       [1383]
2250 IF y>100000 THEN y=100000             [1253]
2260 LOCATE 1,x:PEN 1:PRINT CHR$(20);:CLS# [2951]
2
2270 RETURN                                [555]
2280 :                                     [174]
2300 WINDOW SWAP 0,1:x=VPOS(#0)            [1660]
2310 LOCATE 1,x:PEN 3:PRINT CHR$(18);"Umta [4799]
usch (pro Karte $50): ";
2320 PEN 1:GOSUB 8800:a$=STR$(e)           [1720]
2330 FOR j=1 TO 5:b(j)=0:NEXT              [1211]
2340 FOR j=1 TO LEN(a$)                    [968]
2350 b=ASC(MID$(a$,j,1))-48:IF ABS(b-3)<3  [3516]
THEN b(b)=1
2360 NEXT:LOCATE 1,x:PEN 1:PRINT CHR$(20); [2056]
2370 WINDOW SWAP 0,1                       [1031]
2380 RETURN                                [555]
2390 :                                     [174]
3000 :                                     [174]
3001 REM %%%%%%%%%%%%                      [878]
3002 REM % Computer %                      [434]
3003 REM %%%%%%%%%%%%                      [878]
3004 :                                     [174]
3010 PRINT#2,"."                           [525]
3020 ka=INT(RND*53)                        [1252]
3030 t=TIME+600:WHILE t>TIME:WEND          [1872]
3040 RETURN                                [555]
3050 :                                     [174]
3100 s=a:GOSUB 6000:a=s:p(s)=p:u(s)=u      [2932]
3110 IF RND>0.95 OR p>=9000 THEN f(s)=2:l(  [4111]
s)=ROUND(4000+1000*RND,-2):RETURN
3120 IF RND>0.7 OR p>=4000 OR u>0 THEN f(s  [3922]
)=2:l(s)=ROUND(p/5+1000,-2):RETURN
3130 IF p<1000 AND RND>0.5 THEN f(a)=0:RET  [1789]
URN
3140 f(s)=1:l(s)=ROUND(p/2+500,-2)         [1654]
3150 :                                     [174]
3200 IF z<100 THEN y=ROUND(500*RND+200,-2)  [3202]
*f(a):RETURN
3210 IF f(a)=0 THEN y=0:RETURN             [2249]
3220 IF f(a)=1 THEN y=-z*(z<=l(a)):RETURN  [2166]
3230 IF f(a)=2 AND z>l(a) THEN y=z:RETURN  [2796]
3240 y=ROUND(z+(z*RND/2),-2)               [2135]
3250 IF y>m(a)+10000 THEN y=m(a)+10000     [1070]
3260 IF y>100000 THEN y=100000             [1253]
3270 IF y<z THEN y=z                       [1597]
3280 y=ROUND(y,-2):RETURN                  [1774]
3290 :                                     [174]
3300 FOR j=1 TO 5:b(j)=0:NEXT:s=a          [1409]
3310 f=INT(p(a)/1000):IF f>3 AND f<>5 AND  [2895]
f<>11 THEN RETURN
3320 IF u(a)>0 THEN b(u(a))=1:RETURN       [1525]
3330 IF f>0 THEN 3400                      [1449]
3340 FOR j=0 TO 3:a(j)=0:NEXT:FOR i=1 TO 5 [3225]
3350 IF s(a,i)=0 THEN FOR j=0 TO 3:a(j)=a(  [4097]
j)+1:NEXT:GOTO 3370
3360 b=s(a,i) MOD 4:a(b)=a(b)+1            [1356]
3370 NEXT:f1=0:FOR j=0 TO 3:IF a(j)=3 THEN [2980]
 b=j:f1=1:j=3
3380 NEXT:IF f1=1 THEN FOR j=1 TO 5:b(j)=- [3281]
((s(a,j) MOD 4)<>b AND s(a,j)<>0):NEXT:RET
URN
3390 a=INT(RND*3)+INT(RND*3)+1:GOTO 3410   [2100]
3400 a=(f XOR 2) AND 3:IF f<>3 AND RND>0.8  [1665]
 THEN a=a-1
3410 FOR i=5 TO 1 STEP-1                   [867]
3420 f1=1:FOR j=1 TO 5                     [590]
3430 IF j=i THEN 3450                      [305]
3440 IF INT(s(s,i)/4)=INT(s(s,j)/4) OR s(s  [3342]
,i)=0 THEN f1=0:j=5
3450 NEXT:IF f1=1 THEN b(i)=1:a=a-1:IF a=0  [1827]
 THEN i=1
```

```
3460 NEXT:a=s                                          [1045]
3470 RETURN                                            [555]
3480 :                                                 [174]
3500 s=a:GOSUB 6000:a=s:p(s)=p                         [2376]
3510 IF RND)0.95 OR p)=9000 THEN f(s)=2:l( [4742]
s)=ROUND(10000+4000*RND,-2):RETURN
3520 IF RND)0.7 OR p)=4000 OR u)0 THEN f(s [3534]
)=2:l(s)=ROUND(2000+500*RND,-2):RETURN
3530 IF p(1000 AND RND)0.2 THEN f(s)=0:RET [2796]
URN
3540 f(s)=1:l(s)=ROUND(1000+1000*RND,-2):R [2637]
ETURN
5000 :                                                 [174]
5001 REM %%%%%%%%%%%%%%                               [955]
5002 REM % Wettrunde %                                 [934]
5003 REM %%%%%%%%%%%%%%                               [955]
5004 :                                                 [174]
5010 WINDOW SWAP 0,1:z=1:wr=wr+1                       [1495]
5020 as=4:FOR j=0 TO 4:as=as+(m(j)=-1 OR f [2244]
(j)=-1):NEXT
5030 a=sp:fe=as                                        [1367]
5040 GOSUB 8000:CLS:CLS#2                              [2187]
5050 GOSUB 8110:GOSUB 8200:IF f(a)=-1 THEN [2287]
 5120
5060 ON SGN(a)+1 GOSUB 2200,3200                       [471]
5070 m(a)=m(a)-y:m=m+y:GOSUB 8100                      [1729]
5080 PEN 1:PRINT n$(a);" ";                            [1797]
5090 IF y(z THEN f(a)=-1:fe=fe-1:as=as-1:P [6469]
RINT"passt.":IF a=0 THEN LOCATE#1,40,13:PR
INT#1,CHR$(19);
5100 IF y=z THEN fe=fe-1:PRINT"h";CHR$(253 [3434]
);"lt."
5110 IF y)z THEN z=y:fe=as:PRINT"erh";CHR$ [6132]
(254);"ht auf $";MID$(STR$(z),2);".":GOSUB
 8110
5120 IF fe)0 THEN IF a=sp THEN 5040 ELSE 5 [1552]
050
5130 PRINT#2,CHR$(10);"Ende der ";HEX$(wr) [2304]
;". Wettrunde."
5140 GOSUB 8000:CLS:WINDOW SWAP 0,1:CLS#2: [3356]
CLS#4
5150 IF f(sp))=0 THEN RETURN                           [2608]
5160 sp=sp+1:IF sp=5 THEN sp=0                         [1597]
5170 IF m(sp)=-1 THEN 5160 ELSE 5150                   [1701]
6000 :                                                 [174]
6001 REM %%%%%%%%%%%%%%%%%%%%                         [1148]
6002 REM % Karten sortieren %                          [868]
6003 REM %%%%%%%%%%%%%%%%%%%%                         [1148]
6004 :                                                 [174]
6010 GOSUB 8300:u1=0                                   [1642]
6020 d=1:GOSUB 6800:IF fs=1 THEN p=8100-AB [4856]
S(fj)*100+INT(a(1)/4):GOTO 6200
6030 u1=u:d=2:GOSUB 6800:IF fs=1 THEN p=71 [4411]
00-ABS(fj)*100+INT(a(1)/4):GOTO 6200
6040 IF u1=0 THEN u1=u                                 [1355]
6050 FOR j=1 TO 5                                      [624]
6060 IF a(j)=0 THEN 6080                               [1084]
6070 a(j)=a(j)+28:IF a(j))55 THEN a(j)=a(j [1969]
)-52
6080 NEXT:GOSUB 8310:d=1:GOSUB 6800:IF u)0 [3417]
 THEN u=a(u)
6090 FOR j=1 TO 5                                      [624]
6100 IF a(j)=0 THEN 6120                               [1156]
6110 a(j)=a(j)-28:IF a(j)(4 THEN a(j)=a(j) [1447]
+52
6120 NEXT:IF fs=1 THEN f=INT(a(1)/4):p=610 [10477]
0-ABS(fj)*100+f:IF f=4 THEN p=p+2000 ELSE
IF f=3 AND a(5)=0 THEN FOR j=5 TO 2 STEP-1
:a(j)=a(j-1):NEXT:a(1)=0:p=p+2000
6130 IF fs=1 THEN 6200                                 [882]
6140 GOSUB 8300:p=0                                    [1358]
6150 IF u=0 THEN 6200                                  [470]
6160 u=u-28:IF u(4 THEN u=u+52                         [907]
6170 FOR j=1 TO 5                                      [624]
6180 IF a(j)=u THEN u=j:j=5                            [1809]
6190 NEXT:IF u1=0 THEN u1=u                            [1226]
6200 u=u1:a=a(1) MOD 4                                 [921]
6210 f1=1:FOR j=2 TO 5                                 [748]
6220 IF (a(j) MOD 4)()a AND a(j)()0 THEN f [1778]
1=0:j=5
6230 NEXT:IF f1=0 THEN 6300                            [1108]
6240 IF p)=8000 THEN p=p+4000:GOTO 6480                [2169]
6250 IF p)0 THEN GOSUB 8300                            [1260]
6260 p=9100+(a(1) MOD 4):IF a(5)=0 THEN p= [1900]
p-100
6270 GOTO 6480                                         [427]
6300 IF p)0 THEN 6480                                  [1032]
6310 IF a(5)=0 THEN a=1 ELSE a=0                       [1040]
6320 FOR j=0 TO 3:b(j)=a:NEXT                          [1575]
6330 f1=0:FOR j=1 TO 5                                 [776]
6340 IF a(j))0 THEN a=a(j) MOD 4:b(a)=b(a) [3013]
+1:IF b(a)=4 THEN f1=1:j=5
6350 NEXT:IF f1=0 THEN 6400                            [1120]
6360 l=1:FOR j=1 TO 5                                  [909]
6370 IF (a(j) MOD 4)=a THEN b(l)=a(j):l=l+ [2666]
1 ELSE IF a(j)=0 THEN b(4)=0 ELSE b(5)=a(j
)
6380 NEXT:p=2100+INT(a(1)/4):IF a(5)=0 THE [3301]
N p=p-100
6400 a=1:e=5:IF a(5)=0 THEN e=4                        [1178]
6410 FOR i=1 TO e-1:FOR j=i+1 TO e                     [1257]
6420 IF INT(a(i)/4)=INT(a(j)/4) THEN a=a+1 [1395]
:q=j
6430 NEXT j,i:fj=0:IF a(5)=0 THEN fj=1                 [1745]
6440 u1=u:u=0:IF p=0 OR (fj=1 AND a=2) THE [1879]
N 6460
6450 FOR j=1 TO 5:a(j)=b(j):NEXT:u=5:GOTO              [1927]
6490
6460 ON a GOSUB 6500,6550,6600,6650,6700,6 [2536]
500,6750
6470 IF p=0 THEN p=INT(a(1)/4)                         [1139]
6480 IF p)=1000 THEN u=0                               [545]
6490 FOR j=1 TO 5:s(s,j)=a(j):NEXT:RETURN              [1865]
6500 IF fj=0 THEN 6520                                 [738]
6510 FOR j=5 TO 3 STEP-1:a(j)=a(j-1):NEXT: [4346]
a(2)=0:p=1000+INT(a(1)/4):RETURN
6520 u=u1:RETURN                                       [397]
6550 IF fj=0 THEN 6570                                 [696]
6560 FOR j=5 TO q+1 STEP-1:a(j)=a(j-1):NEX [3527]
T:a(q+1)=0:GOTO 6670
6570 p=1100+INT(a(q)/4)                                [758]
6580 RETURN                                            [555]
6600 IF fj=0 THEN 6620                                 [734]
6610 a(5)=a(4):a(4)=a(3):a(3)=0:GOTO 6700              [1070]
6620 f1=1:FOR j=1 TO 5:IF (a(j))51 OR a(j) [4264]
(40) AND a(j)()0 THEN f1=0:j=5
6630 NEXT:p=3100+1000*f1+INT(a(2)/4)+INT(a [2075]
(4)/4)/15
6640 RETURN                                            [555]
6650 IF fj=0 THEN 6670                                 [700]
6660 FOR j=5 TO q+1 STEP-1:a(j)=a(j-1):NEX [3525]
T:a(q+1)=0:GOTO 6760
6670 p=5100-100*fj+INT(a(q)/4)                         [1679]
6680 RETURN                                            [555]
6700 p=10100-100*fj+INT(a(3)/4)                        [1939]
6710 RETURN                                            [555]
6750 IF fj=1 THEN p=13000+INT(a(1)/4):RETU [2525]
RN
6760 p=11100-100*fj+INT(a(3)/4)                        [1563]
6770 RETURN                                            [555]
6780 :                                                 [174]
6800 fs=0:e=5:IF a(5)=0 THEN e=4                       [984]
6810 fj=5-e:a=1:GOSUB 6900                             [1306]
6820 IF uz=0 THEN fs=1:FOR j=1 TO 5:a(j)=b [2094]
```

```
(j):NEXT:RETURN
6830 IF uz=1 THEN RETURN                                [1046]
6840 fj=5-e:a=2:GOSUB 6900                              [1759]
6850 IF uz=0 OR (uz=1 AND fu=1) THEN u=1:R              [2038]
ETURN
6860 fj=5-e:a=1:e=e-1:GOSUB 6900                        [1779]
6870 u=0:IF uz=0 OR (uz=1 AND fu=1) THEN u              [2190]
=e+1
6880 RETURN                                             [555]
6900 u=0:uz=0:fu=0:fu1=0:k=1:FOR j=a TO e-              [2425]
1
6910 b=INT(a(j)/4)-INT(a(j+1)/4)                        [1754]
6920 IF b=d THEN b(k)=a(j):k=k+1:GOTO 6970              [2106]
6930 IF b=2*d AND fj=1 THEN b(k)=a(j):k=k+              [4330]
1:b(k)=0:k=k+1:fj=-1:GOTO 6970
6940 IF b=2*d AND uz>-1 THEN uz=uz+1:u=e:f              [4370]
u=1:GOTO 6970
6950 IF j<e-1 AND uz>-1 THEN IF INT(a(j)/4              [5509]
)-INT(a(j+2)/4)=2*d THEN u=j+1:uz=uz+1-fu:
fu1=1:GOTO 6970
6960 IF fu1=0 THEN uz=-1                                [765]
6970 NEXT:b(k)=a(j):IF k=4 THEN b(5)=0                  [2155]
6980 RETURN                                             [555]
7000 :                                                  [174]
7001 REM %%%%%%%%%%%%%%%%%%%                           [1166]
7002 REM % Karten anzeigen %                            [1560]
7003 REM %%%%%%%%%%%%%%%%%%%                           [1166]
7004 :                                                  [174]
7010 FOR i=1 TO 5                                       [449]
7020 w=8*(i-1)+1:WINDOW#7,w,w+6,x,x+10:WIN              [2564]
DOW SWAP 0,7:PAPER 1:CLS
7030 f=s(s,i) MOD 4:b=INT(s(s,i)/4):IF f=1              [4786]
 OR f=2 OR b=0 THEN PEN 2 ELSE PEN 0
7040 RESTORE 10100:FOR j=0 TO b:READ a$,b$              [2570]
:NEXT:c$=CHR$(226+f)
7050 l=(LEN(a$) MOD 2)+1                                [642]
7060 LOCATE 1,2:PRINT a$                                [1580]
7070 LOCATE 6,2:PRINT a$                                [657]
7080 LOCATE 1,10:PRINT a$                               [1392]
7090 LOCATE 6,10:PRINT a$                               [1015]
7100 IF b=0 THEN GOSUB 7300:GOTO 7150                   [1537]
7110 FOR j=1 TO 7                                       [630]
7120 LOCATE 3,j+2:a=VAL(MID$(b$,j,1))                   [1603]
7130 ON a GOSUB 7200,7210,7220,7230,7240,7              [2524]
250
7140 NEXT                                               [350]
7150 WINDOW SWAP 0,7:NEXT                               [1427]
7160 RETURN                                             [555]
7200 PRINT " ";c$;:RETURN                               [2178]
7210 PRINT c$;" ";c$;:RETURN                            [1427]
7220 PRINT c$;c$;c$;:RETURN                             [886]
7230 PRINT "  ";c$;:RETURN                              [919]
7240 PRINT c$;c$;:RETURN                                [1336]
7250 PRINT " ";c$;c$;:RETURN                            [2594]
7300 PEN 0:FOR j=1 TO 5                                 [1509]
7310 LOCATE j+1,j+3:PRINT MID$(b$,j,1);:NE              [2136]
XT
7320 RETURN                                             [555]
8000 :                                                  [174]
8001 REM %%%%%%%%%%%%%%%%%%%                           [1169]
8002 REM % Unterprogramme %                             [1739]
8003 REM %%%%%%%%%%%%%%%%%%%                           [1169]
8004 :                                                  [174]
8010 SOUND 2,1000,15,6                                  [1426]
8020 WHILE INKEY$()"":WEND                              [1786]
8030 WHILE INKEY$()" ":r0=RND:WEND                      [2934]
8040 RETURN                                             [555]
8050 :                                                  [174]
8100 a$=FNd$(m,"$$#####,###"):b$="POTT:":w=             [2393]
3:GOTO 8120
8110 a$=FNd$(ROUND(z,-2),"$$#####,###"):b$=             [2745]
"Geboten:":w=4
8120 a$=MID$(a$,INSTR(a$,"$"))                          [1229]
8130 PEN#w,3:PRINT#w,CHR$(30);b$;" ";                   [1754]
8140 PEN#w,1:PRINT#w,a$;CHR$(18);                       [2049]
8150 RETURN                                             [555]
8160 :                                                  [174]
8200 a=a+1:IF a=5 THEN a=0                              [1766]
8210 IF m(a)=-1 THEN 8200                               [1026]
8220 RETURN                                             [555]
8250 a=a-1:IF a=-1 THEN a=4                             [1793]
8260 IF m(a)=-1 THEN 8250                               [1024]
8270 RETURN                                             [555]
8280 :                                                  [174]
8300 FOR j=1 TO 5:a(j)=s(s,j):NEXT                      [1081]
8310 FOR i=1 TO 4:FOR j=i+1 TO 5                        [1923]
8320 IF a(i)<a(j) THEN a=a(i):a(i)=a(j):a(              [2252]
j)=a
8330 NEXT j,i                                           [396]
8340 RETURN                                             [555]
8350 :                                                  [174]
8500 s=i:p=p(s)                                         [503]
8510 x=x+1:GOSUB 7000                                   [1924]
8520 LOCATE 1,x-1:PEN 3                                 [924]
8530 RESTORE 10300:FOR j=0 TO INT(p/1000):              [3164]
READ a$:NEXT
8540 a$=n$(s)+": "+a$                                   [522]
8550 IF LEN(a$)>40 THEN a$=LEFT$(a$,LEN(a$              [1117]
)-9)
8560 PRINT a$;                                          [420]
8570 LOCATE 1,x+11                                      [946]
8580 i=s:RETURN                                         [2018]
8590 :                                                  [174]
8600 ON ERROR GOTO 8650                                 [1631]
8610 DEF FNd$(x,u$)=DEC$(x,u$)                          [311]
8620 brk=&FC:i$=FNd$(1,"#")                             [1126]
8630 ON ERROR GOTO 0                                    [1381]
8640 RETURN                                             [555]
8650 DEF FNd$(x,u$)=DEC$((x,u$)                         [193]
8660 brk=&FF                                            [552]
8670 RESUME NEXT                                        [703]
8680 :                                                  [174]
8700 MODE 1:PAPER 0:PEN 1:CLS                           [1766]
8710 IF m(0)>=20000 THEN 1900                           [909]
8720 a$=FNd$(ABS(m(0)-20000),"$$#####,###")             [3086]
:a$=MID$(a$,INSTR(a$,"$"))
8730 PRINT"Du hast heute ";a$;" verloren."              [4142]
8740 GOTO 1970                                          [337]
8750 :                                                  [174]
8800 KEY DEF 66,0,brk,brk,brk                           [1174]
8810 INPUT"",e                                          [386]
8820 KEY DEF 66,0,&FC,&FC,&FC                           [1892]
8830 RETURN                                             [555]
10000 :                                                 [174]
10001 REM %%%%%%%%%                                     [598]
10002 REM % Daten %                                     [708]
10003 REM %%%%%%%%%                                     [598]
10004 :                                                 [174]
10010 DATA Sam Einauge,Whiskey Kid,Postkut              [3622]
schen Ede,Revolver Joe
10020 :                                                 [174]
10100 DATA *,JOKER,2,0100010,3,0101010,4,0              [4090]
200020,5,2001002,6,2002002,7,2010202
10110 DATA 8,2102012,9,2021202,10,2120212,              [4059]
B,3444421,D,1222236,K,2225222
10120 DATA A,0001000                                    [704]
10130 :                                                 [174]
10200 DATA 13,3,15                                      [671]
10210 DATA 13,6,18                                      [365]
10220 :                                                 [174]
10300 DATA NO PAIR,ONE PAIR,FOUR FLUSH,TWO              [4190]
 PAIRS,BLAZE,THREE OF A KIND
10310 DATA ROUND THE CORNER STRAIGHT,SKIP               [4251]
STRAIGHT,STRAIGHT,FLUSH,FULL HOUSE
10320 DATA FOUR OF A KIND,STRAIGHT FLUSH,F              [1980]
IVE OF A KIND
```




# SECRET CAVERNS

**Die Aufgabe des Spieles besteht darin, per Joystick einen Hubschrauber durch ein großes Höhlensystem zu steuern, und den entführten Spion zu retten.**

Das Höhlensystem besteht aus 25 Räumen, in denen verschiedene Gefahren lauern; kreuz und quer schießende Lasergeschütze, Energiebomben und andere Gefahren, wie unsichtbare Wände, Wände, die man mit einer Bombe oder einem Schlüssel öffnen muß.

Bombe und Schlüssel müssen also auch erst einmal gefunden werden, damit auch der Spion gefunden werden kann.

Hat man den Spion erst einmal gefunden, muß man wieder zum Ausgangspunkt zurück.

Um das Spiel nicht zu schwierig zu gestalten, haben wir hierbei eine Abkürzung eingebaut.

Der Treibstoffvorrat kann an Tankstellen aufgefüllt werden. Die Bombe zu werfen, funktioniert allerdings nur in dem Raum, in dem sie auch gebraucht wird.

Wenn man gegen eine Wand fliegt, von einem Laserstrahl getroffen wird, oder einer Energiebombe zu nahe kommt, verliert man eines seiner Leben.

(Thomas Stahn/SH)

## für 464-664-6128

```
100 MODE 1                                          [506]
101 SYMBOL AFTER 32                                 [1296]
102 SYMBOL 200,254,16,121,255,248,32,120,0 [1610]
103 SYMBOL 201,127,8,158,255,31,4,30,0      [1339]
104 SYMBOL 202,73,146,76,185,60,74,146,73   [1764]
105 SYMBOL 203,37,255,73,255,41,255,148,25 [1756]
5
106 SYMBOL 204,37,254,76,248,48,224,192,12 [2964]
8
107 SYMBOL 205,1,3,5,15,25,63,84,255        [1813]
108 SYMBOL 206,128,192,96,240,40,252,150,2 [2185]
55
109 SYMBOL 207,165,127,41,31,9,7,2,1        [2066]
110 SYMBOL 91,0,126,126,126,126,126,126,0   [2595]
111 SYMBOL 209,24,60,126,126,126,126,126,2 [2536]
55
112 SYMBOL 208,255,126,126,126,126,126,60, [2125]
24
113 SYMBOL 210,56,124,198,214,198,124,56,0 [2404]
114 SYMBOL 211,1,63,127,255,255,127,63,6    [1643]
115 SYMBOL 212,128,252,254,255,255,254,252 [2184]
,128
116 SYMBOL 213,254,130,186,162,178,162,162 [2585]
,254
117 SYMBOL 214,16,16,16,56,56,254,170,68    [2096]
118 SYMBOL 215,0,224,224,191,191,227,227,0 [2539]
119 SYMBOL 216,0,12,18,56,124,124,56,16     [1735]
120 SYMBOL 217,0,20,62,20,26,58,31,20       [1590]
121 SYMBOL 218,62,20,62,44,116,62,116,110   [2253]
122 SYMBOL 219,0,0,0,0,4,46,182,255         [2135]
123 SYMBOL 220,255,255,195,219,219,195,255 [2401]
,255
124 MODE 1:BORDER 22:INK 0,22:INK 1,0:INK   [4613]
2,5:PEN 1:PAPER 0:CLS
125 PRINT"  [[[[[ [[[[[ [[[[[ [[[[  [[[[[  [14623]
[[[[[     [     [     [     [   [ [
[       [     [     [     [   [ [       [
      [[[[[ [[[[  [     [[[[  [[[[    [
        [ [     [     [   [ [       [
      [ [     [     [   [ [       ["
126 PRINT"  [[[[[ [[[[[ [[[[[ [   [ [[[[[  [3802]
  [":PRINT:PRINT
127 PRINT"  [[[[[  [[[  [   [ [[[[[ [[[[   [11863]
[   [     [     [   [ [   [ [     [   [ [[
  [     [     [   [ [   [ [     [   [ [ [
[     [     [[[[[  [ [  [[[[  [[[[  [ [ [
    [     [   [  [ [ [  [     [   [ [  [[
  [     [   [  [ [  [     [   [ [   ["
128 PRINT"  [[[[[ [   [   [   [[[[[ [   [  [3335]
[   ["
129 PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PE [6762]
N 2:PRINT" ";CHR$(164);" 1987 Christian Ha
ase & Thomas Stahn"
130 FOR ii=1 TO 5:FOR i=0 TO 26:INK 1,i:SO [3808]
UND 1,100+i*3,1,7:NEXT:NEXT
131 INK 1,0:CALL &BB18                      [911]
132 RUN"!c"                                 [202]
100 ON BREAK GOSUB 7320                     [757]
110 n1$=STRING$(240,203):n2$=STRING$(180,2 [2879]
03)
120 GOTO 550                                [431]
130 ' -----                                 [438]
140 TAG:PLOT -10,-10,7                      [987]
150 ff=ff+1:IF ff=10 THEN GOSUB 7100:ff=0   [2051]
```

```
160 j=JOY(0)                                                    [892]
170 IF j=1 THEN y=y+4:t1=TEST(x,y):t2=TEST                      [5162]
(x+11,y):t3=TEST(x+28,y):t4=TEST(x+21,y)
180 IF j=2 THEN y=y-4:t1=TEST(x,y-14):t2=T                      [5704]
EST(x+11,y-14):t3=TEST(x+28,y-14):t4=TEST(
x+21,y-14)
190 IF j=4 THEN x=x-8:ch=200:t1=TEST(x,y):                      [5211]
t2=TEST(x,y-5):t3=TEST(x,y-10):t4=TEST(x,y
-14)
200 IF j=8 THEN x=x+8:ch=201:t1=TEST(x+31,                      [5592]
y):t2=TEST(x+31,y-5):t3=TEST(x+31,y-10):t4
=TEST(x+31,y-14)
210  IF J<>0 THEN MOVE x1,y1:PRINT" ";                          [2204]
220 IF t1<>0 OR t2<>0 OR t3<>0 OR t4<>0 TH                      [1694]
EN GOTO 270
230 MOVE x,y:PRINT CHR$(ch);                                    [1820]
240 x1=x:y1=y                                                   [650]
250 FOR i=1 TO 7:SOUND 1,0,1,i,,,1:NEXT                        [2090]
260 RETURN                                                      [555]
270 '----- KONTAKT ------                                       [872]
280 IF t1=1 OR t2=1 OR t3=1 OR t4=1 THEN 3                      [1070]
90
290 IF t1=2 OR t2=2 OR t3=2 OR t4=2 THEN x                      [7130]
=x1:y=y1:t1=0:t2=0:t3=0:t4=0:FOR i=0 TO 26
:BORDER i:NEXT:BORDER 18:GOTO 230
300 IF t1=3 OR t2=3 OR t3=3 OR t4=3 THEN x                      [10274]
=x1:y=y1:t1=0:t2=0:t3=0:t4=0:IF fuel<99 TH
EN fuel=fuel+1:SOUND 2,100,2,7:TAG:MOVE 20
0,38:PRINT USING "##";fuel;:GOTO 230 ELSE
SOUND 1,140,3,7:GOTO 230
310 IF t1=4 OR t2=4 OR t3=4 OR t4=4 THEN x                      [6893]
=x1:y=y1:t1=0:t2=0:t3=0:t4=0:schluessel$="
geholt":TAGOFF:LOCATE 6,6:PRINT" ":SOUND 1
,100,20,7:TAG:GOTO 230
320 IF t1=5 OR t2=5 OR t3=5 OR t4=5 THEN x                      [10047]
=x1:y=y1:t1=0:t2=0:t3=0:t4=0:bombe$="gehol
t":TAGOFF:LOCATE 12,19:PRINT" ":SOUND 1,10
0,20,7:TAG:GOTO 230
330 IF t1=6 OR t2=6 OR t3=6 OR t4=6 THEN 3                      [2325]
90
340 IF t1=10 OR t2=10 OR t3=10 OR t4=10 TH                      [9438]
EN x=x1:y=y1:t1=0:t2=0:t3=0:t4=0:spion$="g
eholt":TAGOFF:LOCATE 12,15:PRINT" ":FOR i=
500 TO 50 STEP -10:SOUND 1,i,2,7:NEXT:TAG:
GOTO 230
350 IF t1=14 OR t2=14 OR t3=14 OR t4=14 TH                      [4529]
EN x=x1:y=y1:t1=0:t2=0:t3=0:t4=0:GOTO 230
360 IF t1=11 OR t2=11 OR t3=11 OR t4=11 TH                      [2833]
EN 390
370 IF (t1=13 OR t2=13 OR t3=13 OR t4=13)                       [13369]
AND schluessel$="geholt" THEN PLOT 320,238
,0:DRAW 412,238:PLOT 320,236:DRAW 412,236:
FOR i=200 TO 100 STEP -5:SOUND 1,i,5,7:NEX
T:schluessel$="":gefaengnis$="geoeffnet":t
1=0:t2=0:t3=0:t4=0:GOTO 230
380 IF t1=13 OR t2=13 OR t3=13 OR t4=13 AN                      [4004]
D schluessel$<>"geholt" THEN 390
390 '----- EXPLOSION -----                                      [1620]
400 PLOT -10,-10,1:OUT &BC00,8:OUT &BD00,1                      [2009]
410 SPEED INK 1,1:BORDER 0,25                                   [2297]
420 l=7:FOR a=1 TO 31                                           [1398]
430 IF a>17 THEN l=l-0.5                                        [1184]
440 SOUND 1,0,4,l,,,a                                           [1118]
450 q=q+1:IF q=2 THEN MOVE x1,y1:PRINT CHR                      [3465]
$(202); ELSE MOVE x1,y1:PRINT CHR$(238);
460 IF q=2 THEN q=0                                             [692]
470 NEXT                                                        [350]
480 OUT &BC00,8:OUT &BD00,2                                     [1548]
490 SPEED INK 9,9:BORDER 18                                     [2430]
500 MOVE x1,y1:PRINT" ";                                        [1062]
510 TAGOFF:lives=lives-1:IF lives<1 THEN 7                      [3396]
190
520 x=xt:y=yt:x1=x:y1=y:t1=0:t2=0:t3=0:t4=                      [1836]
0:fuel=99
530 TAG:uu=uu+40:MOVE uu,38:PRINT" ";                          [1837]
540 GOTO 600                                                    [413]
550 ' ----- HAUPTPROGRAMM -----                                 [1684]
560 GOSUB 7100                                                  [879]
570 MODE 0:X=320:X1=X:Y=200:Y1=Y:fuel=99:l                      [4411]
ives=3:uu=400
580 WINDOW#1,1,20,1,21                                          [1118]
590 ch=200:RAUM=1:BORDER 18:GOSUB 7160                          [2625]
600 ON RAUM GOSUB 630,840,950,1250,1360,16                      [8428]
00,1900,2160,2430,2770,2870,3160,3440,3820
,4210,4480,4820,4970,5320,5440,5820,6000,6
390,6540,6710
610 GOSUB 130                                                   [883]
620 GOTO 600                                                    [413]
630 '----- RAUM 1 -----                                        [824]
640 IF altraum=raum THEN 790                                    [1215]
650 IF altraum2=3 THEN x=0                                      [592]
660 xt=x:yt=y:x1=x:y1=y                                         [1309]
670 TAGOFF:PEN 1:GOSUB 7180:CLS#1                               [1376]
680 FOR a=1 TO 20:LOCATE a,1:PRINT CHR$(20                      [7199]
3):LOCATE a,2:PRINT CHR$(203):LOCATE a,20:
PRINT CHR$(203):LOCATE a,21:PRINT CHR$(203
):NEXT
690 FOR a=2 TO 19:LOCATE 1,a:PRINT CHR$(20                      [5467]
3):LOCATE 20,a:PRINT CHR$(203):NEXT
700 RESTORE 720                                                 [731]
710 READ xa,ya,za:IF ya=0 THEN 730 ELSE LO                      [3860]
CATE xa,ya:PRINT CHR$(za):GOTO 710
720 DATA 1,11,32,1,12,32,1,13,32,1,14,32,2                      [5735]
0,11,32,20,12,32,20,13,32,20,14,32,2,3,204
,19,3,207,2,19,206,19,19,205,0,0,0
730 PEN 3:FOR a=1 TO 20:LOCATE a,22:PRINT                       [8199]
CHR$(233):LOCATE a,25:PRINT CHR$(233);:NEX
T:FOR a=23 TO 24:LOCATE 1,a:PRINT CHR$(233
):LOCATE 20,a:PRINT CHR$(233):NEXT
740 TAG:PLOT -10,-10,1:MOVE 40,38:PRINT"FU                      [2847]
EL:99";
750 MOVE 520,38:PRINT CHR$(200);:MOVE 440,                      [4221]
38:PRINT CHR$(200);:MOVE 480,38:PRINT CHR$
(200);
760 PLOT 0,62,14:DRAW 640,62:PLOT 0,60:DRA                      [2576]
W 640,60
770 GOSUB 7170                                                  [865]
780 '                                                           [117]
790 altraum=raum                                                [651]
800 IF x>608 AND spion$="geholt" THEN raum                      [2107]
=2
810 IF x<0 THEN raum=3:altraum2=1                               [1500]
820 IF x>608 AND spion$<>"geholt" THEN 390                      [2620]

830 RETURN                                                      [555]
840 '----- RAUM 2 -----                                        [994]
850 GOSUB 7180:CLS#1                                            [1240]
860 FOR i=1 TO 20:yz=INT(RND(1)*250)+100:x                      [3940]
z=INT(RND(1)*640)+1:PLOT xz,yz,1:NEXT
870 GOSUB 7170                                                  [865]
880 TAG:FOR i=0 TO 608 STEP 8:MOVE ia,y:PR                      [6929]
INT" ";:MOVE i,y:PRINT CHR$(201);:ia=i:FOR
 iii=1 TO 7:SOUND 1,0,1,iii,,,1:NEXT:NEXT:
TAGOFF
890 PAPER 0:PEN 1:MODE 1                                        [1717]
900 LOCATE 8,2:PRINT"HERZLICHEN GLUECKWUNS                      [2638]
CH !"
910 LOCATE 1,8:PRINT"    Sie haben ihren A                      [5500]
genten aus den          'SECRET CAVERNS'
 befreit."
920 FOR i=1 TO 100:i$=INKEY$:i$="":NEXT                        [1395]
930 FOR i=1 TO 7:SOUND 1,0,1,i,,,1:NEXT:i$                      [3311]
=INKEY$:IF i$="" THEN 930
940 GOTO 7250                                                   [313]
950 '----- RAUM 3 -----                                        [1251]
960 IF altraum=raum THEN 1110                                   [1309]
970 IF altraum2=1 THEN x=608                                    [509]
980 IF altraum2=4 THEN y=400                                    [543]
```

```
990 IF altraum2=5 THEN y=84                                    [838]
1000 xt=x:yt=y:x1=x:y1=y                                       [1309]
1010 TAGOFF:PEN 1:GOSUB 7180:CLS#1                             [1376]
1020 FOR a=1 TO 21:LOCATE 1,a:PRINT CHR$(2                     [12946]
03);:LOCATE 3,a:PRINT CHR$(203);:NEXT:FOR
a=1 TO 13:LOCATE 5,a:PRINT CHR$(203):LOCAT
E 6,a:PRINT CHR$(203):LOCATE 7,a:PRINT CHR
$(203):NEXT:FOR a=9 TO 20:FOR b=1 TO 10:LO
CATE a,b:PRINT CHR$(203):NEXT:NEXT
1030 FOR a=5 TO 20:LOCATE a,20:PRINT CHR$(                     [8684]
203):LOCATE a,21:PRINT CHR$(203);:NEXT:FOR
 a=16 TO 20:FOR b=15 TO 19:LOCATE a,b:PRIN
T CHR$(203):NEXT:NEXT
1040 RESTORE 1060                                              [723]
1050 READ xa,ya,za:IF ya=0 THEN 1070 ELSE                      [3846]
LOCATE xa,ya:PRINT CHR$(za):GOTO 1050
1060 DATA 13,1,32,13,2,32,13,3,32,13,4,32,                     [5263]
5,14,203,6,14,203,7,14,204,5,15,203,6,15,2
04,5,16,204,0,0,0
1070 READ xa,ya,za,pa:IF ya=0 THEN 1090 EL                     [4512]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za):GOTO
 1070
1080 DATA 13,5,213,3,10,10,208,8,12,10,208                     [4384]
,8,14,10,208,8,10,20,209,8,12,20,209,8,14,
20,209,8,7,18,210,9,0,0,0,0
1090 PLOT 160,96,6:DRAW 160,120:DRAW 224,1                     [7332]
52:DRAW 256,152:DRAW 256,96:DRAW 252,96:DR
AW 252,150:DRAW 220,150:DRAW 156,118:DRAW
156,96
1100 GOSUB 7170                                                [865]
1110                                                           [117]
1120 altraum=raum                                              [651]
1130 IF x>608 THEN raum=1:altraum2=3                           [1475]
1140 IF y>400 THEN raum=4                                      [1310]
1150 IF y<84 THEN raum=5:altraum2=3                            [2005]
1160 z=INT(RND(1)*400):IF z<10 THEN 1170 E                     [1938]
LSE 1240
1170 z=INT(RND(1)*3+1):ON z GOTO 1180,1200                     [2076]
,1220
1180 PLOT 432,236,8:DRAW 432,100,8:FOR w=7                     [3823]
 TO 1 STEP -1:SOUND 2,80-w,2,w:NEXT:GOSUB
140
1190 PLOT 432,236,0:DRAW 432,100:IF x<=432                     [5103]
 AND x+32>432 THEN 390 ELSE 1240
1200 PLOT 370,236,8:DRAW 370,100,8:FOR w=7                     [3931]
 TO 1 STEP -1:SOUND 2,80-w,2,w:NEXT:GOSUB
140
1210 PLOT 370,236,0:DRAW 370,100:IF x<=370                     [3335]
 AND x+32>370 THEN 390 ELSE 1240
1220 PLOT 306,236,8:DRAW 306,100,8:FOR w=7                     [4159]
 TO 1 STEP -1:SOUND 2,80-w,2,w:NEXT:GOSUB
140
1230 PLOT 306,236,0:DRAW 306,100:IF x<=306                     [3038]
 AND x+32>306 THEN 390 ELSE 1240
1240 RETURN                                                    [555]
1250 '----- RAUM 4 -----                                       [972]
1260 IF altraum=raum THEN 1320                                 [1409]
1270 y=84:xt=x:yt=y:x1=x:y1=y                                  [1469]
1280 TAGOFF:PEN 1:GOSUB 7180:CLS#1                             [1376]
1290 FOR a=1 TO 20:FOR b=1 TO 21:LOCATE a,                     [3729]
b:PRINT CHR$(203);:NEXT:NEXT
1300 FOR a=2 TO 13:LOCATE a,6:PRINT CHR$(3                     [11641]
2):LOCATE a,7:PRINT CHR$(32):NEXT:FOR a=8
TO 21:LOCATE 2,a:PRINT CHR$(32);:LOCATE 4,
a:PRINT CHR$(32);:LOCATE 13,a:PRINT CHR$(3
2);:NEXT:FOR a=10 TO 21:LOCATE 8,a:PRINT C
HR$(32);:NEXT
1310 GOSUB 7170                                                [865]
1320 '                                                         [117]
1330 altraum=raum                                              [651]
1340 IF y<84 THEN raum=3:altraum2=4                            [2320]
1350 RETURN                                                    [555]
1360 '----- RAUM 5 -----                                       [876]
1370 IF altraum=raum THEN 1550                                 [1093]
1380 IF altraum2=3 THEN y=400                                  [753]
1390 IF altraum2=6 THEN y=84                                   [473]
1400 xt=x:yt=y:x1=x:y1=y                                       [1309]
1410 TAGOFF:PEN 1:GOSUB 7180:CLS#1                             [1376]
1420 LOCATE 1,1:PRINT n1$;n2$;                                 [710]
1430 FOR a=1 TO 21:LOCATE 2,a:PRINT CHR$(3                     [13426]
2);:NEXT:FOR a=6 TO 14:LOCATE a,6:PRINT CH
R$(32):LOCATE a+1,7:PRINT CHR$(32):NEXT:FO
R a=5 TO 13:LOCATE a,10:PRINT CHR$(32):LOC
ATE a,11:PRINT CHR$(32):NEXT:FOR a=5 TO 16
:LOCATE a,17:PRINT CHR$(32):LOCATE a,18
1440 PRINT CHR$(32):NEXT:FOR a=16 TO 19:LO                     [4941]
CATE a,8:PRINT CHR$(32):LOCATE a,9:PRINT C
HR$(32):NEXT
1450 RESTORE 1470                                              [737]
1460 READ xa,ya,za:IF ya=0 THEN 1500 ELSE                      [4417]
LOCATE xa,ya:PRINT CHR$(za);:GOTO 1460
1470 DATA 4,1,32,4,2,32,4,3,32,4,4,32,4,5,                     [9116]
206,5,3,207,5,4,32,5,5,32,5,6,206,6,4,207,
6,5,32,6,7,206,7,5,207,5,12,32,5,13,32,5,1
4,32,5,15,32,5,16,32,6,9,32,7,9,32,8,9,207
,13,12,206
1480 DATA 14,8,206,14,10,207,14,11,32,14,1                     [10579]
2,32,14,13,206,15,6,207,15,8,32,15,9,206,1
5,11,207,15,12,32,15,13,32,15,14,206,16,7,
207,16,11,204,16,12,32,16,13,32,16,14,205,
16,16,204,17,10,204,17,11,32,17,12,32,17,1
3,205,17,15,204,17,16,32,17,17,32,17,18,20
5
1490 DATA 18,10,32,18,11,32,18,12,205,18,1                     [7781]
4,204,18,15,32,18,16,32,18,17,205,19,10,32
,19,11,205,19,14,32,19,15,32,19,16,32,19,1
7,32,19,18,32,19,19,32,19,20,32,19,21,32,0
,0,0
1500 RESTORE 1520                                              [743]
1510 READ xa,ya,za,pa:IF ya=0 THEN 1530 EL                     [4930]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za);:GOT
O 1510
1520 DATA 5,9,210,9,0,0,0,0                                    [1274]
1530 PLOT 130,245,6:DRAW 190,245:DRAW 190,                     [3330]
269:DRAW 186,269:DRAW 186,247:DRAW 130,247
1540 GOSUB 7170                                                [865]
1550 '                                                         [117]
1560 altraum=raum                                              [651]
1570 IF y>400 THEN raum=3:altraum2=5                           [2122]
1580 IF y<84 THEN raum=6:altraum2=5                            [2666]
1590 RETURN                                                    [555]
1600 '----- RAUM 6 -----                                       [855]
1610 IF altraum=raum THEN 1780                                 [1303]
1620 IF altraum2=5 THEN y=400                                  [949]
1630 IF altraum2=7 THEN x=0                                    [1342]
1640 xt=x:yt=y:x1=x:y1=y                                       [1309]
1650 TAGOFF:PEN 1:GOSUB 7180:CLS#1                             [1376]
1660 LOCATE 1,1:PRINT n1$;n2$;                                 [710]
1670 FOR a=1 TO 19:LOCATE 19,a:PRINT" ":NE                     [9048]
XT:FOR a=5 TO 13:FOR b=5 TO 20:LOCATE a,b:
PRINT" ":NEXT:NEXT:FOR a=1 TO 4:LOCATE a,9
:PRINT" ":LOCATE a,10:PRINT" ":NEXT
1680 RESTORE 1700                                              [579]
1690 READ xa,ya,za:IF xa=0 THEN 1710 ELSE                      [4272]
LOCATE xa,ya:PRINT CHR$(za):GOTO 1690
1700 DATA 2,1,32,2,2,32,2,3,32,2,4,32,5,5,                     [11909]
204,13,5,207,14,11,207,14,12,32,14,13,32,1
4,14,32,14,15,32,14,16,32,14,17,205,15,12,
207,15,13,32,15,14,32,15,15,32,15,16,205,1
6,12,204,16,13,32,16,14,32,16,15,205,17,11
,204,17,12,32,17,13,32,17,14,205,18,10,204
1710 DATA 18,11,32,18,12,32,18,13,205,0,0,                     [1620]
0
1720 READ xa,ya,za,pa:IF xa=0 THEN 1740 EL                     [4118]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za):GOTO
 1720
1730 DATA 2,5,210,9,6,6,210,9,9,6,210,9,12                     [1423]
,6,210,9,19,20,213,3,0,0,0,0
```

```
1740 PLOT 128,288,6:DRAW 13*32-8,288:DRAW  [2294]
13*32-8,290:DRAW 128,290
1750 PLOT 32,360:DRAW 60,360:DRAW 60,358:D [1746]
RAW 32,358
1760 xw=165:ri=8:xw1=165                   [1837]
1770 GOSUB 7170                            [865]
1780 '                                     [117]
1790 altraum=raum                          [651]
1800 IF x<0 THEN raum=7:altraum2=6         [1030]
1810 IF y>400 THEN raum=5:altraum2=6       [2422]
1820 TAG:PLOT -10,-10,8:MOVE xw1,94:PRINT" [4715]
";:MOVE xw,94:PRINT CHR$(214);:z=INT(RND(
1)*5+1)
1830 IF z=1 THEN PLOT xw+14,96,8:DRAW xw+1 [6639]
4,286:SOUND 1,50,2,7:PLOT xw+14,96,0:DRAW
xw+14,286:IF xw+14>=x AND xw+14<=x+32 THEN
 390
1840 GOSUB 140                             [921]
1850 IF xw>380 THEN ri=-8                  [797]
1860 IF xw<136 THEN ri=8                   [1908]
1870 xw1=xw                                [736]
1880 xw=xw+ri                              [885]
1890 RETURN                                [555]
1900 '----- RAUM 7 -----                   [1195]
1910 IF altraum=raum THEN 2050             [1321]
1920 IF altraum2=6 THEN x=608              [538]
1930 IF altraum2=8 THEN x=0                [189]
1940 xt=x:yt=y:x1=x:y1=y                   [1309]
1950 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
1960 LOCATE 1,1:PRINT n1$;n2$;             [710]
1970 FOR a=2 TO 18:LOCATE 3,a:PRINT" ":LOC [5606]
ATE 8,a:PRINT" ":LOCATE 12,a:PRINT" ":NEXT
1980 FOR a=9 TO 20:LOCATE 18,a:PRINT" ":NE [15384]
XT:FOR a=12 TO 17:LOCATE a,19:PRINT" ":LOC
ATE a,20:PRINT" ":NEXT:FOR a=5 TO 8:LOCATE
 a,19:PRINT" ":LOCATE a,20:PRINT" ":NEXT:F
OR a=9 TO 11:LOCATE a,3:PRINT" ":LOCATE a,
4:PRINT" ":NEXT
1990 RESTORE 2010                          [564]
2000 READ xa,ya,za:IF xa=0 THEN 2020 ELSE  [3926]
LOCATE xa,ya:PRINT CHR$(za);:GOTO 2000
2010 DATA 1,9,32,1,10,32,2,9,32,2,10,32,4, [7387]
3,32,5,3,32,4,4,32,4,17,207,4,18,32,4,19,3
2,4,20,206,3,19,206,5,18,207,19,9,32,20,9,
32,19,10,32,20,10,32,0,0,0
2020 READ xa,ya,za,pa:IF xa=0 THEN 2040 EL [4736]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za);:GOT
O 2020
2030 DATA 5,4,213,3,3,2,208,8,7,4,212,8,8, [3780]
2,208,8,12,2,208,8,13,4,211,8,0,0,0,0
2040 GOSUB 7170                            [865]
2050 '                                     [117]
2060 altraum=raum                          [651]
2070 IF x>608 THEN raum=6:altraum2=7       [1713]
2080 IF x<0 THEN raum=8::altraum2=7        [2228]
2090 z=INT(RND(1)*100)+1:IF z<>1 THEN 2150 [1095]
2100 z=INT(RND(1)*4)+1:ON z GOTO 2110,2120 [2914]
,2130,2140
2110 PLOT 76,366,8:DRAW 76,240:FOR w=7 TO  [11083]
1 STEP -1:SOUND 2,80-w,2,w:NEXT:PLOT 76,36
6,0:DRAW 76,240:IF x<108 AND x>44 AND y>=2
40 THEN 390 ELSE GOSUB 140:GOTO 2150
2120 PLOT 364,366,8:DRAW 364,250:FOR w=7 T [9526]
O 1 STEP -1:SOUND 2,80-w,2,w:NEXT:PLOT 364
,366,0:DRAW 364,250:IF x<396 AND x>332 AND
 y>=250 THEN 390 ELSE GOSUB 140:GOTO 2150
2130 PLOT 236,366,8:DRAW 236,250:FOR w=7 T [9631]
O 1 STEP -1:SOUND 2,80-w,2,w:NEXT:PLOT 236
,366,0:DRAW 236,250:IF x<268 AND x>204 AND
 y>=250 THEN 390 ELSE GOSUB 140:GOTO 2150
2140 PLOT 226,344,8:DRAW 380,344:FOR w=7 T [8578]
O 1 STEP -1:SOUND 2,80-w,2,w:NEXT:PLOT 226
,344,0:DRAW 380,344:IF x<384 AND x>220 AND
 y<360 AND y>328 THEN 390 ELSE GOSUB 140
2150 RETURN                                [555]
2160 '----- RAUM 8 -----                   [1018]
2170 IF altraum=raum THEN 2380             [1103]
2180 IF altraum2=7 THEN x=608              [1576]
2190 IF altraum2=9 THEN x=0                [583]
2200 IF altraum2=16 THEN y=384             [1338]
2210 xt=x:yt=y:x1=x:y1=y                   [1309]
2220 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
2230 LOCATE 1,1:PRINT n1$;n2$;             [710]
2240 FOR a=3 TO 12:FOR b=10 TO 20:LOCATE a [8208]
,b:PRINT" ":NEXT:NEXT:FOR a=11 TO 18:FOR b
=2 TO 14:LOCATE a,b:PRINT" ":NEXT:NEXT
2250 FOR a=2 TO 9:LOCATE 5,a:PRINT" ":NEXT [1537]
2260 FOR a=5 TO 10:FOR b=7 TO 9:LOCATE a,b [1975]
:PRINT" ":NEXT:NEXT
2270 RESTORE 2290                          [764]
2280 READ xa,ya,za:IF xa=0 THEN 2300 ELSE  [3763]
LOCATE xa,ya:PRINT CHR$(za);:GOTO 2280
2290 DATA 3,9,204,4,8,204,4,9,32,6,6,204,7 [11428]
,5,204,7,6,32,8,4,204,8,6,32,9,3,204,9,4,3
2,9,5,32,9,6,32,10,2,204,10,3,32,10,4,32,1
0,5,32,10,6,32,1,13,32,1,14,32,2,13,32,2,1
4,32,19,9,32,19,10,32,20,9,32,20,10,32,8,5
,32,13,15,32,13,16,205,14,15,205,0,0,0
2300 READ xa,ya:IF xa=0 THEN 2320 ELSE LOC [4729]
ATE xa,ya:PEN 9:PRINT CHR$(210):GOTO 2300
2310 DATA 4,11,6,14,6,17,7,6,7,12,12,3,12, [4079]
8,12,11,12,14,11,17,15,12,17,3,17,7,17,13,
0,0
2320 PLOT 572,272,6:DRAWR -96,0:DRAWR 0,11 [3525]
0:DRAWR 4,0:DRAWR 0,-108:DRAWR 92,0
2330 PLOT 414,382:DRAWR 0,-48:DRAWR -96,0: [5194]
DRAWR 0,48:DRAWR 4,0:DRAWR 0,-46:DRAWR 88,
0:DRAWR 0,46
2340 PLOT 224,334:DRAWR 32,0:DRAWR 0,-48:D [5891]
RAWR -96,0:DRAWR 0,16:DRAWR 4,0:DRAWR 0,-1
4:DRAWR 88,0:DRAWR 0,44:DRAWR -28,0
2350 PLOT 64,254:DRAWR 96,0:DRAWR 0,-16:DR [14953]
AWR 96,0:DRAWR 0,-48:DRAWR -32,0:DRAWR 0,-
80:DRAWR -96,0:DRAWR 0,98:DRAWR -64,0:DRAW
R 0,2:DRAWR 68,0:DRAWR 0,-98:DRAWR 88,0:DR
AWR 0,80:DRAWR 32,0:DRAWR 0,44:DRAWR -96,0
:DRAWR 0,16:DRAWR -92,0
2360 PLOT 288,80:DRAWR 0,80:DRAWR 32,0:DRA [12261]
WR 0,144:DRAWR 96,0:DRAWR 0,-64:DRAWR 96,0
:DRAWR 0,-16:DRAWR 64,0:DRAWR 0,-2:DRAWR -
68,0:DRAWR 0,16:DRAWR -96,0:DRAWR 0,64:DRA
WR -88,0:DRAWR 0,-144:DRAWR -32,0:DRAWR 0,
-78
2370 GOSUB 7170                            [865]
2380 '                                     [117]
2390 altraum=raum                          [651]
2400 IF x<0 THEN raum=9:altraum2=8         [1863]
2410 IF x>608 THEN raum=7:altraum2=8       [1403]
2420 RETURN                                [555]
2430 '----- RAUM 9 -----                   [1177]
2440 IF altraum=raum THEN 2560             [1275]
2450 IF altraum2=8 THEN x=608              [70]
2460 IF altraum2=11 THEN x=0               [1087]
2470 IF altraum2=15 THEN y=400             [644]
2480 IF altraum2=10 THEN y=84              [543]
2490 xt=x:yt=y:x1=x:y1=y                   [1309]
2500 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
2510 LOCATE 1,1:PRINT n1$;n2$;             [710]
2520 FOR a=1 TO 4:LOCATE a,3:PRINT" ":LOCA [14639]
TE a,4:PRINT" ":LOCATE a,18:PRINT" ":LOCAT
E a,19:PRINT" ":NEXT:FOR b=1 TO 2:LOCATE 4
,b:PRINT" ":LOCATE 4,22-b:PRINT" ":NEXT:FO
R b=15 TO 21:LOCATE 18,b:PRINT" ":NEXT:FOR
 a=6 TO 14:FOR b=7 TO 12:LOCATE a,b:PRINT"
```

```
2530 NEXT:NEXT:FOR a=1 TO 20:LOCATE a,13:P [4632]
RINT" ";:LOCATE a,14:PRINT" ";:NEXT
2540 PEN 3:LOCATE 3,5:PRINT CHR$(213):PEN  [13009]
8:LOCATE 8,6:PRINT CHR$(208):LOCATE 10,6:P
RINT CHR$(208):LOCATE 12,6:PRINT CHR$(208)
:LOCATE 8,15:PRINT CHR$(209):LOCATE 10,15:
PRINT CHR$(209):LOCATE 12,15:PRINT CHR$(20
9)
2550 GOSUB 7170                            [865]
2560 '                                     [117]
2570 altraum=raum                          [651]
2580 IF x<0 THEN raum=11:altraum2=9        [1464]
2590 IF x>608 THEN raum=8:altraum2=9       [1895]
2600 IF y<84 THEN raum=10                  [1326]
2610 IF y>400 THEN raum=15:altraum2=9      [2428]
2620 z=INT(RND(1)*20+1):IF z>1 THEN 2760   [1143]
2630 z=INT(RND(1)*3+1):ON z GOTO 2640,2680 [2071]
,2720
2640 IF l1=1 THEN l1=0 ELSE l1=1           [1839]
2650 IF l1=1 THEN PLOT 240,302,8:DRAWR 0,- [5166]
126,8 ELSE PLOT 240,302,0:DRAWR 0,-126,0
2660 IF x<=240 AND x+32>240 THEN PLOT 304, [2032]
302,0:DRAWR 0,-126,0:GOTO 390
2670 GOTO 2750                             [389]
2680 IF l2=1 THEN l2=0 ELSE l2=1           [988]
2690 IF l2=1 THEN PLOT 304,302,8:DRAWR 0,- [2638]
126,8 ELSE PLOT 304,302,0:DRAWR 0,-126,0
2700 IF x<=304 AND x+32>304 THEN PLOT 304, [3602]
302,0:DRAWR 0,-126,0:GOTO 390
2710 GOTO 2750                             [389]
2720 IF l3=1 THEN l3=0 ELSE l3=1           [1287]
2730 IF l3=1 THEN PLOT 368,302,8:DRAWR 0,- [4850]
126,8 ELSE PLOT 368,302,0:DRAWR 0,-126,0
2740 IF x<=368 AND x+32>368 THEN PLOT 368, [4042]
302,0:DRAWR 0,-126,0:GOTO 390
2750 SOUND 1,70,2,7                        [1052]
2760 RETURN                                [555]
2770 '----- RAUM 10 -----                  [962]
2780 IF altraum=raum THEN 2840             [1472]
2790 y=400:xt=x:yt=y:x1=x:y1=y             [1649]
2800 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
2810 LOCATE 1,1:PRINT n1$;n2$;             [710]
2820 FOR b=1 TO 17:LOCATE 4,b:PRINT" ":LOC [9607]
ATE 18,b:PRINT" ":NEXT:FOR a=5 TO 17:LOCAT
E a,17:PRINT" ":LOCATE a,16:PRINT" ":NEXT:
PEN 3:LOCATE 10,18:PRINT CHR$(213)
2830 GOSUB 7170                            [865]
2840 altraum=raum                          [651]
2850 IF y>400 THEN raum=9:altraum2=10      [1755]
2860 RETURN                                [555]
2870 '----- RAUM 11 -----                  [983]
2880 IF altraum=raum THEN 3050             [1407]
2890 IF altraum2=9 THEN x=608              [717]
2900 IF altraum2=12 THEN x=0               [456]
2910 IF altraum2=14 THEN y=400             [334]
2920 xt=x:yt=y:x1=x:y1=y                   [1309]
2930 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
2940 FOR a=1 TO 9:LOCATE a,1:PRINT CHR$(20 [3478]
3):LOCATE a,2:PRINT CHR$(203):NEXT
2950 FOR a=11 TO 20:LOCATE a,1:PRINT CHR$( [2498]
203);:LOCATE a,2:PRINT CHR$(203);:NEXT
2960 FOR b=5 TO 17:LOCATE 1,b:PRINT CHR$(2 [4612]
03):LOCATE 2,b:PRINT CHR$(203):NEXT
2970 FOR b=3 TO 17:LOCATE 13,b:PRINT CHR$( [2819]
203):NEXT
2980 FOR a=1 TO 20:LOCATE a,20:PRINT CHR$( [3783]
203);:LOCATE a,21:PRINT CHR$(203);:NEXT
2990 FOR a=17 TO 20:FOR b=5 TO 12:LOCATE a [4619]
,b:PRINT CHR$(203);:NEXT:NEXT
3000 FOR b=6 TO 11:LOCATE 16,b:PRINT CHR$( [5851]
203):NEXT:FOR b=7 TO 10:LOCATE 15,b:PRINT
CHR$(203):NEXT
3010 LOCATE 16,5:PRINT CHR$(205):LOCATE 15 [14099]
,6:PRINT CHR$(205):LOCATE 15,11:PRINT CHR$
(207):LOCATE 16,12:PRINT CHR$(207):LOCATE
14,3:PRINT CHR$(204):LOCATE 14,14:PRINT CH
R$(206):LOCATE 3,10:PRINT CHR$(206):LOCATE
 12,10:PRINT CHR$(205):LOCATE 3,12
3020 PRINT CHR$(204):LOCATE 12,10:PRINT CH [11975]
R$(205):LOCATE 12,12:PRINT CHR$(207):PEN 8
:LOCATE 3,11:PRINT CHR$(212):LOCATE 12,11:
PRINT CHR$(211):LOCATE 6,20:PRINT CHR$(209
):LOCATE 9,20:PRINT CHR$(209):LOCATE 6,2:P
RINT CHR$(208):LOCATE 9,2:PRINT CHR$(208)
3030 PEN 1:FOR a=14 TO 20:FOR b=15 TO 17:L [4560]
OCATE a,b:PRINT CHR$(203);:NEXT:NEXT
3040 GOSUB 7170                            [865]
3050 altraum=raum                          [651]
3060 IF y>400 THEN raum=14:altraum2=11     [1887]
3070 IF x>608 THEN raum=9:altraum2=11      [1951]
3080 IF x<0 THEN raum=12:altraum2=11       [2567]
3090 tt=tt+0.25:t=INT(tt):IF tt>5 THEN tt= [1526]
1
3100 IF t<>tx THEN ON t GOTO 3110,3120,313 [2108]
0,3120 ELSE 3140
3110 PLOT 176,96,8:DRAW 176,366:SOUND 1,10 [6121]
0,2,7:GOSUB 130:PLOT 176,96,0:DRAW 176,366
:IF x<=176 AND x+32>176 THEN 390 ELSE 3140
3120 PLOT 96,232,8:DRAW 348,232:SOUND 1,10 [7595]
0,2,7:GOSUB 130:PLOT 96,232,0:DRAW 348,232
:IF y>=232 AND y-16<232 AND x<400 THEN 390
 ELSE 3140
3130 PLOT 272,96,8:DRAW 272,366:SOUND 1,10 [7580]
0,2,7:GOSUB 130:PLOT 272,96,0:DRAW 272,366
:IF x<=272 AND x+32>272 THEN 390 ELSE 3140
3140 tx=t                                  [152]
3150 RETURN                                [555]
3160 '----- RAUM 12 -----                  [1113]
3170 IF altraum=raum THEN 3320             [1361]
3180 IF altraum2=11 THEN x=608             [1324]
3190 IF altraum2=24 AND y>150 THEN x=64    [1506]
3200 IF altraum2=24 AND y<150 THEN x=0     [1209]
3210 IF altraum2=13 THEN y=400             [960]
3220 xt=x:yt=y:x1=x:y1=y                   [1309]
3230 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
3240 FOR b=1 TO 17:LOCATE 1,b:PRINT CHR$(2 [11183]
03):LOCATE 2,b:PRINT CHR$(203):NEXT:FOR a=
3 TO 20:FOR b=12 TO 17:LOCATE a,b:PRINT CH
R$(203);:NEXT:NEXT:FOR a=1 TO 20:LOCATE a,
20:PRINT CHR$(203);:LOCATE a,21:PRINT CHR$
(203);:NEXT
3250 FOR a=17 TO 20:FOR b=7 TO 11:LOCATE a [12281]
,b:PRINT CHR$(203);:NEXT:NEXT:FOR a=4 TO 1
5:FOR b=1 TO 4:LOCATE a,b:PRINT CHR$(203):
NEXT:NEXT:FOR a=16 TO 20:LOCATE a,1:PRINT
CHR$(203);:LOCATE a,2:PRINT CHR$(203);:NEX
T
3260 RESTORE 3280                          [727]
3270 READ xa,ya,za:IF xa=0 THEN 3290 ELSE  [5253]
LOCATE xa,ya:PRINT CHR$(za);:GOTO 3270
3280 DATA 16,3,203,17,3,204,16,4,204,18,5, [6033]
205,19,5,203,20,5,203,17,6,205,18,6,203,19
,6,203,20,6,203,4,4,207,0,0,0
3290 PEN 8:FOR a=5 TO 15:LOCATE a,4:PRINT  [3351]
CHR$(208):LOCATE a,12:PRINT CHR$(209)
3300 NEXT                                  [350]
3310 GOSUB 7170                            [865]
3320 '                                     [117]
3330 altraum=raum                          [651]
3340 IF x>608 THEN raum=11:altraum2=12     [1590]
3350 IF x<0 THEN raum=24:altraum2=12       [2581]
3360 IF y>400 THEN raum=13:altraum2=12     [2055]
3370 lx=INT(RND(1)*11+5):lx=lx*32-16       [1893]
```

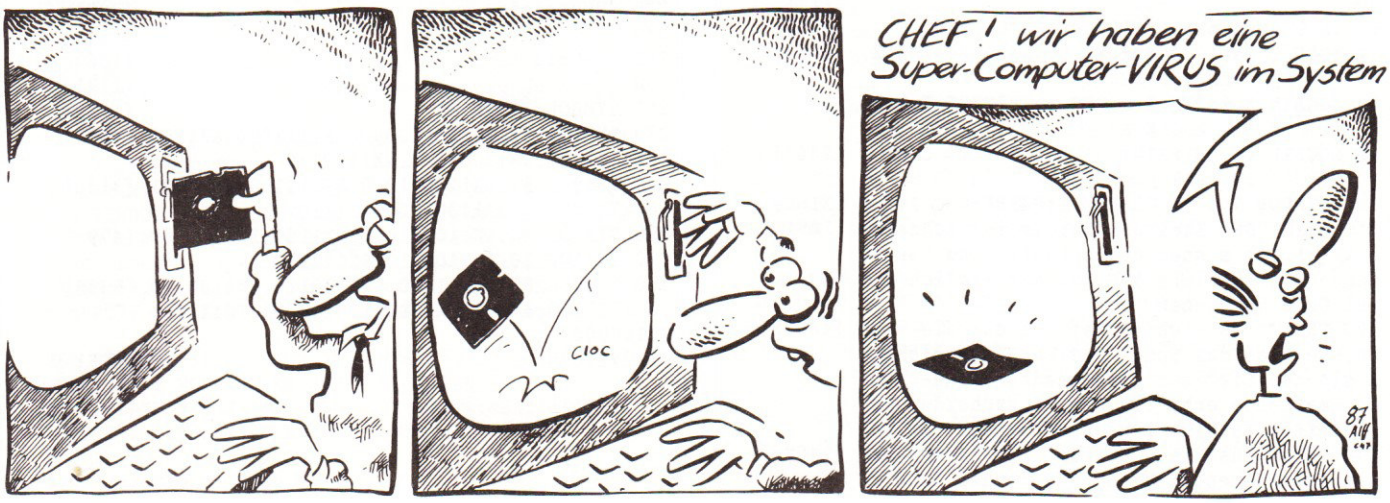
```
3380 PLOT lx,334,8:DRAW lx,294:PLOT lx,224 [4997]
:DRAW lx,264:SOUND 1,80,1,7
3390 GOSUB 140:PLOT lx,334,0:DRAW lx,294:P [2485]
LOT lx,224:DRAW lx,264
3400 IF x<=lx AND x+32>lx THEN 3410 ELSE 3 [1805]
430
3410 IF y<200 THEN 3430                    [1173]
3420 IF y<294 AND y>264 THEN 3430 ELSE 390 [2345]
3430 RETURN                                [555]
3440 '----- RAUM 13 -----                  [1058]
3450 IF altraum=raum THEN 3610             [1391]
3460 IF altraum2=14 THEN x=608             [463]
3470 IF altraum2=12 THEN y=84              [501]
3480 xt=x:yt=y:x1=x:y1=y                   [1309]
3490 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
3500 FOR b=1 TO 21:LOCATE 1,b:PRINT CHR$(2 [14206]
03):LOCATE 2,b:PRINT CHR$(203):NEXT:FOR a=
4 TO 20:LOCATE a,21:PRINT CHR$(203);:LOCAT
E a,20:PRINT CHR$(203);:NEXT:FOR a=3 TO 20
:LOCATE a,1:PRINT CHR$(203);:LOCATE a,2:PR
INT CHR$(203);:NEXT:FOR a=10 TO 17
3510 LOCATE a,3:PRINT CHR$(203):LOCATE a,4 [6418]
:PRINT CHR$(203):NEXT:FOR a=15 TO 20:FOR b
=9 TO 16:LOCATE a,b:PRINT CHR$(203);:NEXT:
NEXT
3520 RESTORE 3540                          [691]
3530 READ xa,ya,za:IF xa=0 THEN 3550 ELSE  [4263]
LOCATE xa,ya:PRINT CHR$(za);:GOTO 3530
3540 DATA 9,2,32,7,3,207,8,3,203,8,4,207,1 [11291]
8,3,203,19,3,203,20,3,203,18,4,204,19,6,20
5,20,6,203,18,7,205,19,7,203,20,7,203,17,8
,205,19,8,203,18,8,203,20,8,203,15,17,203,
16,17,203,17,17,204,19,17,207,20,17,203,20
,18,203,20,19,203,17,20,206,19,20,205,0,0,
0
3550 READ xa,ya,za,pa:IF xa=0 THEN 3570 EL [5350]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za);:GOT
O 3550
3560 DATA 2,10,212,8,2,12,212,8,2,14,212,8 [8351]
,2,16,212,8,15,10,211,8,15,12,211,8,15,14,
211,8,15,16,211,8,10,4,208,8,4,4,210,9,16,
7,210,9,18,20,213,3,10,20,209,8,0,0,0,0
3570 PLOT 64,316,6:DRAW 160,316:DRAW 64,31 [11635]
4:DRAW 160,314:DRAW 160,364:PLOT 164,314:D
RAW 164,364:PLOT 448,270:DRAW 448,312:DRAW
 536,312:DRAW 536,288:PLOT 448,314:DRAW 53
6,314:PLOT 452,270:DRAW 452,312:PLOT 532,2
88:DRAW 532,312
3580 zb=1:zc=0                             [669]
3590 GOSUB 7170                            [865]
3600 '                                     [117]
3610 altraum=raum                          [651]
3620 IF y<84 THEN altraum2=13:raum=12      [1937]
3630 IF y>372 THEN altraum2=13:raum=17     [2094]
3640 IF x>608 THEN altraum2=13:raum=14     [1830]
3650 zb=zb+1:IF zb>4 THEN zb=1             [1027]
3660 GOSUB 140                             [921]
3670 ON zb GOTO 3680,3700,3720,3740        [1725]
3680 PLOT 64,152,8:DRAW 444,152:SOUND 1,60 [3620]
,2,7:PLOT 64,152,0:DRAW 444,152
3690 IF y>=152 AND y-16<152 THEN 390 ELSE  [2236]
3760
3700 PLOT 64,184,8:DRAW 444,184:SOUND 1,60 [3104]
,2,7:PLOT 64,184,0:DRAW 444,184
3710 IF y>=184 AND y-16<184 THEN 390 ELSE  [2433]
3760
3720 PLOT 64,216,8:DRAW 444,216:SOUND 1,60 [4293]
,2,7:PLOT 64,216,0:DRAW 444,216
3730 IF y>=216 AND y-16<216 THEN 390 ELSE  [1741]
3760
3740 PLOT 64,248,8:DRAW 444,248:SOUND 1,60 [3798]
,2,7:PLOT 64,248,0:DRAW 444,248
3750 IF y>=248 AND y-16<248 THEN 390       [1708]
3760 GOSUB 140                             [921]
3770 z=INT(RND(1)*10)+1:IF z=1 AND zc=0 TH [6770]
EN zc=8:SOUND 1,100,2,7:ELSE IF z=1 AND zc
=8 THEN zc=0
3780 PLOT 304,334,zc:DRAW 304,96           [1212]
3790 IF zc<>8 AND z<>1 THEN 3810           [2018]
3800 IF x<=304 AND x+32>304 THEN 390       [842]
3810 RETURN                                [555]
3820 '----- RAUM 14 -----                  [998]
3830 IF altraum=raum THEN 4010             [1375]
3840 IF altraum2=11 THEN y=84              [1106]
3850 IF altraum2=15 THEN x=608             [612]
3860 IF altraum2=13 THEN x=0               [788]
3870 IF altraum2=18 THEN y=400             [1097]
3880 xt=x:yt=y:x1=x:y1=y                   [1309]
3890 TAGOFF:PEN 1:GOSUB 7180:CLS#1         [1376]
3900 LOCATE 1,1:PRINT n1$;n2$;             [710]
3910 FOR a=1 TO 12:LOCATE a,4:PRINT" ":LOC [11881]
ATE a,5:PRINT" ":NEXT:FOR a=3 TO 12:FOR b=
6 TO 19:LOCATE a,b:PRINT" ":NEXT:NEXT:LOCA
TE 10,20:PRINT" ":LOCATE 10,21:PRINT" ":FO
R a=13 TO 20:LOCATE a,13:PRINT" ":LOCATE a
,14:PRINT" ":LOCATE a,18:PRINT" ":LOCATE a
,1
3920 LOCATE a,19:PRINT" ";:NEXT:FOR b=1 TO [10150]
 8:LOCATE 15,b:PRINT" ":NEXT:FOR b=1 TO 6:
LOCATE 18,b:PRINT" ":NEXT:LOCATE 19,5:PRIN
T" ":LOCATE 20,5:PRINT" ":LOCATE 19,6:PRIN
T" ":LOCATE 20,6:PRINT" "
3930 FOR a=7 TO 12:FOR b=4 TO 5:LOCATE a,b [3109]
:PRINT CHR$(203):NEXT:NEXT
3940 PEN 3:LOCATE 15,9:PRINT CHR$(213)     [2147]
3950 RESTORE 3980:PEN 8                    [1368]
3960 ri=12                                 [468]
3970 READ xa,ya,za:IF xa=0 THEN 3990 ELSE  [4124]
LOCATE xa,ya:PRINT CHR$(za):GOTO 3970
3980 DATA 2,7,212,2,11,212,2,16,212,13,9,2 [2602]
11,13,16,211,0,0,0
3990 PEN 1:FOR a=5 TO 12:LOCATE a,12:PRINT [7859]
 CHR$(203):NEXT:LOCATE 5,4:PRINT CHR$(207)
:LOCATE 6,4:PRINT CHR$(203):LOCATE 6,5:PRI
NT CHR$(207)
4000 GOSUB 7170                            [865]
4010 '                                     [117]
4020 altraum=raum                          [651]
4030 IF y<84 THEN altraum2=14:raum=11      [1850]
4040 IF y>400 THEN altraum2=14:raum=18     [1846]
4050 IF x>608 THEN altraum2=14:raum=15     [1447]
4060 IF x<0 THEN altraum2=14:raum=13       [1470]
4070 la=la+ri:IF la>310 THEN ri=-12        [1739]
4080 IF la<12 THEN ri=12                   [960]
4090 GOSUB 140                             [921]
4100 PLOT 64,232,8:DRAW 64+la,232,8:DRAW 3 [4457]
80,232,0:PLOT 64,296,8:DRAW 64+la,296:DRAW
 380,296,0
4110 PLOT 380,264,8:DRAW 380-la,264:DRAW 6 [1786]
4,264,0
4120 IF y>=296 AND y-16<296 AND x<la+64 TH [2488]
EN 390
4130 IF y>=232 AND y-16<232 AND x<la+64 TH [2522]
EN 390
4140 IF y>=264 AND y-16<264 AND x+28>380-l [2204]
a THEN 390
4150 z=INT(RND(1)*5+1):IF z=2 AND l=1 THEN [1849]
 l=0
4160 IF z=3 AND l=0 THEN l=1               [1565]
4170 IF l<>11 THEN IF l=1 THEN PLOT 64,152 [5111]
,8:DRAW 380,152,8:SOUND 1,100,2,7 ELSE PLO
T 64,152,0:DRAW 380,152,0
4180 IF y>=152 AND y-16<152 AND l=1 THEN P [3443]
LOT 64,152,0:DRAW 380,152,0:GOTO 390
```

```
4190 l1=1                                          [484]
4200 RETURN                                        [555]
4210 '----- RAUM 15 -----                          [1030]
4220 IF altraum=raum THEN 4360                     [1377]
4230 IF altraum2=9 THEN y=84                       [770]
4240 IF altraum2=14 THEN x=0                       [455]
4250 IF altraum2=16 THEN x=608                     [1322]
4260 xt=x:yt=y:x1=x:y1=y                           [1309]
4270 TAGOFF:PEN 1:GOSUB 7180:CLS#1                 [1376]
4280 LOCATE 1,1:PRINT n1$;n2$;                     [710]
4290 FOR a=1 TO 20:LOCATE a,5:PRINT" ";:LO         [9263]
CATE a,6:PRINT" ";:LOCATE a,13:PRINT" ";:L
OCATE a,14:PRINT" ";:LOCATE a,18:PRINT" ";
:LOCATE a,19:PRINT" ";:NEXT
4300 LOCATE 3,18:PRINT CHR$(203):LOCATE 3,         [10532]
19:PRINT CHR$(203):LOCATE 4,20:PRINT" ":LO
CATE 4,21:PRINT" ":IF bombe$<>"geholt" THE
N PEN 5:LOCATE 12,19:PRINT CHR$(216)
4310 PEN 8:FOR a=3 TO 18:LOCATE a,3:PRINT          [5555]
CHR$(208):LOCATE a,4:PRINT" ":LOCATE a,7:P
RINT" ":LOCATE a,8:PRINT CHR$(209):NEXT
4320 PEN 2:LOCATE 15,18:PRINT CHR$(149):LO         [3850]
CATE 15,19:PRINT CHR$(149)
4330 lx=304:lxa=lx                                 [907]
4340 GOSUB 7170                                    [865]
4350 '                                             [117]
4360 altraum=raum                                  [651]
4370 IF x<0 THEN raum=14:altraum2=15               [1791]
4380 IF x>608 THEN raum=16:altraum2=15             [1455]
4390 IF y<84 THEN raum=9:altraum2=15               [1992]
4400 z=INT(RND(1)*2)+1:IF z=1 THEN lx=lx-3         [2757]
2 ELSE lx=lx+32
4410 IF lx<80 THEN lx=560                          [579]
4420 IF lx>560 THEN lx=80                          [1740]
4430 PLOT lxa,350,0:DRAW lxa,288:PLOT lx,3         [4405]
50,8:DRAW lx,288:SOUND 1,30,2,7
4440 IF x<=lx AND x+32>lx AND y>240 THEN P         [3936]
LOT lx,350,0:DRAW lx,288:GOTO 390
4450 GOSUB 140                                     [921]
4460 lxa=lx                                        [287]
4470 RETURN                                        [555]
4480 '----- RAUM 16 -----                          [1068]
4490 IF altraum=raum THEN 4680                     [1361]
4500 IF altraum2=15 THEN x=0                       [325]
4510 IF altraum2=25 THEN x=608                     [977]
4520 IF altraum2=22 THEN y=400                     [735]
4530 xt=x:yt=y:x1=x:y1=y                           [1309]
4540 TAGOFF:PEN 1:GOSUB 7180:CLS#1                 [1376]
4550 FOR a=1 TO 11:FOR b=1 TO 4:LOCATE a,b         [14911]
:PRINT CHR$(203):NEXT:NEXT:FOR a=1 TO 11:F
OR b=7 TO 12:LOCATE a,b:PRINT CHR$(203):NE
XT:NEXT:FOR a=16 TO 18:FOR b=5 TO 8:LOCATE
 a,b:PRINT CHR$(203):NEXT:NEXT:FOR b=1 TO
17:LOCATE 19,b:PRINT CHR$(203):LOCATE 20,b
:PR
4560 FOR a=6 TO 20:LOCATE a,20:PRINT CHR$(         [12974]
203);:LOCATE a,21:PRINT CHR$(203);:NEXT:FO
R b=15 TO 17:LOCATE 1,b:PRINT CHR$(203):LO
CATE 2,b:PRINT CHR$(203):NEXT:FOR a=1 TO 4
:LOCATE a,21:PRINT CHR$(203):LOCATE a,20:P
RINT CHR$(203):NEXT
4570 FOR a=5 TO 20:LOCATE a,16:PRINT CHR$(         [7959]
203);:LOCATE a,17:PRINT CHR$(203);:NEXT:FO
R a=11 TO 20:LOCATE a,15:PRINT CHR$(203):N
EXT:FOR a=17 TO 20:LOCATE a,12:PRINT CHR$(
203);:NEXT
4580 FOR a=9 TO 11:LOCATE a,5:PRINT CHR$(2         [6439]
03):LOCATE a,6:PRINT CHR$(203):NEXT:FOR b=
13 TO 21:LOCATE 4,b:PRINT CHR$(203):NEXT
4590 RESTORE 4610                                  [717]
4600 READ xa,ya,za,pa:IF xa=0 THEN 4610 EL         [6347]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za);:GOT
O 4600
4610 DATA 12,12,203,1,13,12,203,1,14,12,20         [11371]
6,1,12,11,203,1,13,11,206,1,12,10,206,1,16
,12,205,1,17,11,205,1,18,11,203,1,18,10,20
5,1,8,6,213,3,16,13,211,8,5,12,208,8,7,12,
208,8,9,12,208,8,7,16,209,8,9,16,209,8,17,
13,203,1,18,13,203,1
4620 DATA 16,14,203,1,17,14,203,1,18,14,20         [5464]
3,1,5,16,32,1,5,17,32,1,12,2,210,9,16,2,21
0,9,14,7,210,9,0,0,0,0
4630 IF schluessel$<>"geholt" THEN PEN 4:L         [5306]
OCATE 6,6:PRINT CHR$(215)
4640 PLOT 352,352,6:DRAW 412,352:DRAW 412,         [15862]
400:PLOT 352,354:DRAW 412,354:PLOT 448,400
:DRAW 448,352:DRAW 500,352:DRAW 500,336:PL
OT 448,354:DRAW 500,354:PLOT 476,318:DRAW
384,318:DRAW 400,272:DRAW 476,272:DRAW 476
,274:DRAW 400,274:PLOT 476,316:DRAW 384,31
6
4650 PLOT 408,400:DRAW 408,352:PLOT 452,40         [5041]
0:DRAW 452,352:PLOT 388,318:DRAW 404,272
4660 GOSUB 7170                                    [865]
4670 '                                             [117]
4680 altraum=raum                                  [651]
4690 IF x<0 THEN raum=15:altraum2=16               [1899]
4700 IF x>608 THEN raum=25:altraum2=16             [1856]
4710 IF y<84 THEN raum=8:altraum2=16               [2822]
4720 IF y>400 THEN raum=22:altraum2=16             [2910]
4730 z=INT(RND(1)*30+1):IF z=1 THEN PLOT 4         [7865]
76,200,8:DRAW 224,200,8:SOUND 1,70,2,7:PLO
T 476,200,0:DRAW 224,200:IF x+32>224 AND y
>=200 AND y-16<200 THEN 390
4740 GOSUB 140                                     [921]
4750 z=INT(RND(1)*4+1):ON z GOTO 4760,4770         [2144]
,4780,4790
4760 PLOT 208,206,8:DRAW 208,183:SOUND 1,9         [5806]
0,2,7:PLOT 208,206,0:DRAW 208,183:IF y>=18
3 AND y<250 AND x<=208 AND x+32>208 THEN 3
90 ELSE 4800
4770 PLOT 208,160,8:DRAW 208,163:SOUND 1,9         [5582]
0,2,7:PLOT 208,160,0:DRAW 208,163:IF y-16<
163 AND y>150 AND x<=208 AND x+32>208 THEN
 390 ELSE 4800
4780 PLOT 272,206,8:DRAW 272,203:SOUND 1,9         [7200]
0,2,7:PLOT 272,206,0:DRAW 272,203:IF y>=20
3 AND y<250 AND x<=272 AND x+32>272 THEN 3
90 ELSE 4800
4790 PLOT 272,160,8:DRAW 272,183:SOUND 1,9         [6374]
0,2,7:PLOT 272,160,0:DRAW 272,183:IF y-16<
183 AND y>150 AND x<=272 AND x+32>272 THEN
 390
4800 z=INT(RND(1)*30)+1:IF z=1 THEN PLOT 1         [5748]
44,206,8:DRAW 144,160:SOUND 1,90,2,7:PLOT
144,206,0:DRAW 144,160:IF y>=160 AND y<250
 AND x<=144 AND x+32>144 THEN 390
4810 RETURN                                        [555]
4820 '----- RAUM 17 -----                          [1093]
4830 IF altraum=raum THEN 4930                     [1407]
4840 IF altraum2=13 THEN y=120                     [921]
4850 IF altraum2=21 THEN x=0                       [834]
4860 IF altraum2=18 THEN x=608                     [1226]
4870 xt=x:yt=y:x1=x:y1=y                           [1309]
4880 TAGOFF:PEN 1:GOSUB 7180:CLS#1                 [1376]
4890 LOCATE 1,1:PRINT n1$;n2$;                     [710]
4900 FOR a=1 TO 20:LOCATE a,7:PRINT" ";:LO         [13308]
CATE a,8:PRINT" ";:LOCATE a,18:PRINT" ";:L
OCATE a,19:PRINT" ";:NEXT:FOR a=1 TO 8:LOC
ATE a,13:PRINT" ":LOCATE a,14:PRINT" ":NEX
T:FOR b=10 TO 17:LOCATE 9,b:PRINT" ":NEXT
4910 GOSUB 7170                                    [865]
4920 '                                             [117]
4930 altraum=raum                                  [651]
```

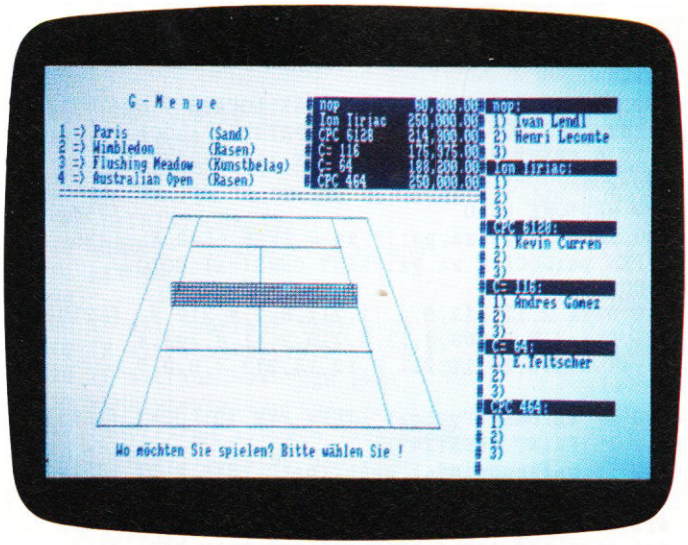
```
4940 IF x<0 THEN raum=21:altraum2=17          [1601]
4950 IF x>608 THEN raum=18:altraum2=17        [1284]
4960 RETURN                                   [555]
4970 '----- RAUM 18 -----                     [1397]
4980 IF altraum=raum THEN 5150                [1428]
4990 IF altraum2=19 THEN y=400                [469]
5000 IF altraum2=17 THEN x=0                  [834]
5010 IF altraum2=14 THEN y=84                 [515]
5020 IF altraum2=20 THEN x=608                [626]
5030 xt=x:yt=y:x1=x:y1=y                      [1309]
5040 TAGOFF:PEN 1:GOSUB 7180:CLS#1            [1376]
5050 LOCATE 1,1:PRINT n1$;n2$;                [710]
5060 FOR a=1 TO 20:LOCATE a,7:PRINT" ";:LO    [12040]
CATE a,8:PRINT" ";:NEXT:FOR a=1 TO 15:LOCA
TE a,18:PRINT" ":LOCATE a,19:PRINT" ":NEXT
:FOR b=1 TO 6:LOCATE 15,b:PRINT" ":NEXT:FO
R b=9 TO 21:LOCATE 18,b:PRINT" ":NEXT
5070 FOR b=13 TO 17:FOR a=b TO 17:LOCATE a    [2226]
,b-4:PRINT" ":NEXT:NEXT
5080 RESTORE 5100                             [922]
5090 READ xa,ya,za:IF xa=0 THEN 5110 ELSE     [4169]
LOCATE xa,ya:PRINT CHR$(za):GOTO 5090
5100 DATA 12,6,204,12,9,206,13,5,204,13,6,    [7151]
32,13,10,206,14,5,32,14,6,32,14,11,206,15,
12,32,15,20,32,15,21,32,16,5,32,16,6,32,16
,13,206,17,5,207,17,6,32,17,14,206,18,6,20
7,0,0,0
5110 PEN 8:READ xa,ya,za:IF xa=0 THEN 5130    [3420]
 ELSE LOCATE xa,ya:PRINT CHR$(za):GOTO 511
0
5120 DATA 7,17,208,7,20,209,9,17,208,9,20,    [3065]
209,15,13,209,0,0,0
5130 GOSUB 7170                               [865]
5140 '                                        [117]
5150 altraum=raum                             [651]
5160 IF x<0 THEN raum=17:altraum2=18          [1612]
5170 IF x>608 THEN raum=20:altraum2=18        [1822]
5180 IF y>400 THEN raum=19                    [1417]
5190 IF y<84 THEN raum=14:altraum2=18         [2064]
5200 z=INT(RND(1)*30)+1:IF z=1 THEN p1=0 E    [1970]
LSE IF z=2 THEN p1=8
5210 IF p1<>pa1 THEN PLOT 208,96,p1:DRAW 2    [4103]
08,126:IF p1=8 THEN SOUND 1,120,2,7
5220 IF p1=8 AND y<200 AND x<=208 AND x+32    [4938]
>208 THEN PLOT 208,96,0:DRAW 208,126:GOTO
390
5230 pa1=p1                                   [110]
5240 GOSUB 140                                [921]
5250 z=INT(RND(1)*30)+1:IF z=1 THEN p2=0 E    [1277]
LSE IF z=2 THEN p2=8
5260 IF p2<>pa2 THEN PLOT 272,96,p2:DRAW 2    [3864]
72,126:IF p2=8 THEN SOUND 1,120,2,7
5270 IF p2=8 AND y<200 AND x<=272 AND x+32    [4329]
>272 THEN PLOT 272,96,0:DRAW 272,126:GOTO
390
5280 pa2=p2                                   [435]
5290 GOSUB 140                                [921]
5300 z=INT(RND(1)*15)+1:IF z=1 THEN PLOT 4    [7713]
64,208,8:DRAW 464,400:SOUND 1,60,2,7:PLOT
464,208,0:DRAW 464,400:IF y>150 AND x<=464
 AND x+32>464 THEN 390
5310 RETURN                                   [555]
5320 '----- RAUM 19 -----                     [1301]
5330 IF altraum=raum THEN 5410                [1448]
5340 y=84:xt=x:yt=y:x1=x:y1=y                 [1469]
5350 TAGOFF:PEN 1:GOSUB 7180:CLS#1            [1376]
5360 LOCATE 1,1:PRINT n1$;n2$;                [710]
5370 FOR b=17 TO 21:LOCATE 15,b:PRINT" ":N    [6670]
EXT:FOR a=10 TO 14:LOCATE a,17:PRINT" ":LO
CATE a,18:PRINT" ":NEXT
5380 PEN 3:LOCATE 10,18:PRINT CHR$(213)       [1240]
5390 GOSUB 7170                               [865]
5400 '                                        [117]
5410 altraum=raum                             [651]
5420 IF y<84 THEN raum=18:altraum2=19         [2785]
5430 RETURN                                   [555]
5440 '----- RAUM 20 -----                     [858]
5450 IF altraum=raum THEN 5570                [1440]
5460 IF altraum2=18 THEN x=0                  [1158]
5470 IF altraum2=22 THEN x=608                [798]
5480 xt=x:yt=y:x1=x:y1=y                      [1309]
5490 TAGOFF:PEN 1:GOSUB 7180:CLS#1            [1376]
5500 LOCATE 1,1:PRINT n1$;n2$;                [710]
5510 FOR a=1 TO 20:LOCATE a,7:PRINT" ";:LO    [14701]
CATE a,8:PRINT" ";:NEXT:FOR a=3 TO 13:FOR
b=3 TO 12:LOCATE a,b:PRINT" ":NEXT:NEXT:FO
R b=10 TO 19:LOCATE 17,b:PRINT" ":NEXT:FOR
 a=18 TO 20:LOCATE a-2,9:PRINT" ":LOCATE a
,18:PRINT" ";:LOCATE a,19:PRINT" ";:NEXT
5520 LOCATE 15,9:PRINT CHR$(206):LOCATE 16    [6829]
,10:PRINT CHR$(206):LOCATE 18,10:PRINT CHR
$(205):LOCATE 19,9:PRINT CHR$(205)
5530 PEN 8:LOCATE 2,6:PRINT CHR$(208):LOCA    [6463]
TE 2,9:PRINT CHR$(209):LOCATE 14,6:PRINT C
HR$(208):LOCATE 14,9:PRINT CHR$(209)
5540 xb=8:xba=xb:yb=7:yba=yb                  [2293]
5550 GOSUB 7170                               [865]
5560 '                                        [117]
5570 altraum=raum                             [651]
5580 IF x<0 THEN raum=18:altraum2=20          [2149]
5590 IF x>608 THEN raum=22:altraum2=20        [1832]
5600 '                                        [117]
5610 z=INT(RND(1)*20)+1:IF z=1 THEN p1=0 E    [2735]
LSE IF z=2 THEN p1=8
5620 IF p1<>pa1 THEN PLOT 48,302,p1:DRAW 4    [4213]
8,272:IF p1=8 THEN SOUND 1,120,2,7
5630 IF p1=8 THEN IF x<=48 AND x+32>48 THE    [3334]
N PLOT 48,302,0:DRAW 48,272:GOTO 390
5640 GOSUB 140                                [921]
5650 pa1=p1                                   [110]
5660 z=INT(RND(1)*20)+1:IF z=1 THEN p2=0 E    [2838]
LSE IF z=2 THEN p2=8
5670 IF p2<>pa2 THEN PLOT 432,302,p2:DRAW     [4649]
432,272:IF p2=8 THEN SOUND 1,120,2,7
5680 IF p2=8 THEN IF x<=432 AND x+32>432 T    [4898]
HEN PLOT 432,302,0:DRAW 432,272:GOTO 390
5690 GOSUB 140                                [921]
5700 pa2=p2                                   [435]
5710 TAGOFF:PEN 11                            [1272]
5720 z=INT(RND(1)*3)+1:IF z=1 THEN xb=xb-1    [3837]
 ELSE IF z=3 THEN xb=xb+1
5730 z=INT(RND(1)*3)+1:IF z=1 THEN yb=yb-1    [3768]
 ELSE IF z=3 THEN yb=yb+1
5740 IF xb<3 THEN xb=3                        [1014]
5750 IF xb>13 THEN xb=13                      [1449]
5760 IF yb<3 THEN yb=3                        [1271]
5770 IF yb>12 THEN yb=12                      [1108]
5780 LOCATE xba,yba:PRINT" ":LOCATE xb,yb:    [3322]
PRINT CHR$(220)
5790 IF x+32>(xb-1)*32 AND x<=xb*32-4 AND     [5699]
y-16<398-(yb-1)*16 AND y>=400-yb*16 THEN L
OCATE xb,yb:PRINT" ":TAG:GOTO 390
5800 xba=xb:yba=yb                            [557]
5810 RETURN                                   [555]
5820 '----- RAUM 21 -----                     [1093]
5830 IF altraum=raum THEN 5960                [1203]
5840 IF altraum2=17 THEN x=608                [953]
5850 IF altraum2=23 THEN y=84                 [1106]
5860 xt=x:yt=y:x1=x:y1=y                      [1309]
5870 TAGOFF:PEN 1:GOSUB 7180:CLS#1            [1376]
5880 LOCATE 1,1:PRINT n1$;n2$;                [710]
5890 FOR a=3 TO 20:LOCATE a,7:PRINT" ";:LO    [13629]
CATE a,8:PRINT" ";:LOCATE a,13:PRINT" ";:L
OCATE a,14:PRINT" ";:NEXT:FOR b=15 TO 21:L
```

```
OCATE 3,b:PRINT" ":LOCATE 7,b:PRINT" ":LOC
ATE 12,b:PRINT" ":LOCATE 14,b:PRINT" ":NEX
T
5900 RESTORE 5920                                   [567]
5910 READ xa,ya:IF xa=0 THEN 5930 ELSE LOC          [4715]
ATE xa,ya:PRINT" ";:GOTO 5910
5920 DATA 3,9,3,10,3,11,3,12,18,18,18,19,1          [3500]
8,20,18,21,19,18,19,19,20,18,20,19,0,0
5930 PEN 3:LOCATE 5,14:PRINT CHR$(213):PEN          [3913]
 2:LOCATE 5,13:PRINT CHR$(149)
5940 GOSUB 7170                                     [865]
5950 '                                              [117]
5960 altraum=raum                                   [651]
5970 IF x>608 THEN raum=17:altraum2=21              [1780]
5980 IF y<84 THEN raum=23:altraum2=21               [2331]
5990 RETURN                                         [555]
6000 '----- RAUM 22 -----                           [1178]
6010 IF altraum=raum THEN 6190                      [1396]
6020 IF altraum2=20 THEN x=0                        [456]
6030 IF altraum2=16 THEN y=84                       [1105]
6040 xt=x:yt=y:x1=x:y1=y                            [1309]
6050 TAGOFF:PEN 1:GOSUB 7180:CLS#1                  [1376]
6060 LOCATE 1,1:PRINT n1$;n2$;                      [710]
6070 FOR a=12 TO 18:FOR b=3 TO 21:LOCATE a          [11087]
,b:PRINT" ":NEXT:NEXT:FOR a=1 TO 9:LOCATE
a,7:PRINT" ":LOCATE a,8:PRINT" ":NEXT:FOR
a=8 TO 11:FOR b=3 TO 6:LOCATE a-b+3,b:PRIN
T" ":NEXT:NEXT
6080 RESTORE 6100                                   [739]
6090 READ xa,ya,za:IF xa=0 THEN 6110 ELSE           [3058]
LOCATE xa,ya:PRINT CHR$(za):GOTO 6090
6100 DATA 1,18,32,1,19,32,2,18,32,2,19,32,          [5365]
3,18,32,4,6,204,5,5,204,6,4,204,7,3,204,9,
6,32,10,5,205,11,4,32,18,3,207,0,0,0
6110 READ xa,ya,za,pa:IF xa=0 THEN 6130 EL          [4184]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za):GOTO
 6110
6120 DATA 3,19,213,3,8,7,210,9,11,10,212,8          [4577]
,11,16,212,8,13,2,208,8,16,2,208,8,19,10,2
11,8,19,16,211,8,0,0,0,0
6130 PLOT 192,272,6:DRAW 208,318:DRAW 284,          [2814]
318:PLOT 208,316:DRAW 284,316
6140 PLOT 352,78,2:DRAW 412,78:DRAW 412,64          [7347]
:PLOT 352,76:DRAW 412,76:PLOT 448,64:DRAW
448,78:DRAW 572,78:PLOT 448,76:DRAW 572,76
6150 PEN 12:IF mauer$<>"gesprengt" THEN LO          [6556]
CATE 4,7:PRINT CHR$(217):LOCATE 4,8:PRINT
CHR$(218) ELSE LOCATE 4,8:PRINT CHR$(219)
6160 z=0                                            [343]
6170 GOSUB 7170                                     [865]
6180 '                                              [117]
6190 altraum=raum                                   [651]
6200 IF y<84 THEN raum=16:altraum2=22               [2641]
6210 IF x<0 THEN raum=20:altraum2=22                [2218]
6220 z=z+1:IF z>4 THEN z=1                          [867]
6230 GOSUB 140                                      [921]
6240 ON z GOTO 6250,6260,6270,6280                  [1323]
6250 PLOT 400,366,8:DRAW 400,84:SOUND 1,80          [5702]
,2,7:PLOT 400,366,0:DRAW 400,84:IF x<=400
AND x+32>400 THEN 390 ELSE 6290
6260 PLOT 352,152,8:DRAW 572,152:SOUND 1,8          [6141]
0,2,7:PLOT 352,152,0:DRAW 572,152:IF y>=15
2 AND y-16<152 THEN 390 ELSE 6290
6270 PLOT 496,366,8:DRAW 496,84:SOUND 1,80          [7144]
,2,7:PLOT 496,366,0:DRAW 496,84:IF x<=496
AND x+32>496 THEN 390 ELSE 6290
6280 PLOT 352,248,8:DRAW 572,248:SOUND 1,8          [5174]
0,2,7:PLOT 352,248,0:DRAW 572,248:IF y>=24
8 AND y-16<248 THEN 390
6290 GOSUB 140                                      [921]
6300 IF JOY(0)=16 AND bombe$="geholt" THEN          [3805]
 6310 ELSE 6380
6310 FOR i=50 TO 200:SOUND 1,i,1,7:NEXT             [2939]
6320 OUT &BC00,8:OUT &BD00,1                        [1549]
6330 l=7:FOR i=1 TO 20:IF i>6 THEN l=l-0.5          [1958]

6340 SOUND 1,0,10,l,,,i:FOR ii=0 TO 26:BOR          [3906]
DER ii:NEXT:NEXT
6350 OUT &BC00,8:OUT &BD00,2                        [1548]
6360 TAGOFF:PEN 12:LOCATE 4,7:PRINT" ":LOC          [3847]
ATE 4,8:PRINT CHR$(219)
6370 bombe$="":mauer$="gesprengt"                   [2187]
6380 RETURN                                         [555]
6390 '----- RAUM 23 -----                           [1082]
6400 IF altraum=raum THEN 6500                      [1271]
6410 IF altraum2=21 THEN y=400                      [953]
6420 IF altraum2=24 THEN y=84                       [1105]
6430 xt=x:yt=y:x1=x:y1=y                            [1309]
6440 TAGOFF:PEN 1:GOSUB 7180:CLS#1                  [1376]
6450 LOCATE 1,1:PRINT n1$;n2$;                      [710]
6460 FOR b=1 TO 21:LOCATE 3,b:PRINT" ":LOC          [10649]
ATE 7,b:PRINT" ":LOCATE 12,b:PRINT" ":LOCA
TE 18,b:PRINT" ":NEXT:FOR b=3 TO 21:LOCATE
 16,b:PRINT" ":NEXT
6470 LOCATE 14,1:PRINT" ":LOCATE 14,2:PRIN          [6221]
T" ":LOCATE 14,3:PRINT" ":LOCATE 14,4:PRIN
T" ":LOCATE 15,3:PRINT" ":LOCATE 15,4:PRIN
T" "
6480 GOSUB 7170                                     [865]
6490 '                                              [117]
6500 altraum=raum                                   [651]
6510 IF y>400 THEN raum=21:altraum2=23              [2272]
6520 IF y<84 THEN raum=24:altraum2=23               [2511]
6530 RETURN                                         [555]
6540 '----- RAUM 24 -----                           [976]
6550 IF altraum=raum THEN 6670                      [1245]
6560 IF altraum2=23 THEN y=400                      [1388]
6570 IF altraum2=12 THEN x=608                      [626]
6580 xt=x:yt=y:x1=x:y1=y                            [1309]
6590 TAGOFF:PEN 1:GOSUB 7180:CLS#1                  [1376]
6600 LOCATE 1,1:PRINT n1$;n2$;                      [710]
6610 FOR a=3 TO 20:LOCATE a,18:PRINT" ";:L          [11003]
OCATE a,19:PRINT" ";:NEXT:FOR b=1 TO 17:LO
CATE 3,b:PRINT" ":NEXT:FOR b=1 TO 14:LOCAT
E 7,b:PRINT" ":NEXT:FOR b=1 TO 12:LOCATE 1
2,b:PRINT" ":NEXT
6620 FOR b=11 TO 17:LOCATE 15,b:PRINT" ":N          [8158]
EXT:FOR b=1 TO 7:LOCATE 16,b:PRINT" ":NEXT
:FOR a=17 TO 20:LOCATE a,6:PRINT" ";:LOCAT
E a,7:PRINT" ";:NEXT
6630 LOCATE 18,1:PRINT" ":LOCATE 18,2:PRIN          [7160]
T" ":LOCATE 13,11:PRINT" ":LOCATE 13,12:PR
INT" ":LOCATE 14,11:PRINT" ":LOCATE 14,12:
PRINT" "
6640 PEN 2:LOCATE 17,18:PRINT CHR$(149):LO          [4210]
CATE 17,19:PRINT CHR$(149)
6650 GOSUB 7170                                     [865]
6660 '                                              [117]
6670 altraum=raum                                   [651]
6680 IF y>400 THEN raum=23:altraum2=24              [1992]
6690 IF x>608 THEN raum=12:altraum2=24              [1693]
6700 RETURN                                         [555]
6710 '----- RAUM 25 -----                           [1135]
6720 IF altraum=raum THEN 6880                      [1243]
6730 x=0:xt=x:yt=y:x1=x:y1=y                        [1644]
6740 TAGOFF:PEN 1:GOSUB 7180:CLS#1                  [1376]
6750 LOCATE 1,1:PRINT n1$;n2$;                      [710]
6760 FOR a=1 TO 17:LOCATE a,18:PRINT" ":LO          [15945]
CATE a,19:PRINT" ":NEXT:FOR b=3 TO 19:LOCA
TE 18,b:PRINT" ":NEXT:FOR a=4 TO 17:LOCATE
 a,3:PRINT" ":LOCATE a,4:PRINT" ":NEXT:FOR
 a=8 TO 13:LOCATE a,5:PRINT" ":LOCATE a,6:
PRINT" ":LOCATE a,9:PRINT" ":LOCATE a,10
6770 PRINT" ":NEXT:FOR b=3 TO 16:LOCATE 3,          [10568]
b:PRINT" ":NEXT:FOR a=4 TO 6:LOCATE a,15:P
RINT" ":LOCATE a,16:PRINT" ":NEXT:FOR b=9
```

```
TO 16:LOCATE 7,b:PRINT" ":NEXT:FOR a=11 TO
 13:FOR b=11 TO 15:LOCATE a,b:PRINT" ":NEX
T:NEXT
6780 LOCATE 6,5:PRINT CHR$(206):LOCATE 7,6 [7731]
:PRINT CHR$(206):LOCATE 14,5:PRINT" ":LOCA
TE 14,6:PRINT CHR$(205):LOCATE 15,5:PRINT
CHR$(205)
6790 RESTORE 6810                          [757]
6800 READ xa,ya,za,pa:IF xa=0 THEN 6820 EL [4259]
SE PEN pa:LOCATE xa,ya:PRINT CHR$(za):GOTO
 6800
6810 DATA 2,11,212,8,4,11,211,8,7,5,210,9, [5219]
8,20,213,3,9,8,208,8,11,4,210,9,11,8,208,8
,17,11,212,8,19,11,211,8,12,16,213,3,0,0,0
,0
6820 IF spion$<>"geholt" THEN PEN 10:LOCAT [3685]
E 12,15:PRINT CHR$(249)
6830 IF gefaengnis$<>"geoeffnet" THEN PLOT [3519]
 320,238,13:DRAW 412,238:PLOT 320,236:DRAW
 412,236
6840 PLOT 160,336,6:DRAW 160,346:DRAW 252, [11216]
346:DRAW 252,304:DRAW 248,304:DRAW 248,344
:DRAW 164,344:DRAW 164,336:PLOT 288,366:DR
AW 288,324:DRAW 380,324:DRAW 380,366:DRAW
376,366:DRAW 376,326:DRAW 292,326:DRAW 292
,366
6850 p1=0:p2=0:p3=0:p4=0:pa1=0:pa2=0:pa3=0 [2654]
:pa4=0
6860 GOSUB 7170                            [865]
6870 '                                     [117]
6880 altraum=raum                          [651]
6890 IF x<0 THEN raum=16:altraum2=25       [2026]
6900 z=INT(RND(1)*20)+1:IF z=1 THEN p1=0 E [2735]
LSE IF z=2 THEN p1=8
6910 IF p1<>pa1 THEN PLOT 64,232,p1:DRAW 9 [3550]
2,232:IF p1=8 THEN SOUND 1,120,2,7
6920 IF p1=8 THEN IF x<100 AND y>=232 AND  [4600]
y-16<232 THEN PLOT 64,232,0:DRAW 92,232:GO
TO 390
6930 pa1=p1                                [110]
6940 GOSUB 140                             [921]
6950 z=INT(RND(1)*20)+1:IF z=1 THEN p2=0 E [2838]
LSE IF z=2 THEN p2=8
6960 IF p2<>pa2 THEN PLOT 544,232,p2:DRAW  [3936]
572,232:IF p2=8 THEN SOUND 1,120,2,7
6970 IF p2=8 THEN IF x>500 AND y>=232 AND  [5115]
y-16<232 THEN PLOT 544,232,0:DRAW 572,232:
GOTO 390
6980 pa2=p2                                [435]
6990 GOSUB 140                             [921]
7000 z=INT(RND(1)*20)+1:IF z=1 THEN p3=0 E [3602]
LSE IF z=2 THEN p3=8
7010 IF p3<>pa3 THEN PLOT 272,270,p3:DRAW  [4223]
272,240:IF p3=8 THEN SOUND 1,120,2,7
7020 IF p3=8 THEN IF y>200 AND y<300 AND x [4788]
<=272 AND x+32>272 THEN PLOT 272,270,0:DRA
W 272,240:GOTO 390
7030 pa3=p3                                [92]
7040 GOSUB 140                             [921]
7050 z=INT(RND(1)*20)+1:IF z=1 THEN p4=0 E [2126]
LSE IF z=2 THEN p4=8
7060 IF p4<>pa4 THEN PLOT 336,270,p4:DRAW  [3755]
336,240:IF p4=8 THEN SOUND 1,120,2,7
7070 IF p4=8 THEN IF y<300 AND y>=240 AND  [5743]
x<=336 AND x>336 THEN PLOT 336,270,0:DRAW
336,240:GOTO 390
7080 pa4=p4                                [688]
7090 RETURN                                [555]
7100 '----- FUEL -----                     [1555]
7110 fuel=fuel-1:PLOT -10,-10,1            [2422]
7120 TAG:MOVE 200,38:PRINT USING "##";fuel [1464]
;
7130 IF fuel<10 THEN SOUND 1,70,1,7        [1489]
7140 IF fuel=0 THEN ff=0:GOTO 390          [1475]
7150 PLOT -10,-10,7:RETURN                 [1610]
7160 '----- FARBEN -----                   [1693]
7170 INK 0,0:INK 1,26:INK 2,0:INK 3,12:INK [8315]
 4,10:INK 5,7:INK 6,0:INK 7,18:INK 8,6:INK
 9,10,15:INK 10,11:INK 11,15:INK 12,12:INK
 13,3:INK 14,0:LOCATE 1,1:RETURN
7180 FOR farben=0 TO 15:INK farben,0:NEXT: [3032]
RETURN
7190 '                                     [117]
7200 PAPER 0:PEN 1:MODE 1                  [1717]
7210 FOR i=500 TO 50 STEP -5:SOUND 1,i,2,7 [7891]
:NEXT:FOR i=50 TO 500 STEP 5:SOUND 1,i,2,7
:NEXT:FOR i=1 TO 50:SOUND 1,0,1,7,,,20:BOR
DER INT(RND(1)*27):FOR ii=1 TO 15:NEXT:NEX
T
7220 LOCATE 1,5:PRINT"Ihre Mission ist lei [5082]
der gescheitert."
7230 LOCATE 1,10:PRINT"Vielleicht schaffen [5897]
 Sie es das naechste Mal."
7240 FOR i=1 TO 100:i$=INKEY$:i$="":NEXT:C [1479]
ALL &BB18
7250 '                                     [117]
7260 MODE 1                                [506]
7270 LOCATE 8,12:PRINT"Noch einmal spielen [3153]
 (j/n)?"
7280 i$=INKEY$:i$=LOWER$(i$):IF i$="" THEN [2324]
 7280
7290 IF i$="j" THEN RUN                    [909]
7300 IF i$="n" THEN CALL 0                 [1269]
7310 GOTO 7280                             [395]
7320 '                                     [117]
7330 RETURN                                [555]
```



**Wenn Sie schon immer mal Tiriac spielen wollten, nur noch keine Gelegenheit dazu hatten. . .
Ihr CPC machts möglich!**




# TENNIS-

Wie schon der Name sagt, liegt hier ein bemerkenswertes Simulationsprogramm vor, bei dem sich bis zu sechs Spieler als Manager für Boris und Co. versuchen können.
Das Programm ist vollkommen Menue-gesteuert, wodurch bei den Eingaben eigentlich keine Probleme auftreten dürften. Einzige Ausname stellt die Abfrage dar, welcher Spieler in Aktion treten soll. Hier muß mit den Cursortasten gesteuert werden und mit COPY können Sie die Eingabe bestätigen.

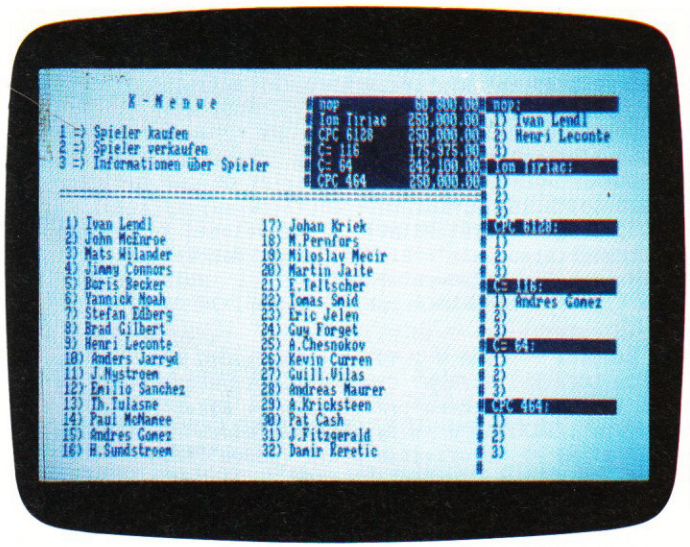
**Die fünf Hauptmenues:**
1) W – Menue
2) K-Menue
3) G-Menue
4) Z-Menue
5) E-Menue

**Das W-MENUE:**
Im W-Menue können Sie sich die Weltrangliste (bis Platz 32) ansehen, abspeichern und bereits gespeicherte einladen und verändern. Z.B. könnten Sie Boris Becker in Harald Junke umbenennen, ihm 60 Lenze auf den Buckel laden und ihn zum Grundlinienspezialisten umfunktionieren.

**Das K-MENUE:**
Mit dem K-Menue werden geschäftliche Dinge wie das Einkaufen von Spielern erledigt. Pro Turnier kann ein drei-Mann-Team aufgestellt werden. Außerdem können hier wichtige Informationen über die Spieler eingeholt werden.

## für 464-664-6128

```
10 '                                                    [117]
20 ' ********************                               [669]
30 ' **  T E I L   I  **                                [636]
40 ' ********************                               [669]
50 '                                                    [117]
60 CALL &BB4E:TROFF:TAGOFF:GOSUB 260                    [2260]
70 MODE 1:INK 0,3:BORDER 5:INK 1,24:INK 2,              [4053]
18:INK 3,9:PEN 1:PAPER 0
80 LOCATE 6,2:PEN 2:PAPER 3:PRINT"T E N N               [3419]
I S  -  M A N A G E R"
90 LOCATE 10,24:PRINT"<< ANY KEY TO CONT >              [2443]
>"
100 WINDOW 1,40,4,22:PEN 1:PAPER 0                      [2160]
110 PRINT"Das Ziel des Spiels ist schnell               [14513]
erkl[rt.Sie m^ssen durch Kaufen und Verkau
fen   von Spielern als Manager m]glichst v
iel Geld verdienen."
120 PRINT"Was wichtig ist,ist,da\ Sie wiss              [18022]
en,wie  man das Spiel bedient.Haupts[chlic
h k]n-nen Sie ^ber verschiedene Menues Inf
or -mationen erhalten bzw. Entscheidungen
  treffen.Es gibt f^nf Hauptmenues,von wo"
130 PRINT"aus Sie wiederum in Untermenues               [14576]
springenk]nnen.Diese 5 Hauptmenues habe ic
h nachden Anfangsbuchstaben ihrer jeweilig
en  Funktionen W,K,Z,E- und G-Menue genann
t.";
140 PRINT"(f^r W-eltrangliste,K-aufen,Z-ah               [3821]
len,E-nd und G-ame)"
150 PRINT"Wie Sie dorthin gelangen k]nnen,              [13374]
ist sehreinfach.Wollen Sie z.B.ins W-Menue
,dr^k-ken Sie die Taste W,wollen Sie ins K
-Me-nue,dr^cken Sie eben die Taste K."
160 WHILE INKEY$="" :WEND                               [1607]
170 CLS:PRINT"Ein Tip am Rande:Wenn es so               [13577]
aussieht,als w[ren Sie in einer Endlos-Sch
leife ge- fangen,kommen Sie durch Druck au
f ENTER oder durch die Eingabe von Nachbar
zahlen";
180 PRINT"(z.B.Sie m^ssen w[hlen,welchen S              [17143]
pieler  Sie kaufen wollen (1-32),dann sind
 0 und33 Nachbarzahlen) wieder zum Hauptme
nue oder anderen Eingaberoutinen zur^ck."
190 PRINT"            Das Spiel wird immer              [23816]
 mit    sechs Teams gespielt,um die Spannu
ng zu erh]hen.Wenn Sie alleine oder zu zwe
it  spielen m]chten,werden die restlichen
  Teams vom Computer ^bernommen und die
K[ufe bzw.Verk[ufe vollautomatisch get[-";
200 PRINT"tigt.Viel Spa\ beim Tiriac-Nacha              [7406]
hmen !!":PRINT:PRINT"  MARTIN  HUEMMERICH"
210 WINDOW 1,40,1,25:LOCATE 9,24:WHILE INK              [2902]
EY$="":WEND
220 RUN"tennis                                          [392]
230 '                                                   [117]
240 '--Umlaute--                                        [1067]
250 '                                                   [117]
260 SYMBOL AFTER 91                                     [1213]
270 SYMBOL 91,&X1101100,0,&X1111000,&X1100              [4382]
,&X1111100,&X11001100,&X1110110
280 SYMBOL 93,&X1100110,0,&X111100,&X11001              [3138]
10,&X1100110,&X1100110,&X111100
290 SYMBOL 94,&X1100110,0,&X1100110,&X1100              [4794]
110,&X1100110,&X1100110,&X111110
300 SYMBOL 92,&X111100,&X1100110,&X1100110              [5398]
,&X1111100,&X1100110,&X1100110,&X1101100,&
X1100000
310 RETURN                                              [555]
```




# MANAGER

```
10 ' ************************           [978]
20 ' *  RHEIN-SOFT presents:  *         [1829]
30 ' *     Tennis-Manager     *         [955]
40 ' *     Juni-August 86     *         [841]
50 ' *                        *         [854]
60 ' ************************           [978]
70 '                                    [117]
80 ' written by Martin Huemmerich       [3132]
90 '                                    [117]
100 SYMBOL 254,30,51,33,35,38,56,96,192:SY [7033]
MBOL 255,120,204,132,196,100,28,6,3:SYMBOL
 124,255,255,255,255,255,255,255,255:KEY D
EF 15,1,212,213:KEY DEF 7,1,214,215
110 MODE 1:WINDOW 4,35,4,25:INK 0,2:BORDER [10337]
 2:INK 1,23:SPEED INK 2,2:INK 2,9,12:INK 3
,12,9:RESTORE 140:FOR w=1 TO 7:READ titel$
(w):NEXT:ENV 1,1,0,10,10,-1,3
120 DIM player$(32),spieler$(32),alter(32) [10790]
,posi(32),weise(32),boden(32),we(32),preis
(32),verk(32),paar(32),gesamt(32),punkt(32
),stpunkt(32),preisgeld(16)
130 RESTORE 130:FOR w=1 TO 16:READ preisge [4030]
ld(w):NEXT:DATA 10,5,0,3,0,0,0,1,0,0,0,0,0
,0,0,1
140 DATA "!!!!! !!!W W    ! W    ! ! V!!!T [3315]
150 DATA "  !   !   !W   ! !W   !   !T    [2030]
"
160 DATA "  !   !   !UW  ! !UW  ! ! !W    [3939]
"
170 DATA "  !   !!!! ! UW ! ! UW ! ! U!!!W [2674]
180 DATA "  !   !    !  UW! !  UW! !    U! [2839]
190 DATA "  !   !    !   U! !   U! !    V! [2261]
200 DATA "  !   !!!T !    U !    U ! U!!!T [3342]
210 PAPER 0:FOR wdh=1 TO 7:i=0              [1781]
220 f=f+1:i=i+1:IF f MOD 2 THEN PEN 2 ELSE [1581]
 PEN 3
230 IF i>LEN(titel$(wdh)) THEN IF wdh<=7 T [3551]
HEN NEXT ELSE 250
240 PRINT MID$(titel$(wdh),i,1);:GOTO 220  [1850]
250 WINDOW 1,40,1,25:LOCATE 14,13:PEN 1:PR [11167]
INT"M A N A G E R":LOCATE 11,13:PRINT CHR$
(254):LOCATE 29,13:PRINT CHR$(255):LOCATE
15,17:PRINT"written by:":LOCATE 12,20:PRIN
T"MARTIN HUEMMERICH":t$="<< SPACE to go on
 >>"
260 RESTORE 3110:GOSUB 3100:FOR w=1 TO 20: [3125]
i=i+1:IF i MOD 2 THEN PEN 2 ELSE PEN 3
270 LOCATE 9+w,25:PRINT MID$(t$,w,1):NEXT: [4078]
LOCATE 1,1:WHILE INKEY$<>" ":WEND
280 '                                   [117]
290 '--Vorbereitungen--                 [2537]
```

**Das G-MENUE:**
Dieses ist das GAME-Menue. Nachdem man das Turnier, auf dem gespielt werden soll, ausgewählt hat, treten die Spieler im K.O.–System gegeneinander an.
Nach dem Finale werden die Prämien ausgegeben und die neue Weltrangliste berechnet.

**Das Z-MENUE**
enthält die Punkte Gehälter sehen, also anschauen, was Ihre Spieler verdienen, Gehälter zahlen und Gehälter kürzen. Die Gehälter sollten regelmäßig bezahlt werden, sonst wird bei Benutzung des E-Menues abgerechnet, was zu 99,9 % ein böses Erwachen gibt.
Wenn Sie die Gehälter kürzen wollen, ist das immer ein Spiel mit dem Feuer, da die Spieler auch selbstständig kündigen können.

**Das E-MENU:**
Hiermit können Sie das Programm beenden, den Spielstand abspeichern oder einladen und zu den anderen Menuepunkten zurückkehren.

Das Menue wechseln können Sie meistens durch einfaches Drücken der jeweiligen Taste (G für G-Menu).

Wenn Sie mit weniger als den sechs möglichen Leuten spielen wollen, übernimmt der Computer die restlichen Spieler.

**Zum Abtippen**
Nachdem Sie die beiden Listings eingegeben und abgespeichert haben,können Sie das Spiel mit RUN »tennisma« starten. Benutzer von Diskettenlaufwerken können die in den Zeilen 770, 3260 und 3330 vorkommenden PRINT-Anweisungen, die sich auf den Kassettenrekorder beziehen, natürlich vergessen.

(Martin Hümmerich/SH)

```
300 '                                   [117]
310 TROFF:TAGOFF:SPEED WRITE 1:SPEED KEY 2 [10510]
0,1:KEY 140,"mode 2:pen 1:paper 0:list ":M
ODE 2:INK 1,0:INK 0,26:BORDER 26:FOR w=1 T
O 6:zaehl(w)=0:NEXT
320 FOR w=1 TO 32:verk(w)=0:NEXT:FOR w=1 T [5474]
O 6:FOR s=1 TO 3:hab$(w,s)="":NEXT s,w:zae
hlwelt=0
330 RESTORE 330:FOR s=1 TO 4:READ tun$(s): [6851]
NEXT:DATA "        P  A  R  I  S","    W
I  M  B  L  E  D  O  N"," F L U S H I N G
  M E A D O W"," A U S T R A L I A N   O P
 E N"
340 FOR s=1 TO 6:READ sp$(s):NEXT:DATA Joy [5423]
ce,Ion Tiriac,CPC 6128,C= 116,C= 64,CPC 46
4
350 FOR s=1 TO 6:ge(s)=250000:NEXT:RESTORE [7948]
 360:FOR wdh=1 TO 32:READ player$(wdh),alt
er(wdh),posi(wdh),weise(wdh),boden(wdh):NE
XT
360 DATA Ivan Lendl,24,1,1,2,John McEnroe, [6860]
25,2,2,3,Mats Wilander,22,3,1,1,Boris Beck
er,18,5,3,3,Jimmy Connors,33,4,2,2
370 DATA Yannick Noah,27,6,2,3,Th.Tulasne, [4271]
19,13,1,1,Guy Forget,20,24,2,1,Henri Lecon
te,22,9,3,2,Stefan Edberg,18,7,2,2
380 DATA Anders Jarryd,26,10,2,1,J.Nystroe [6573]
m,23,11,2,2,A.Kricksteen,19,29,1,1,E.Telts
cher,27,21,2,2,Kevin Curren,23,26,3,3
```

```
390 DATA Miloslav Mecir,23,19,2,2,Tomas Sm [6940]
id,31,22,2,3,Andreas Maurer,28,28,2,1,Dami
r Keretic,26,32,2,3,Andres Gomez,24,15,2,2
400 DATA Pat Cash,20,30,3,3,J.Fitzgerald,3 [5909]
3,31,2,1,Martin Jaite,21,20,2,3,Johan Krie
k,28,17,2,1,A.Chesnokov,19,25,1,2
410 DATA Emilio Sanchez,24,12,3,2,M.Pernfo [7776]
rs,23,18,2,2,Guill.Vilas,23,27,3,3,Eric Je
len,19,23,2,2,H.Sundstroem,25,16,2,1
420 DATA Brad Gilbert,23,8,3,2,Paul McName [1874]
e,30,14,1,1
430 FOR w=1 TO 32:punkt(w)=33-posi(w):NEXT [3211]
440 ' [117]
450 '--Bildschirm-Maske-- [1726]
460 ' [117]
470 FOR s=1 TO 25:LOCATE 62,s:PRINT"#":NEX [5166]
T:FOR s=1 TO 6:LOCATE 37,s:PRINT"#":NEXT:L
OCATE 1,7:PRINT STRING$(61,"=")
480 menue$="W":WINDOW 1,61,8,25:WINDOW #1, [9792]
1,36,1,6:WINDOW #2,63,80,1,25:WINDOW #3,38
,61,1,6:FOR w=0 TO 2:PAPER #w,0:PEN #w,1:N
EXT:PAPER #3,1:PEN #3,0
490 GOSUB 1710:GOSUB 1530 [1093]
500 CLS#3:GOSUB 1780:LOCATE#1,6,3:PRINT#1, [10274]
"Die aktuelle Weltrangliste":LOCATE#1,7,4:
PRINT#1,"wird gerade ausgedruckt.":LOCATE
#1,5,5:PRINT#1,STRING$(28,"-"):GOSUB 1580:
FOR wa=1 TO 3000:NEXT:CLS
510 ' [117]
520 '--Spielbeginn-- [1677]
530 ' [117]
540 EVERY 500,3 GOSUB 2590:EVERY 2600,2 GO [14107]
SUB 3470:IF menue$="W" THEN 600 ELSE IF me
nue$="K" THEN 990 ELSE IF menue$="G" THEN
s=REMAIN(3):GOTO 1840 ELSE IF menue$="E" T
HEN s=REMAIN(3):GOTO 3180 ELSE IF menue$="
Z" THEN 3540
550 ' [117]
560 ' *************** [345]
570 ' ** W-Menue ** [904]
580 ' *************** [345]
590 ' [117]
600 CLS#1:CLS:LOCATE #1,10,1:PRINT#1," W - [3625]
 M e n u e ":PRINT#1
610 PRINT#1,"1 => Weltrangliste ausdrucken [11543]
":PRINT#1,"2 => Weltrangliste laden":PRINT
#1,"3 => Weltrangliste abspeichern":PRINT#
1,"4 => Spieler um[ndern"
620 ein$="":WHILE ein$="":ein$=INKEY$:WEND [3799]
:GOSUB 1660
630 IF ein$="1" THEN 670 ELSE IF ein$="2" [6357]
THEN 720 ELSE IF ein$="3" THEN 770 ELSE IF
 ein$="4" THEN 820 ELSE 620
640 ' [117]
650 '--W.ausdrucken-- [2471]
660 ' [117]
670 GOSUB 1580 [831]
680 LOCATE 13,18:PRINT" << Press SPACE-BAR [5444]
 to go on >> ":WHILE INKEY$<>" ":WEND:CLS:
GOTO 620
690 ' [117]
700 '--alte W.laden-- [904]
710 ' [117]
720 FOR w=1 TO 6:FOR s=1 TO 3:hab$(w,s)="" [12107]
:NEXT s,w:FOR w=1 TO 32:verk(w)=0:NEXT:CLS
#1:LOCATE #1,8,2:PRINT#1,"PLAY druecken...
":LOCATE #1,9,4:PRINT#1,"Dateiname ? ";:LI
NE INPUT#1,d$:d$="!"+d$:OPENIN d$
730 FOR w=1 TO 32:INPUT#9,player$(w):INPUT [10669]
#9,alter(w):INPUT#9,posi(w):INPUT#9,weise(
w):INPUT#9,boden(w):INPUT#9,we(w):NEXT:CLO
SEIN:CLS:CLS#1:GOSUB 1530:GOSUB 1490:GOTO
600
740 ' [117]
750 '--neue W.abspeichern-- [1308]
760 ' [117]
770 CLS#1:LOCATE #1,5,2:PRINT#1,"REC & PLA [5273]
Y druecken...":LOCATE #1,9,4:PRINT#1,"Date
iname ? ";:LINE INPUT#1,d$:d$="!"+d$:OPENO
UT d$
780 FOR w=1 TO 32:PRINT#9,player$(w):PRINT [10140]
#9,alter(w):PRINT#9,posi(w):PRINT#9,weise(
w):PRINT#9,boden(w):PRINT#9,we(w):NEXT:CLO
SEOUT:CLS:CLS#1:GOTO 600
790 ' [117]
800 '--Spieler umaendern-- [2544]
810 ' [117]
820 GOSUB 1580 [831]
830 CLS#1:PRINT#1:PRINT#1," Auf welchem We [19180]
ltranglistenplatz":PRINT#1,"steht der Spie
ler,den Sie um[ndern":PRINT#1," wolle
n ";:INPUT#1,zahl:IF zahl<1 OR zahl>32 OR
zahl<>INT(zahl) THEN PRINT#1," Falsche
 Eingabe !!":FOR w=1 TO 1000:NEXT:GOTO 830
840 LOCATE 1,1:PRINT CHR$(7):CLS#1:PRINT#1 [13346]
,"1 => Namen [ndern":PRINT#1,"2 => Alter [
ndern":PRINT#1,"3 => Spielweise [ndern":PR
INT#1,"4 => Bevorzugten Boden [ndern":PRIN
T#1,"5 => Neue Nummer":PRINT#1,"6 => Zur^c
k zum W-Menue"
850 ein$=INKEY$:IF ein$<"1" OR ein$>"6" TH [2043]
EN 850 ELSE za=VAL(ein$)
860 ON za GOTO 880,890,900,920,830,870 [1471]
870 CLS#1:CLS:GOTO 600 [922]
880 CLS#1:LOCATE #1,16,3:PRINT#1,"..... [7073]
.....":LOCATE #1,4,3:INPUT#1,"Neuer Name
 ";a$:IF LEN(a$)>14 THEN PRINT#1,"Name zu
lang !":FOR w=1 TO 2000:NEXT:GOTO 880
881 FOR w=1 TO 6:FOR s=1 TO 3:IF hab$(w,s) [8068]
=player$(we(zahl)) THEN hab$(w,s)=a$:LOCAT
E#2,5,w*4-3+s:PRINT#2,SPACE$(14);:GOSUB 14
90
882 NEXT s,w:player$(we(zahl))=a$:GOTO 840 [2681]
890 CLS#1:LOCATE #1,10,3:INPUT#1,"Neues Al [13297]
ter ";alter(we(zahl)):IF alter(we(zahl))<1
0 OR alter(we(zahl))>49 THEN PRINT#1:PRINT
#1," Verkaspern kann ich mich selber !":F
OR wa=1 TO 1500:NEXT:GOTO 890 ELSE 840
900 CLS#1:PRINT#1," Versch. Spielweise [11741]
n ":PRINT#1:PRINT#1,"1 => Grundlinie (defe
nsiv)":PRINT#1,"2 => Sowohl...als auch (Mi
ttelding)":PRINT#1,"3 => Netz (agressiv)""
910 ein$=INKEY$:IF ein$<"1" OR ein$>"3" TH [3455]
EN 910 ELSE weise(we(zahl))=VAL(ein$):GOTO
 840
920 CLS#1:PRINT#1," Verschiedene Bode [12482]
ntypen ":PRINT#1:PRINT#1,"1 => Langsamer B
oden wird bevorzugt":PRINT#1,"2 => Mittler
er Boden wird bevorzugt":PRINT#1,"3 => Sch
neller Boden wird bevorzugt"
930 ein$=INKEY$:IF ein$<"1" OR ein$>"3" TH [4890]
EN 930 ELSE boden(we(zahl))=VAL(ein$):GOTO
 840
940 ' [117]
950 ' *************** [345]
960 ' ** K-Menue ** [733]
970 ' *************** [345]
980 ' [117]
990 CLS#1:LOCATE #1,10,1:PRINT#1," K - M e [3983]
 n u e ":PRINT#1
1000 PRINT#1,"1 => Spieler kaufen":PRINT#1 [8976]
,"2 => Spieler verkaufen":PRINT#1,"3 => In
formationen ^ber Spieler"
1010 ein$="":WHILE ein$="":ein$=INKEY$:WEN [3799]
D:GOSUB 1660
1020 IF ein$="1" THEN 1060 ELSE IF ein$="2 [4258]
" THEN 1310 ELSE IF ein$="3" THEN 2820 ELS
E 1010
1030 ' [117]
1040 '--Spieler kaufen-- [1649]
1050 ' [117]
1060 GOSUB 1580:CLS#1 [1040]
1070 IF mitspieler<2 THEN cur=1:GOTO 1120 [2024]
1080 PRINT#1:PRINT#1,"Sagen Sie mir,":PRIN [7916]
T#1,"Welcher Spieler":PRINT#1," Sie sind
??":cur=2:LOCATE#1,34,2:PRINT#1,"X"
1090 LOCATE#1,34,cur:i$=INKEY$:IF i$="" TH [6447]
EN 1090 ELSE LOCATE 1,1:PRINT CHR$(7):PRIN
T#1," ":IF i$=CHR$(240) THEN cur=cur-1:IF
cur<1 THEN cur=1
```

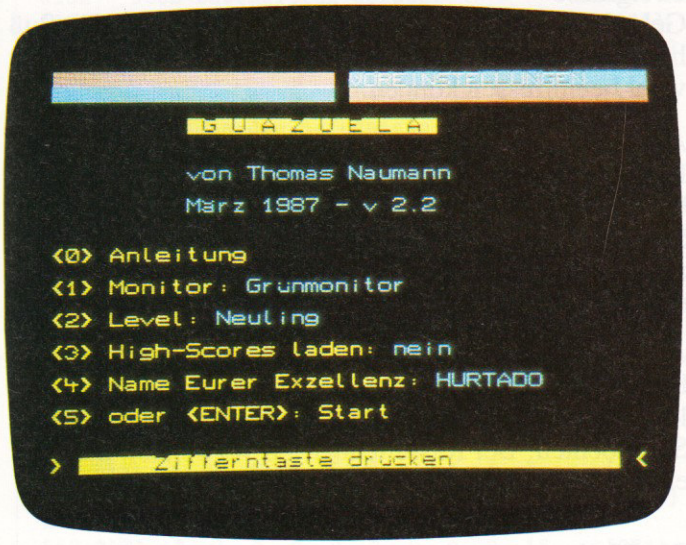
```
1100 IF i$=CHR$(241) THEN cur=cur+1:IF cur [6053]
>mitspieler THEN cur=mitspieler
1110 LOCATE#1,34,cur:PRINT#1,"X":IF i$<>CH [2600]
R$(224) THEN 1090
1120 IF ein$="2" AND menue$<>"Z" THEN 1320 [4358]
 ELSE IF menue$="Z" THEN RETURN
1130 FOR w=1 TO 32:preis(we(w))=(35-w)*300 [5003]
0+((40-alter(we(w)))*300)+INT(RND*20-5)*10
00:NEXT
1140 CLS#1:PRINT#1                          [872]
1150 PRINT#1," Auf welchem Weltranglistenp [16002]
latz":PRINT#1," steht der Spieler,den Sie
kaufen":PRINT#1,"  wollen,";sp$(cur);" ";:
INPUT#1,zahl:IF zahl<0 OR zahl>33 OR zahl<
>INT(zahl) THEN PRINT#1,"     Falsche Eing
abe !!":FOR w=1 TO 1000:NEXT:GOTO 1140
1160 IF zahl=0 THEN 990 ELSE IF zahl=33 TH [4067]
EN CLS#1:GOTO 1070
1170 IF hab$(cur,3)<>"" AND hab$(cur,3)<>S [10927]
PACE$(14) AND hab$(cur,2)<>"" AND hab$(cur
,1)<>"" AND hab$(cur,2)<>SPACE$(14) AND ha
b$(cur,1)<>SPACE$(14) THEN PRINT#1,"Sie ha
ben schon 3 Spieler !":FOR wa=1 TO 2000:NE
XT:GOTO 1140
1180 IF verk(we(zahl))<>0 THEN PRINT#1,"De [8373]
r ist schon verkauft !":FOR wa=1 TO 2000:N
EXT:GOTO 1140
1190 CLS#1:PRINT#1,"So,Sie m]chten also de [8057]
n ":PRINT#1,player$(we(zahl));" kaufen..."
:PRINT#1:PRINT#1,"Er kostet";preis(we(zahl
));"US-Dollar."
1200 PRINT#1,"Kaufen Sie ihn ?(J-a/N-ein/Z [8046]
-ur^ck)";:a$="":WHILE a$<>"J" AND a$<>"N"
AND a$<>"Z":a$=INKEY$:a$=UPPER$(a$):WEND:I
F a$="Z" THEN 990 ELSE IF a$="N" THEN 1140
1210 zaehl(cur)=zaehl(cur)+1:IF zaehl(cur) [3409]
>3 THEN zaehl(cur)=3
1220 FOR tej=3 TO 1 STEP -1:IF hab$(cur,te [6232]
j)="" OR hab$(cur,tej)=SPACE$(14) THEN zae
hl(cur)=tej
1230 NEXT                                   [350]
1240 IF preis(we(zahl))>ge(cur) THEN PRINT [8017]
#1,"Er ist doch zu teuer !":FOR wa=1 TO 20
00:NEXT:GOTO 1140
1250 ge(cur)=ge(cur)-preis(we(zahl)):GOSUB [7386]
 1780:hab$(cur,zaehl(cur))=player$(we(zahl
)):verk(we(zahl))=1:GOSUB 1490
1260 posi1(cur,zaehl(cur))=posi(we(zahl)): [11201]
gehalt(cur,zaehl(cur))=INT(preis(we(zahl))
/5):preis1(cur,zaehl(cur))=preis(we(zahl))
-8000-(INT(RND*80)*100)
1270 GOTO 1140                              [311]
1280 '                                      [117]
1290 '--Spieler verkaufen--                 [1846]
1300 '                                      [117]
1310 GOTO 1060                              [311]
1320 FOR w=1 TO 32:preis(we(w))=(35-w)*300 [4038]
0+((40-alter(we(w)))*300)-INT(RND*3+6)*100
0:NEXT
1330 CLS#1:PRINT#1                          [872]
1340 PRINT#1," Auf welchem Weltranglistenp [17789]
latz":PRINT#1,"steht der Spieler,den Sie v
erkaufen":PRINT#1,"  wollen,";sp$(cur);" "
;:INPUT#1,zahl:IF zahl<0 OR zahl>33 OR zah
l<>INT(zahl) THEN PRINT#1,"     Falsche Ei
ngabe !!":FOR w=1 TO 1000:NEXT:GOTO 1140
1350 IF zahl=0 THEN 990 ELSE IF zahl=33 TH [4067]
EN CLS#1:GOTO 1070
1360 IF hab$(cur,1)<>player$(we(zahl)) AND [13663]
 hab$(cur,2)<>player$(we(zahl)) AND  hab$(
cur,3)<>player$(we(zahl)) THEN PRINT#1,"Ha
ha,den haben Sie zwar gar nicht,  aber das
 sage ich Ihnen ja nicht !":FOR wa=1 TO 30
00:NEXT:GOTO 1330
1370 CLS#1:PRINT#1,"So,Sie m]chten also de [15403]
n ":PRINT#1,player$(we(zahl));" verschache
rn...":PRINT#1:PRINT#1,"Sie w^rden fuer ih
n";preis(we(zahl));"US-Dollar";"bekommen.V
erkaufen Sie ihn fuer die-";
1380 PRINT#1," sen Preis?(J-a/N-ein/Z-ur^c [3669]
k)";
1390 a$="":WHILE a$<>"J" AND a$<>"N" AND a [5089]
$<>"Z":a$=INKEY$:a$=UPPER$(a$):WEND:IF a$=
"Z" THEN 990 ELSE IF a$="N" THEN 1330
1400 zaehl(cur)=zaehl(cur)-1:IF zaehl(cur) [3665]
<1 THEN zaehl(cur)=1
1410 merk=0:FOR w=1 TO 3:IF hab$(cur,w)=pl [4259]
ayer$(we(zahl)) THEN merk=w
1420 NEXT                                   [350]
1430 ge(cur)=ge(cur)+preis(we(zahl)):GOSUB [2505]
 1780
1440 FOR q=1 TO 32:IF player$(we(q))=hab$( [4225]
cur,merk) THEN verk(we(q))=0 ELSE NEXT
1450 hab$(cur,merk)=SPACE$(14):GOSUB 1490: [5365]
posi1(cur,merk)=0:gehalt(cur,merk)=0:preis
1(cur,merk)=0:GOTO 1330
1460 '                                      [117]
1470 '--Anzeige WINDOW 2--                  [3026]
1480 '                                      [117]
1490 FOR k1=1 TO 6:FOR k2=1 TO 3:LOCATE#2, [6876]
5,k1*4-3+k2:PRINT#2,hab$(k1,k2);:NEXT k2,k
1:RETURN
1500 '                                      [117]
1510 '--Anzeige der Spielernamen--          [4218]
1520 '                                      [117]
1530 CLS#2:FOR w=1 TO 6:LOCATE#2,1,w*4-3:P [5744]
RINT#2,STRING$(18,"|");:LOCATE#2,1,w*4-3:P
RINT#2,CHR$(24)" ";sp$(w);": "CHR$(24):NEX
T
1540 FOR w=1 TO 6:FOR s=1 TO 3:LOCATE#2,1, [4100]
w*4-3+s:PRINT#2,s;CHR$(8);")":NEXT s,w:RET
URN
1550 '                                      [117]
1560 '--Ausdrucken der Weltrangliste--      [2008]
1570 '                                      [117]
1580 CLS:PRINT CHR$(7):FOR spi=1 TO 32:FOR [3468]
 te=1 TO 32
1590 IF posi(te)=spi THEN we(spi)=te:IF ve [7747]
rk(te)=0 THEN PRINT posi(te);CHR$(8);") ";
player$(te) ELSE PRINT CHR$(24);posi(te);C
HR$(8);") ";player$(te);" "CHR$(24)
1600 NEXT:IF spi=16 THEN WINDOW 30,61,9,25 [2125]
1610 NEXT:WINDOW 1,61,8,25                  [1529]
1620 RETURN                                 [555]
1630 '                                      [117]
1640 '--Neues Menue ?--                     [692]
1650 '                                      [117]
1660 ein$=UPPER$(ein$):LOCATE 1,1:PRINT CH [6427]
R$(7):IF ein$="K" OR ein$="G" OR ein$="E"
OR ein$="W" OR ein$="Z" THEN menue$=ein$:G
OTO 540
1670 RETURN                                 [555]
1680 '                                      [117]
1690 '--Abfrage Mitspieler--                [1494]
1700 '                                      [117]
1710 LOCATE #1,3,3:INPUT#1,"Wieviele Mitsp [5703]
ieler (1-6) ";m$:IF m$<"1" OR m$>"6" OR LE
N(m$)>1 THEN 1710 ELSE mitspieler=VAL(m$):
CLS#1
1720 FOR w=1 TO mitspieler:PRINT#1,w;CHR$( [3585]
8);") ........."
1730 LOCATE #1,5,w:LINE INPUT#1,sp$(w):l=L [4525]
EN(sp$(w)):IF l>10 OR l<3 THEN 1730
1740 NEXT:CLS#1:RETURN                      [942]
1750 '                                      [117]
1760 '--Anzeige WINDOW 3--                  [3189]
1770 '                                      [117]
1780 FOR k3=1 TO 6:LOCATE#3,2,k3:PRINT#3,s [6110]
p$(k3):LOCATE#3,12,k3:PRINT#3,USING"##,###
,###.##";ge(k3);:NEXT:RETURN
1790 '                                      [117]
1800 ' ***************                      [345]
1810 ' **  G-Menue  **                      [1144]
1820 ' ***************                      [345]
1830 '                                      [117]
1840 RESTORE 1860:CLS#1:CLS:LOCATE#1,10,1: [6004]
PRINT#1," G - M e n u e ":FOR w=1 TO 6:pra
emie(w)=0:NEXT
1850 READ a,b:IF a=-2 THEN READ c:MOVE b,c [4648]
:GOTO 1850 ELSE IF a=-1 THEN 1870 ELSE DRA
```

```
W a,b,1:GOTO 1850
1860 DATA -2,48,48,448,48,360,272,136,272, [5928]
48,48,-2,96,48,168,272,-2,392,48,320,272,-
2,240,128,240,240,-2,160,240,331,240,-2,12
3,128,368,128,-1,-1
1870 FOR sp=135 TO 350 STEP 5:MOVE sp,176: [7382]
DRAW sp,200:NEXT:FOR ze=176 TO 200 STEP 4:
MOVE 135,ze:DRAW 350,ze:NEXT
1880 RESTORE 1880:FOR w=1 TO 9:READ t,d:SO [5889]
UND 1,t,d/2,15,1:NEXT:DATA 478,40,426,40,3
79,40,426,40,379,40,358,40,379,40,358,40,3
38,100
1890 LOCATE 9,17:PRINT" Wo m]chten Sie spi [20181]
elen? Bitte w[hlen Sie ! ":PRINT#1:PRINT#1
,"1 => Paris            (Sand)":PRINT#1,"2
 => Wimbledon        (Rasen)":PRINT#1,"3 =
> Flushing Meadow  (Kunstbelag)":PRINT#1,"
4 => Australian Open  (Rasen)"
1900 ein$="":WHILE ein$="":ein$=INKEY$:WEN [3799]
D:GOSUB 1660
1910 IF ein$<"1" OR ein$>"4" THEN 1900 ELS [3191]
E einga=VAL(ein$):tun=einga
1920 IF einga=2 THEN einga=3 ELSE IF einga [3980]
=4 THEN einga=3 ELSE IF einga=3 THEN einga
=2
1930 PRINT#1,CHR$(24):CLS#1:PRINT#3,CHR$(2 [6440]
4):CLS#3:GOSUB 1780:LOCATE#1,3,2:PRINT#1,t
un$(tun):finale=16
1940 LOCATE #1,10,5:PRINT#1,finale;CHR$(8) [5017]
;"-tel Finale ":CLS:IF finale=16 THEN 1950
 ELSE 2020
1950 LOCATE 18,2:PRINT" T h e   6   T e a [11759]
m s : ":FOR w=1 TO 6:stsp$(w)=" "+UPPER$(s
p$(w))+" ":NEXT:LOCATE 3,5:PRINT stsp$(1):
LOCATE 23,5:PRINT stsp$(2):LOCATE 43,5:PRI
NT stsp$(3)
1960 LOCATE 3,12:PRINT stsp$(4):LOCATE 23, [4815]
12:PRINT stsp$(5):LOCATE 43,12:PRINT stsp$
(6)
1970 FOR w=1 TO 3:FOR s=1 TO 3:LOCATE (w-1 [5441]
)*20+1,6+s:PRINT s;CHR$(8)") ";hab$(w,s):N
EXT s,w
1980 FOR w=4 TO 6:FOR s=1 TO 3:LOCATE (w-4 [12226]
)*20+1,13+s:PRINT s;CHR$(8)") ";hab$(w,s):
NEXT s,w:LOCATE 15,18:PRINT" << Hit any Ke
y to Continue >> ":FOR w=1 TO 32:paar(w)=0
:NEXT:CALL &BB06:CLS
1990 '                                  [117]
2000 '-Die Paarungen per RND ermitteln-  [1482]
2010 '                                  [117]
2020 LOCATE 23,1:PRINT" Die Paarungen: ":P [2817]
RINT
2030 FOR wdh=1 TO finale*2 STEP 2        [1209]
2040 IF finale<16 THEN 2100              [1260]
2050 r=INT(RND*finale*2+1):FOR te=1 TO wdh [4233]
:IF r=paar(te) THEN 2050
2060 NEXT:paar(wdh)=r                    [1167]
2070 r1=INT(RND*finale*2+1):FOR te=1 TO wd [3911]
h:IF r1=paar(te) THEN 2070
2080 NEXT:paar(wdh+1)=r1                 [1486]
2090 IF finale=16 THEN spieler$(wdh)=playe [4260]
r$(paar(wdh)):spieler$(wdh+1)=player$(paar
(wdh+1))
2100 PRINT wdh/2+0.5;CHR$(8);") ";spieler$ [2369]
(wdh);" - ";spieler$(wdh+1):NEXT
2110 '                                  [117]
2120 '--Die Ergebnisse berechnen--       [1477]
2130 '                                  [117]
2140 FOR w=1 TO 32:gesamt(w)=0:NEXT:LOCATE [5333]
 1,1:PRINT CHR$(7):LOCATE 22,1:PRINT" Die
Ergebnisse: :PRINT
2150 FOR wdh=1 TO finale*2 STEP 2:diff=we( [7741]
paar(wdh))-we(paar(wdh+1)):diff=INT(diff*1
.3):IF weise(paar(wdh))=einga THEN diff=di
ff-INT(RND*3+3)
2160 IF weise(paar(wdh+1))=einga THEN diff [4479]
=diff+INT(RND*3+3)
2170 IF ABS(weise(paar(wdh))-einga)=1 THEN [3381]
 diff=diff-2
2180 IF ABS(weise(paar(wdh+1))-einga)=1 TH [2103]
EN diff=diff+2
2190 IF boden(paar(wdh+1))=einga THEN diff [4631]
=diff+INT(RND*4+5)
2200 IF boden(paar(wdh))=einga THEN diff=d [3213]
iff-INT(RND*4+5)
2210 IF ABS(boden(paar(wdh))-einga)=1 THEN [4149]
 diff=diff-3
2220 IF ABS(boden(paar(wdh+1))-einga)=1 TH [2944]
EN diff=diff+3
2230 IF alter(paar(wdh))>alter(paar(wdh+1) [2905]
) THEN diff=diff+2 ELSE diff=diff-2
2240 IF we(paar(wdh))>we(paar(wdh+1)) THEN [1732]
 2270
2250 IF RND>0.8 THEN diff=diff-(RND*20-40) [3196]
 ELSE diff=diff-(RND*13-7)
2260 GOTO 2280                           [339]
2270 IF RND>0.8 THEN diff=diff+(RND*20-40) [3447]
 ELSE diff=diff+(RND*13-7)
2280 IF diff<1 THEN sieger=1 ELSE sieger=2 [1886]
2290 satzanzahl=INT(RND*3+3)             [3725]
2300 IF satzanzahl=3 THEN ausn=0:ausn1=0 E [4405]
LSE IF satzanzahl=4 THEN ausn=INT(RND*(sat
zanzahl-1)+1):ausn1=0 ELSE 2320
2310 GOTO 2330                           [345]
2320 ausn=INT(RND*(satzanzahl-1)+1):ausn1= [4530]
INT(RND*(satzanzahl-1)+1):IF ausn1=ausn TH
EN 2320
2330 FOR satz=1 TO satzanzahl:IF satz=ausn [4096]
 OR satz=ausn1 THEN 2340 ELSE 2360
2340 pl(3-sieger)=6:pl(sieger)=INT(RND*7): [3343]
IF pl(sieger)>4 THEN pl(3-sieger)=7
2350 GOTO 2380                           [399]
2360 pl(sieger)=6:pl(3-sieger)=INT(RND*7): [5590]
IF pl(3-sieger)>4 THEN pl(sieger)=7
2370 IF pl(1)>pl(2) THEN gesamt(wdh)=gesam [4615]
t(wdh)+1 ELSE gesamt(wdh+1)=gesamt(wdh+1)+
1
2380 LOCATE 36+((satz-1)*5),2+wdh/2+0.5:PR [5106]
INT pl(1);CHR$(8);":";USING"#";pl(2):NEXT:
NEXT
2390 '                                   [117]
2400 '--Wer scheidet aus?--              [1008]
2410 '                                   [117]
2420 FOR wa=1 TO finale*2 STEP 2:IF gesamt [10440]
(wa+1)=3 THEN spieler$(wa/2+0.5)=spieler$(
wa+1):paar(wa/2+0.5)=paar(wa+1) ELSE IF ge
samt(wa)=3 THEN spieler$(wa/2+0.5)=spieler
$(wa):paar(wa/2+0.5)=paar(wa)
2430 NEXT wa:GOSUB 2920:finale=finale/2:IF [6248]
 finale<1 THEN 2480 ELSE IF finale<8 THEN
LOCATE 20,15:PRINT" << SPACE to go on >> "
2440 WHILE INKEY$<>" ":WEND:CLS:GOTO 1940  [2512]
2450 '                                   [117]
2460 '--Runde aus--                      [1805]
2470 '                                   [117]
2480 FOR w=1 TO 32:stpunkt(w)=punkt(w):NEX [1877]
T:hi=0
2490 FOR dimvari=1 TO 32:FOR w=1 TO 32:IF  [5737]
stpunkt(w)>hi THEN hi=stpunkt(w):zae=w
2500 NEXT:posi(zae)=dimvari:stpunkt(zae)=0 [3069]
:hi=0:NEXT
2510 FOR w=1 TO 32:punkt(w)=33-posi(w):NEX [3211]
T
2520 CLS#1:PRINT#1:PRINT#1,"  P R A E M I  [12215]
E N - V E R G A B E":PRINT#1,"  ";STRING$(
33,"-"):PRINT#1,"   Jeder bekommt 1/3 dess
en,was       seine Sch^tzlinge einspielen.
"
2530 FOR w=1 TO 1500:NEXT:CLS#1:FOR w=1 TO [10397]
 6:LOCATE#1,1,w:PRINT#1,sp$(w):LOCATE#1,14
,w:PRINT#1," :    + ";USING"###,### $";pra
emie(w):NEXT:RESTORE 3050:GOSUB 3040
2540 FOR w=1 TO 6:ge(w)=ge(w)+praemie(w):G [7599]
OSUB 1780:FOR wa=1 TO 500:NEXT wa,w:PRINT#
1,CHR$(24):PRINT#3,CHR$(24):CLS#3:GOTO 184
0
2550 END                                 [110]
2560 '                                   [117]
2570 '--Kauf des Computers--             [1349]
2580 '                                   [117]
```

```
2590 IF mitspieler>5 THEN RETURN                [1740]
2600 IF RND>0.75 THEN IF RND>0.3 THEN 2730 [3580]
 ELSE RETURN
2610 rn=INT(RND*(6-mitspieler)+1+mitspiele [2017]
r)
2620 FOR wiederholung=1 TO 32:preis(we(wie [8828]
derholung))=(35-wiederholung)*3000+((40-al
ter(we(wiederholung)))*300)+INT(RND*20-5)*
1000:NEXT
2630 ideal=INT(RND*32+1)                        [1019]
2640 IF hab$(rn,3)<>"" AND hab$(rn,3)<>SPA [2156]
CE$(14) THEN RETURN
2650 IF verk(we(ideal))<>0 THEN RETURN      [2608]
2660 IF preis(we(ideal))>ge(rn) AND ge(rn) [1968]
>0 THEN RETURN
2670 FOR qw=1 TO 3:IF hab$(rn,qw)="" OR ha [5093]
b$(rn,qw)=SPACE$(14) THEN zaehl(rn)=qw:GOT
O 2690
2680 NEXT                                       [350]
2690 ge(rn)=ge(rn)-preis(we(ideal)):GOSUB [13101]
1780:hab$(rn,zaehl(rn))=player$(we(ideal))
:verk(we(ideal))=1:GOSUB 1490:preis1(rn,za
ehl(rn))=preis(we(ideal))-5000-(INT(RND*50
)*100):gehalt(rn,zaehl(rn))=INT(preis(we(i
deal))/8)
2700 posi1(rn,zaehl(rn))=posi(we(ideal)):R [2900]
ETURN
2710 '                                          [117]
2720 '--Verkauf des Computers--                 [873]
2730 '                                          [117]
2740 rn1=INT(RND*(6-mitspieler)+1+mitspiel [3445]
er)
2750 FOR id=3 TO 1 STEP -1:IF hab$(rn1,id) [4560]
="" OR hab$(rn1,id)=SPACE$(14) THEN NEXT:G
OTO 2600
2760 ge(rn1)=ge(rn1)+preis1(rn1,id):GOSUB  [3620]
1780:gehalt(rn1,id)=0:preis1(rn,id)=0
2770 FOR q=1 TO 32:IF player$(we(q))=hab$( [5492]
rn1,id) THEN verk(we(q))=0 ELSE NEXT
2780 hab$(rn1,id)=SPACE$(14):GOSUB 1490:po [2079]
si1(rn1,id)=0:RETURN
2790 '                                          [117]
2800 '--Informationen ausdrucken--              [1035]
2810 '                                          [117]
2820 CLS:PRINT"NAME:              ! ALTER:    [3804]
! SPIELWEISE:  ! BODENTYP"
2830 adj$(1)="Grundlinie":adj$(2)="Mitteld [6355]
ing":adj$(3)="Netz":bod$(1)="Sandplatz":bo
d$(2)="Kunstbelag":bod$(3)=" Rasen"
2840 PRINT"----------------!---------!--- [3600]
----------!-----------"
2850 FOR w=1 TO 16:LOCATE 1,2+w:PRINT play [7031]
er$(we(w)):LOCATE 18,2+w:PRINT"! ";alter(w
e(w));" J.":LOCATE 28,2+w:PRINT"!  ";adj$(
weise(we(w))):LOCATE 43,2+w:PRINT"!  ";bod
$(boden(we(w))):NEXT
2860 WHILE INKEY$="":WEND:LOCATE 1,1:PRINT [9617]
 CHR$(7):CLS:PRINT"NAME:              ! ALTE
R:  ! SPIELWEISE:  ! BODENTYP:":PRINT"----
------------!---------!-------------!---
---------"
2870 FOR w=17 TO 32:LOCATE 1,w-14:PRINT pl [10877]
ayer$(we(w)):LOCATE 18,w-14:PRINT"! ";alte
r(we(w));" J.":LOCATE 28,w-14:PRINT"!  ";a
dj$(weise(we(w))):LOCATE 43,w-14:PRINT"!
";bod$(boden(we(w))):NEXT
2880 WHILE INKEY$="":WEND:LOCATE 1,1:PRINT [4690]
 CHR$(7):CLS:GOTO 990
2890 '                                          [117]
2900 '--Weltranglistenpunkte vergeben--         [1774]
2910 '                                          [117]
2920 FOR w=1 TO finale:FOR w1=1 TO 32           [2437]
2930 IF spieler$(w)=player$(we(w1)) THEN p [4383]
unkt(we(w1))=punkt(we(w1))+1.2+RND
2940 NEXT w1,w                                  [376]
2950 '                                          [117]
2960 '--Praemien vergeben--                     [2408]
2970 '                                          [117]
2980 FOR w=1 TO finale:FOR a=1 TO 6:FOR b= [3117]
1 TO 3
2990 IF spieler$(w)=hab$(a,b) THEN praemie [4315]
(a)=praemie(a)+preisgeld(finale)*10000
3000 NEXT b,a,w:RETURN                          [1554]
3010 '                                          [117]
3020 '--Praemiensound--                         [1957]
3030 '                                          [117]
3040 READ t,d:SOUND 1,t*1.3,d/1.5,15,1:IF  [4373]
t=1 THEN RETURN ELSE 3040
3050 DATA 119,20,127,20,119,40,159,40,159, [7812]
40,159,20,179,20,190,20,159,20,119,40,119,
40,106,20,113,20,106,40,159,40,159,40,106,
20,119,20,127,20,106,20,80,40,80,40
3060 DATA 71,20,63,20,60,20,63,20,71,20,80 [8362]
,20,71,20,80,20,89,20,95,20,89,20,95,20,10
6,20,119,20,106,20,119,20,127,20,142,20,15
9,20,142,20,127,20,119,20,106,20,95,20,89,
20,106,20,95,40,119,40,119,80,1,1
3070 '                                          [117]
3080 '--Titelsound--                            [708]
3090 '                                          [117]
3100 READ t,d:SOUND 1,t,d/1.3,15,1:IF t=1  [2386]
THEN RETURN ELSE 3100
3110 DATA 319,40,159,40,159,20,179,20,190, [7591]
20,179,20,159,40,190,40,190,40,179,40,213,
40,213,40,159,40,190,40,190,40,159,40,159,
20,179,20,190,20,179,20,159,40,190,40,190,
40,179,40,213,40,213,40,239,80
3120 DATA 319,20,159,20,179,20,159,20,190, [10074]
20,159,20,213,20,159,20,190,20,159,20,213,
20,159,20,239,20,159,20,213,20,159,20,239,
20,159,20,253,20,159,20,239,20,159,20,213,
20,159,20,190,20,159,20
3130 DATA 179,20,159,20,190,20,159,20,213, [5895]
20,159,20,190,20,159,20,213,20,159,20,239,
20,159,20,213,20,159,20,239,20,159,20,253,
20,159,20,239,80
3140 DATA 319,40,159,40,159,20,179,20,190, [7824]
20,179,20,159,40,190,40,190,40,179,40,213,
40,213,40,159,40,190,40,190,40,159,40,159,
20,179,20,190,20,179,20,159,40,190,40,190,
40,179,40,213,40,213,40,239,80,1,1
3150 '                                          [117]
3160 '--Programm beenden--                      [1375]
3170 '                                          [117]
3180 CLS#1:CLS:LOCATE #1,10,1:PRINT#1," E  [2868]
- M e n u e ":PRINT#1
3190 PRINT#1,"1 => Programm verlassen":PRI [14534]
NT#1,"2 => Spielstand abspeichern":PRINT#1
,"3 => Alten Spielstand laden":PRINT#1,"4
=> Programm doch weiterf`hren"
3200 ein$="":WHILE ein$="":ein$=INKEY$:WEN [2253]
D
3210 IF ein$<"1" OR ein$>"4" THEN 3200 ELS [3050]
E ein=VAL(ein$):ON ein GOTO 3220,3260,3330
,3400
3220 GOSUB 4020:MODE 1:BORDER 3:INK 0,6:IN [3373]
K 1,16:END
3230 '                                          [117]
3240 '--Spielstand speichern--                  [1900]
3250 '                                          [117]
3260 CLS#1:LOCATE #1,5,2:PRINT#1,"REC & PL [5273]
AY druecken...":LOCATE #1,9,4:PRINT#1,"Dat
einame ? ";:LINE INPUT#1,d$:d$="!"+d$:OPEN
OUT d$
3270 FOR w=1 TO 32:PRINT#9,player$(w):PRIN [8100]
T#9,alter(w):PRINT#9,posi(w):PRINT#9,weise
(w):PRINT#9,boden(w):PRINT#9,verk(w):PRINT
#9,we(w):NEXT
3280 FOR w=1 TO 6:PRINT#9,ge(w):PRINT#9,sp [2199]
$(w):NEXT
3290 PRINT#9,mitspieler:FOR w=1 TO 6:FOR s [8757]
=1 TO 3:PRINT#9,hab$(w,s):PRINT#9,gehalt(w
,s):PRINT#9,preis1(w,s):PRINT#9,posi1(w,s)
:NEXT s,w:CLOSEOUT:GOTO 3180
3300 '                                          [117]
3310 '--Spielstand laden--                      [1993]
3320 '                                          [117]
3330 CLS#1:LOCATE #1,8,2:PRINT#1,"PLAY dru [8694]
ecken...":LOCATE #1,9,4:PRINT#1,"Dateiname
 ? ";:LINE INPUT#1,d$:d$="!"+d$:OPENIN d$
```

```
3340 FOR w=1 TO 32:INPUT#9,player$(w):INPU [8948]
T#9,alter(w):INPUT#9,posi(w):INPUT#9,weise
(w):INPUT#9,boden(w):INPUT#9,verk(w):INPUT
#9,we(w):NEXT
3350 FOR w=1 TO 6:INPUT#9,ge(w):INPUT#9,sp [2667]
$(w):NEXT
3360 INPUT#9,mitspieler:FOR w=1 TO 6:FOR s [12954]
=1 TO 3:INPUT#9,hab$(w,s):INPUT#9,gehalt(w
,s):INPUT#9,preis1(w,s):INPUT#9,posi1(w,s)
:NEXT s,w:CLOSEIN:CLS#3:GOSUB 1530:GOSUB 1
490:GOSUB 1780:GOSUB 1580:GOTO 3180
3370 '                                  [117]
3380 '--Doch Weiter--                   [1335]
3390 '                                  [117]
3400 CLS#1:LOCATE#1,8,1:PRINT#1," Zu welch [2331]
em Menue ? "
3410 PRINT#1:PRINT#1,"1 => W-Menue":PRINT# [7523]
1,"2 => K-Menue":PRINT#1,"3 => G-Menue":PR
INT#1,"4 => Z-Menue"
3420 ein$="":WHILE ein$="":ein$=INKEY$:WEN [2253]
D
3430 IF ein$<"1" OR ein$>"4" THEN 3420 ELS [2959]
E ein=VAL(ein$):ON ein GOTO 600,990,1840,3
540
3440 '                                  [117]
3450 '--Neuer Monat--                   [804]
3460 '                                  [117]
3470 FOR w9=1 TO 6:monat(w9)=monat(w9)+1:N [2785]
EXT
3480 FOR w9=mitspieler+1 TO 6:FOR s=1 TO 3 [6531]
:ge(w9)=ge(w9)-gehalt(w9,s):NEXT:GOSUB 178
0:monat(w9)=0:NEXT:RETURN
3490 '                                  [117]
3500 ' ***************                  [345]
3510 ' **  Z-Menue  **                  [946]
3520 ' ***************                  [345]
3530 '                                  [117]
3540 CLS#1:CLS:GOSUB 1780:LOCATE#1,10,1:PR [3642]
INT#1," Z - M e n u e ":PRINT#1
3550 PRINT#1,"1 => Geh[lter sehen":PRINT#1 [7949]
,"2 => Geh[lter zahlen":PRINT#1,"3 => Geh[
lter k^rzen"
3560 ein$="":WHILE ein$="":ein$=INKEY$:WEN [3799]
D:GOSUB 1660
3570 IF ein$<"1" OR ein$>"3" THEN 3560 ELS [4240]
E ein=VAL(ein$):ON ein GOTO 3610,3670,3750
3580 '                                  [117]
3590 '--Geh[lter sehen--                [2289]
3600 '                                  [117]
3610 CLS#1:GOSUB 1070:FOR d=1 TO 3:LOCATE  [8850]
3,3+d*3:PRINT hab$(cur,d):LOCATE 23,3+d*3:
PRINT USING"##,###.## $";gehalt(cur,d):LOC
ATE 35,3+d*3:PRINT"Gehalt.":NEXT
3620 LOCATE 3,14:PRINT"  Sie liegen";monat [6153]
(cur);"Monatsgeh[lter im R^ckstand.  "
3630 z=(gehalt(cur,1)+gehalt(cur,2)+gehalt [11294]
(cur,3))*monat(cur):LOCATE 3,16:PRINT" Das
 erg[be insgesamt ";:PRINT USING"###,###.#
#";z;:PRINT" $ zu zahlen. "
3640 LOCATE 3,18:PRINT" << Any key to go o [1857]
n >> ":CALL &BB06:CLS:GOTO 3540
3650 '                                  [117]
3660 '--Geh[lter zahlen--               [1547]
3670 '                                  [117]
3680 CLS#1:GOSUB 1070:GOSUB 3940:IF merk=1 [4008]
 THEN PRINT CHR$(7):GOTO 3540
3690 ge(cur)=ge(cur)-((gehalt(cur,1)+gehal [4078]
t(cur,2)+gehalt(cur,3))*monat(cur)):GOSUB
1780
3700 CLS:LOCATE 3,3:PRINT" ";sp$(cur);",Si [6668]
e haben Ihre Geh[lter ordentlich gezahlt.
"
3710 FOR w=1 TO 1500:NEXT:monat(cur)=0:GOT [2591]
O 3540
3720 '                                  [117]
3730 '--Geh[lter k^rzen--               [2179]
3740 '                                  [117]
3750 CLS#1:GOSUB 1070                   [793]
3760 CLS#1:LOCATE#1,6,1:PRINT#1," Wessen G [3840]
ehalt k^rzen ? ":PRINT#1
3770 FOR w=1 TO 3:PRINT#1,w;"=> ";hab$(cur [3986]
,w):NEXT:PRINT#1," 4 => Keinem,zum Z-Menue
"
3780 ein$="":WHILE ein$="":ein$=INKEY$:WEN [2253]
D
3790 IF ein$<"1" OR ein$>"4" THEN 3780 ELS [2213]
E ein=VAL(ein$)
3800 IF ein=4 THEN 3540 ELSE IF hab$(cur,e [3717]
in)="" OR hab$(cur,ein)=SPACE$(14) THEN 37
60
3810 LOCATE 4,3:PRINT" ";hab$(cur,ein);" b [5818]
ekommt z.Z. ";:PRINT USING"##,###.##";geha
lt(cur,ein);:PRINT" $ Gehalt. "
3820 LOCATE 1,6:INPUT"Um wieviel Dollar m] [7111]
chten Sie sein Gehalt k^rzen ";money
3830 random=money/10000:IF RND<random THEN [2205]
 3870
3840 IF RND>0.8 THEN 3870               [606]
3850 gehalt(cur,ein)=gehalt(cur,ein)-money [16571]
:GOSUB 3920:LOCATE 1,8:PRINT hab$(cur,ein)
;" akzeptiert die Gehaltsk^rzung und bekom
mt nun";:PRINT"ein Monatsgehalt von";:PRIN
T USING" ##,###.## $.";gehalt(cur,ein)
3860 PRINT:PRINT" << Any key to go on >> " [4436]
:CALL &BB06:CLS:GOTO 3760
3870 a$(1)="das h[tte er nicht verdient.": [10476]
a$(2)="Sie w[ren ein gieriger Geizhals.":a
$(3)="das w[re eine Frechheit."
3880 LOCATE 1,8:PRINT hab$(cur,ein);" akze [7826]
ptiert Ihre Gehaltsk^rzung nicht.":PRINT"E
r meint,";a$(INT(RND*3+1))
3890 PRINT:PRINT" Fristlose K^ndigung !! " [3240]
3900 FOR q=1 TO 32:IF player$(we(q))=hab$( [6413]
cur,ein) THEN verk(we(q))=0 ELSE NEXT
3910 hab$(cur,ein)=SPACE$(14):gehalt(cur,e [9615]
in)=0:posi1(cur,ein)=0::ge(cur)=ge(cur)+pr
eis1(cur,ein):preis1(cur,ein)=0:GOSUB 1780
:GOSUB 1490:CALL &BB06:GOTO 3540
3920 IF gehalt(cur,ein)<100 THEN gehalt(cu [1721]
r,ein)=100
3930 RETURN                             [555]
3940 FOR w=1 TO 3:IF hab$(cur,w)=SPACE$(14 [5204]
) OR hab$(cur,w)="" THEN NEXT:GOTO 3950 EL
SE merk=0:RETURN
3950 LOCATE 3,3:PRINT"Sie haben keinen ein [4033]
zigen Spieler."
3960 LOCATE 3,5:PRINT"Wem wollen Sie denn  [4912]
da etwas zahlen ?"
3970 LOCATE 3,7:PRINT"Der Putzfrau viellei [4670]
cht ?"
3980 FOR wa=1 TO 5000:NEXT:merk=1:RETURN   [1533]
3990 '                                  [117]
4000 '--Endabrechnung--                 [1071]
4010 '                                  [117]
4020 FOR w=1 TO 6:ge(w)=ge(w)-((gehalt(w,1 [8085]
)+gehalt(w,2)+gehalt(w,3))*monat(w)):ge(w)
=ge(w)+preis1(w,1)+preis1(w,2)+preis1(w,3)
:NEXT
4030 CLS:LOCATE 20,3:PRINT" Endabrechnung  [3009]
: "
4040 PRINT:PRINT"Alle Geh[lter wurden geza [10977]
hlt und die Spieler anschliessend":PRINT"i
n Freundschaft entlassen."
4050 WINDOW #4,1,13,16,21:PAPER#4,1:CLS#4: [7056]
FOR w=1 TO 6:LOCATE 1,8+w:PRINT" ";sp$(w);
" ";:LOCATE 15,8+w:PRINT USING"#,###,###.#
# $";ge(w):NEXT
4060 FOR w=1 TO 6:IF ge(w)>hi THEN hi=ge(w [2371]
):zae=w
4070 NEXT                               [350]
4080 LOCATE 3,17:PRINT" Der Sieger heisst  [7269]
damit... ";sp$(zae);" !!! ":RESTORE 4100
4090 READ t,d:IF t=-1 THEN 4110 ELSE SOUND [3997]
 1,t,d,13,1:SOUND 2,t+1,d,13,1:GOTO 4090
4100 DATA 478,15,379,15,319,15,358,15,379, [5926]
15,426,15,379,15,426,15,478,15,426,15,478,
15,506,15,478,30,379,30,319,30,239,90,-1,-
1
4110 WHILE UPPER$(INKEY$)<>"E":LOCATE 33,1 [6318]
1:PRINT"Press 'E' to End !":LOCATE 33,11:P
RINT SPC(18):WEND:RETURN
```

# GUAZUELA




**GUAZUELA ist ein Strategiespiel für eine Person. Sie spielen die Rolle des Präsidenten eines fiktiven mittelamerikanischen Staates. Sie müssen versuchen, möglichst lange zu regieren und dabei soviel Geld wie möglich auf Ihr Nummernkonto in die Schweiz zu schaffen.**

## Bildschirmaufteilung

Oben links wird das gerade zur Verfügung stehende Geld angezeigt. Darunter sieht man den Namen, unter dem der Spieler in GUAZUELA als Präsident agiert und daneben die Level-Nr. (0: schwach, 5: stark). Oben rechts wird angegeben, in welchem Vierteljahr man spielt. Darunter wird angegeben, in welcher Spielphase bzw. in welchem Menuepunkt des Hauptmenues man sich befindet.

## Ablauf

Das Spiel ist in Vierteljahre gegliedert. Das erste Vierteljahr des Spiels ist Frühling 1965. Ein Vierteljahr läuft folgendermaßen ab:

### Tagesfrage

Hier wird von Ihnen eine Ja/Nein-Entscheidung über eine gerade aktuelle politische Frage verlangt. Unter Umständen verursacht eine positive Entscheidung Kosten, die dann von dem Ihnen zur Verfügung stehenden Geld abgezogen werden. Negative Entscheidungen sind immer kostenlos.

### Attentate/Rebellionen/Wahlen/Krieg

Falls Sie sich den Zorn einer Bevölkerungsgruppe zugezogen haben, besteht die Gefahr von Attentaten und Rebellionen. Bei Attentaten brauchen Sie Glück, um dem Anschlag auszuweichen. Eine Rebellion überstehen Sie nur, wenn Sie andere Bevölkerungsgruppen hinter sich haben.

Zu bestimmten Terminen werden Wahlen durchgeführt, die Ihnen zeigen, ob Sie im Volk beliebt sind.

Sollten Sie Ihren Nachbarstaat Conduyana allzu schlecht behandelt haben (oder er aus irgendeinem anderen Grund kriegslüstern geworden sein), wird er Sie unter Umständen angreifen.

## Hauptmenue

Hier können Sie die Rahmenbedingungen Ihrer Politik festlegen und wirtschaften, um den Inhalt der Staatskasse zu vermehren.

Zu Punkt <0>:
Sie können die Leistungsfähigkeit Ihrer Wirtschaft dadurch beeinflussen, daß Sie sie subventionieren. Das mag volkswirtschaftlich gesehen ein Strohfeuer sein, aber Ihnen kommt es ja hauptsächlich darauf an, die Beteiligten zufriedenzustellen und Ihr Konto in der Schweiz zu füllen.

Zu Punkt <1>:
Sie können Bündnisse schließen und wieder kündigen. Wenn Sie ein Bündnis kündigen, können Sie mit derselben Großmacht erst nach einer ganzen Weile (bis zu zweieinhalb Jahren) wieder ein Bündnis schließen. Wenn Sie mit keiner Großmacht ein Bündnis schließen können (z.B. weil Sie ausländische Firmen verstaatlicht haben), wird dieser Punkt im Menue nicht mehr aufgeführt.

Zu Punkt <2>:
Achten Sie stets darauf, für einen Konflikt mit Conduyana wohlgerüstet zu sein. In diesem Menuepunkt können Sie Ihrer Armee Waffen kaufen und damit deren Ausrüstungsstand verbessern. Gelegentlich können Sie sich gegen Conduyana auch präventiv verteidigen.

Ein Krieg besteht aus mehreren Schlachten. Je nach Kampfstärke beider Nationen bewegt sich der Brennpunkt des Krieges mehr und mehr auf die Hauptstadt Ihres Staates oder auf Conduyanas Hauptstadt zu. Falls Sie meinen, Sie werden den Krieg sicher verlieren, empfiehlt es sich, mit <Q> die Aufgabe zu signalisieren. Beachten Sie bitte, daß das Programm die laufende Schlacht noch beendet.

Zu Punkt <3>:
Sie können zwischen 5 Sorten Wertpapieren wählen. Das Programm zeigt den Börsenstand graphisch und durch Angabe des $-Preises an. Die Zeile darunter gibt an, wieviel

Aktien von jeder Sorte Sie besitzen. Der Kurs hängt von der Jahreszeit und von Ihrer Wirtschaftspolitik ab. Er bleibt während eines Vierteljahres stabil.

Zu Punkt <4>:
Hier können Sie Geld in die Schweiz transferieren und sich einen Fluchtweg organisieren. Je länger Sie spielen, desto teurer wird es, die geeigneten Leute zu bestechen, um sich den Fluchtweg zu öffnen. Und denken Sie daran: die Schweizer Botschaft braucht einige Zeit, um Ihnen das Visum auszustellen!

Zu Punkt <5>:
Oberhalb der gestrichelten Linie sehen Sie, wie beliebt Sie sind, unterhalb, wie leistungsfähig Ihre Armee und Ihre Wirtschaft ist. Es ist nicht einfach, es allen recht zu machen!

Zu Punkt <6>:
Wenn Sie Spielstände abspeichern wollen, legen Sie eine dafür bestimmte Kassette/Diskette ein und wählen Sie diesen Punkt an.

Zu Punkt <8>:
Wählen Sie diesen Punkt, wenn Sie meinen, daß Sie für dieses Vierteljahr genug getan haben. Sie können auch <ENTER> statt <8> drücken.

**Ereignisse**
Gelegentlich passieren Dinge, auf die Sie keinen Einfluß haben, die aber Ihr Volk und Sie (und die Staatskasse) sehr wohl beeinflussen können.

**Auswertung**
Das Spiel endet ohne Punktwertung, wenn Sie erschossen werden. Sollten Sie es schaffen zu flüchten (oder wurden Sie dazu gezwungen), wird ein Score berechnet, der sich aus der Regierungsdauer und dem in die Schweiz geschafften Geld ergibt.

Außerdem werden Ihnen die Highscores angezeigt, die bisher erreicht wurden. Wenn Sie nicht mehr spielen wollen, können Sie die sechs High-Scores abspeichern, wenn Sie möchten. Bei Kassettenbetrieb verwenden Sie bitte eine Extra-Kassette.

**Die Listings**
Geben Sie die Listings 1–4 bitte der Reihe nach ein, und speichern Sie sie auf Disk oder Kassette. Listing 1 und 2 erstellen nach dem Start die Tagesfragen, d.h. sie müssen beide vor dem ersten Spiel einmal gestartet werden. Danach müßten sich auf Ihrem Datenträger die Dateien Guafrag. 0 und Guafrag. 1 befinden. Nun können Sie das Spiel mit RUN"GUAZUELA starten.

(Thomas Naumann/SH)

**für 464-664-6128**

```
10 'Stellt Guafrag.0 her                          [2086]
20 MODE 2:PRINT"Bitte Datentraeger bereith [6101]
alten - GUAFRAG.0 wird hergestellt."
30 OPENOUT"guafrag.0"                             [1058]
40 READ a$:IF a$<>"@" THEN PRINT#9,a$:FOR  [2361]
i=1 TO 10:READ b:PRINT#9,b:NEXT:GOTO 40
50 CLOSEOUT:PRINT"Fertig"                         [1621]
60 DATA"Sollen die Gewerkschaften verboten [4724]
 wer- den?"
70 DATA  1, 1,-1,-5, 5, 1,-1, 0,-2, 0             [1368]
80 DATA"Sollen Gewerkschaften zugelassen w [3748]
erden?"
90 DATA -1,-1, 0, 5,-4,-1,-1, 0,-2, 2             [1321]
100 DATA"Sollen die Gro~grundbesitzer ente [6016]
ignet  werden?"
110 DATA -6,-2, 5, 0,-4,-3, 2,-4, 0, 1            [1092]
120 DATA"Soll eine Flurbereinigung durchge [3245]
f|hrt  werden?"
130 DATA -4, 0,-2, 0, 0, 0, 0, 2, 0, 3            [1441]
140 DATA"Soll eine allgemeine Wehrpflicht  [4243]
einge- f|hrt werden?"
150 DATA -2, 8,-6,-8,-3,-2,-6,-1,-3, 1            [1657]
160 DATA"Soll die Armee Pre~kommandos eins [8815]
etzen  d|rfen, die junge M|nner zum Wehrdi
enst '|berreden' d|rfen?"
170 DATA -4, 10,-8,-10,-4,-2,-6,-3,-5, 0          [1159]
180 DATA"Soll die Kirche vom Staat getrenn [8134]
t wer- den und das Recht verlieren, den Ze
hnteneinzuziehen?"
190 DATA -2, 0,-2, 2, 0, 0, 1, 0, 0, 0            [1165]
200 DATA"Soll ein Importstop gegen ausl|nd [6451]
ische  Agrarprodukte verh|ngt werden?"
210 DATA  8, 0, 3,-4,-4, 0,-6, 4, 0, 1            [1291]
220 DATA"Sollen Z|lle auf ausl|ndische Ind [8058]
ustrie-produkte erhoben werden?"
230 DATA -3, 0,-3, 0, 4, 0,-6, 0, 4, 1            [869]
240 DATA"Soll die 40-Stunden-Woche eingef| [3231]
hrt    werden?"
250 DATA -8,-8, 6, 6,-8, 0, 2,-2,-2, 2            [1348]
260 DATA"Soll die 50-Stunden-Woche wiedere [4356]
inge-  f|hrt werden?"
270 DATA  5, 5,-6,-6, 5, 0,-2, 2, 2, 0            [1228]
280 DATA"Soll die Kinderarbeit verboten we [3136]
rden?"
290 DATA -6, 2, 6, 6,-3, 0, 1, 0,-1, 1            [773]
300 DATA"Sollen Sozialwohnungen eingericht [3927]
et wer-den?"
310 DATA  2, 0, 2, 6, 4, 0, 2, 1, 3, 10           [925]
320 DATA"Soll eine allgemeine Krankenversi [5529]
cherungeingef|hrt werden?"
330 DATA  0, 2, 6, 6, 2, 0, 2, 0, 0, 10           [1253]
340 DATA"Sollen die Nahrungsmittelpreise g [5886]
esenkt werden?"
350 DATA -8, 1, 0, 8, 2, 0, 0,-4, 4, 0            [1267]
360 DATA"Sollen die Nahrungsmittelpreise e [6654]
rh|ht  werden?"
370 DATA  6, 0, 0,-8,-2,-1, 0, 4,-3, 0            [877]
380 DATA"Soll General Conte del Oregano al [7648]
s Ver- teidigungsminister ernannt werden?"
390 DATA  2, 10, 0,-4,-4,-8, 0, 0, 0, 0           [1128]
400 DATA"Soll der Notstand ausgerufen werd [2350]
en?"
410 DATA -2, 7,-2,-6,-3,-7,-4,-2,-2, 1            [1453]
420 DATA"Sollen die Kompetenzen des Geheim [5379]
```

```
dien-  stes beschnitten werden?"
430 DATA  2, 6, 4, 6, 2,-10, 4, 1, 1, 0    [985]
440 DATA"Soll die Armee atomar bewaffnet w [2486]
erden?"
450 DATA  0, 8, 0, 0, 2,-2,-12, 0, 4, 10   [1289]
460 DATA"Sollen die Befugnisse des Geheimd [6685]
iensteserweitert werden?"
470 DATA -2,-8,-2,-2,-2, 10,-2,-1,-1, 0    [1518]
480 DATA"Soll es verboten werden, die Arme [13504]
e zur  Aufrechterhaltung von Ruhe und Ordn
ung  innerhalb der Landesgrenzen einzusetz
en?"
490 DATA -4,-8, 2, 2,-6, 6, 2, 0, 0, 4     [1425]
500 DATA"Sollen Z|lle abgeschafft werden?" [3477]
510 DATA -4, 0, 2, 2,-2, 0, 5, 1, 1, 1     [1087]
520 DATA"Sollen landwirtschaftliche Fl|che [6810]
n in  Industriegebiete umgewandelt werden
?"
530 DATA -6, 2,-6, 6, 6, 0,-2,-4, 4, 3     [821]
540 DATA"Sollen den Unternehmern Subventio [7386]
nen ge-w|hrt werden?"
550 DATA -2,-2, 0, 0, 8, 0, 0, 0, 2, 15    [974]
560 DATA"Soll der Arbeiterf|hrer Juan Cele [8804]
stes  als Volksverr|ter hingerichtet werd
en?"
570 DATA  4, 4,-4,-10, 7, 6,-6, 0,-2, 0    [1020]
580 DATA"Soll eine staatliche Lotterie ein [5856]
gef|hrtwerden?"
590 DATA  2, 2, 5, 5, 2, 2, 0, 0, 0, 1     [758]
600 DATA"Soll die allgemeine Schulpflicht  [5071]
einge- f|hrt werden?"
610 DATA -2, 0, 2, 4,-4,-4, 0,-2,-2, 8     [1390]
620 DATA"Soll das Lied 'Land der Himmel, L [5053]
and derBerge' als Nationalhymne eingef|hrt
 wer-den?"
630 DATA  2, 2, 2, 2, 2, 2,-2, 0, 0, 0     [989]
640 DATA"Soll die Fu~ballnationalmannschaf [15687]
t unse-res Landes von Ihnen den Orden 'Pou
r le merite' bekommen, weil sie ins Halbfi
na-le der Fu~ball-WM vorgesto~en ist?"
650 DATA  0,-2, 4, 8, 0, 0, 0, 0, 0, 0     [913]
660 DATA"Sollen die Ureinwohner unseres La [9965]
ndes,  die Xualtemec-Indianer, in Reservat
e zu-sammengehalten werden?"
670 DATA  2, 2,-2,-2, 2, 6, 0, 0, 0, 4     [1192]
680 DATA"Soll ein Propagandafeldzug f|r Eu [6835]
er Ex- zellenz durchgef|hrt werden?"
690 DATA  2, 2, 5, 5, 2, 2, 2, 0, 0, 5     [897]
700 DATA"Soll eine Kampagne 'Mehr Flei~' g [4582]
estar- tet werden?"
710 DATA  4, 2, 0, 0, 4, 0,-2, 4, 4, 5     [743]
720 DATA"Soll Ihre Frau das Waisenhaus in  [3857]
Ciudad de Guazuela besuchen?"
730 DATA  0, 0, 4, 4, 0, 0, 0, 0, 0, 0     [1039]
740 DATA"Sollen die Priester der Kirche vo [8323]
n der  Kanzel die Gottgef|lligkeit der Reg
ie-  rung Eurer Exzellenz verk|nden?"
750 DATA  0, 0, 6, 4, 0, 0,-2, 0, 0, 6     [1279]
760 DATA"Soll der Geheimdienst zur Rauschg [7843]
iftbe- k|mpfung eingesetzt werden?"
770 DATA -3,-1,-2, 1, 1, 4, 0,-1, 1, 2.5   [1054]
780 DATA"Sollen mehr Spione in Conduyana e [5441]
inge-  setzt werden?"
790 DATA  1, 2, 0, 0, 1, 5,-5, 0, 0, 6     [1092]
800 DATA"Sollen f|hrende M|nner der Politi [9072]
k und  Wirtschaft durch den Geheimdienst |
ber- pr|ft werden?"
810 DATA -1,-4, 1, 0,-4, 5,-3, 0, 0, 2     [1178]
820 DATA"Soll jedem h|heren Armeeoffizier  [8911]
ein Po-litischer Kommissar beigeordnet wer
den?"
ige  Nachtleben in unserer Hauptstadt zu u
```

```
830 DATA  0,-5, 0, 0, 0, 5, 0, 0, 0, 4     [894]
840 DATA"Sollen die Spielschulden von Caes [7673]
ar de  Buonviega, dem Chef des Geheimdiens
tes, |bernommen werden?"
850 DATA -1,-1,-3,-3,-1, 6, 0, 0, 0, 10    [975]
860 DATA"Die Armee w|nscht, da~ der Marine [10858]
st|tz- punkt Incarcion an der Nordk|ste un
seresLandes ausgebaut wird."
870 DATA  0, 4, 0,-1, 0, 0,-3, 0, 0, 10    [1015]
880 DATA"Sollen neue Boden-Luft- und Luft- [7236]
Luft-  Raketensysteme angeschafft werden?"
890 DATA -1, 8,-1,-1, 0, 0,-6, 0, 0, 10    [1557]
900 DATA"Soll der Sold erh|ht werden?"     [2114]
910 DATA -1, 6,-1,-1,-1,-1, 0, 0, 0, 5     [1073]
920 DATA"Soll eine Kampagne 'The Army Want [5694]
s You!'gestartet werden?"
930 DATA -3, 6,-1,-1,-3, 0,-2,-1,-1, 3     [1072]
940 DATA"In Tonuela haben randalierende So [12074]
ldaten Zivilisten zusammengeschlagen und F
rauenvergewaltigt. Soll ein Exempel statui
ertwerden?"
950 DATA  1,-6, 3, 3, 1, 1, 0, 0, 0, 0     [1340]
960 DATA"Sollen Insektengifte verboten wer [3506]
den?"
970 DATA -4, 0,-2, 0,-1, 0, 0,-1, 0, 0     [1546]
980 DATA"Soll den Gro~grundbesitzern die n [9677]
iedere Gerichtsbarkeit genommen werden?"
990 DATA -5, 0, 5, 1, 0, 0, 0,-1, 0, 0     [1306]
1000 DATA"Viele Landarbeiter z|chten als N [12615]
ebener- werb Mohn, um daraus Haschisch und
 Mari-huana zu gewinnen. Soll das in Zukun
ft  untersagt werden?"
1010 DATA -1, 2,-6, 0, 2, 1,-1,-2, 1, 0    [1728]
1020 DATA"Soll das Farbfernsehen eingef|hr [3010]
t wer-  den?"
1030 DATA  1, 0, 2, 5, 2, 1, 0,-1,-1, 1.5  [1504]
1040 DATA @                                [215]
```

```
10 'Stellt Guafrag.1 her                   [1918]
20 MODE 2:PRINT"Bitte Datentraeger bereith [5837]
alten - GUAFRAG.1 wird hergestellt."
30 OPENOUT"guafrag.1"                      [1064]
40 READ a$:IF a$<>"@" THEN PRINT#9,a$:FOR  [2361]
i=1 TO 10:READ b:PRINT#9,b:NEXT:GOTO 40
50 CLOSEOUT:PRINT"Fertig"                  [1621]
60 DATA"Soll f|r alle eine Altersrente ein [9040]
ge-  f|hrt werden, die aus der Staatskass
e  finanziert wird?"
70 DATA  1, 1, 4, 4, 1, 1, 1, 0, 0, 20     [1311]
80 DATA"Wollen Sie die ber|hmte Schauspiel [5550]
erin  Carmencita Villiaga heiraten?"
90 DATA  0, 0, 4, 5, 0,-1, 0, 0, 0, 0      [949]
100 DATA"Soll in der Romero-Halle die Popg [3510]
ruppe  'Big Swindle' auftreten?"
110 DATA -1,-1, 1, 4, 0, 0, 0, 0, 0, 4     [1460]
120 DATA"Soll die Kfz-Steuer abgeschafft w [2765]
erden?"
130 DATA  1, 0, 3, 4, 1, 0, 0, 0, 0, 12    [902]
140 DATA"Soll die Todesstrafe abgeschafft  [3461]
werden?"
150 DATA -3,-4, 1, 1,-2,-3, 0, 0, 0, 0     [1072]
160 DATA"Soll der Bitte der katholischen K [11485]
irche, strenger durchzugreifen, um das |pp
```

```
n- terbinden, stattgegeben werden?"
170 DATA  3,-2, 3,-1,-1, 3, 0, 0, 0, 3      [1253]
180 DATA"Soll mehr Geld gedruckt werden?"   [3373]
190 DATA -2, 1, 2, 2,-2, 0, 1,-1,-1, 1      [832]
200 DATA"Sollen die Kreditzinsen der Banca [6111]
 Nacio-nal gesenkt werden?"
210 DATA  3, 0, 1, 1, 4, 0,-1, 0, 2, 6      [899]
220 DATA"Sollen diplomatische Beziehungen  [7652]
zu Con-duyana aufgenommen werden?"
230 DATA  1,-1, 2, 0, 2,-3, 4, 1, 1, 1      [1010]
240 DATA"Sollen die diplomatischen Beziehu [6893]
ngen zuConduyana abgebrochen werden?"
250 DATA -1, 1, 0, 0,-2, 3,-4,-1,-1, 0      [1374]
260 DATA"Soll die Zensur verst{rkt werden? [1843]
"
270 DATA  3, 4,-3,-5, 4, 4,-2, 0,-1, 3      [1242]
280 DATA"Soll den bedingungslosen Rational [6898]
isie-  rungspl{nen der Unternehmer zugesti
mmt  werden?"
290 DATA  0, 0, 0,-5, 5, 0,-1, 0, 3, 0      [1352]
300 DATA"Soll unser Land dem Mittelamerika [3623]
-Pakt  beitreten?"
310 DATA  1, 3, 0, 0, 1, 1,-5, 0, 0, 4      [1336]
320 DATA"Wird Euer Exzellenz Seine K{nigli [6534]
che Ho-heit Prinz Jose von Conduyana empfa
ngen?"
330 DATA  0,-2, 0, 1, 1,-3, 5, 0, 0, 0      [1377]
340 DATA"Wollen Euer Exzellenz einen neuen [8203]
 Orden,den des Verdienten Volksgenossen, s
tif- ten?"
350 DATA  0, 0, 1, 1, 0, 0, 0, 0, 0, 1      [1260]
360 DATA"Soll dem Murren der Arbeiter Geh| [9684]
r ge-  schenkt werden, die sicherere Arbei
ts-  pl{tze fordern?"
370 DATA -1, 0, 1, 4,-4, 0, 0, 0,-1, 2      [1204]
380 DATA"Soll das Stra~ennetz unseres Land [6656]
es aus-gebaut werden?"
390 DATA  1, 3, 1, 1, 5, 0, 0, 1, 5, 12     [1177]
400 DATA"Sollen ausl{ndische Fluggesellsch [7640]
aften  Landerechte in unserem Land bekomme
n?"
410 DATA  0,-2, 0, 0, 3,-1, 1, 0, 2, 0      [878]
420 DATA"Wollen Euer Exzellenz der Premier [9446]
e von  'Guter Mond', dem neuesten St|ck de
s so-zialistischen Dichters Giorge Henricu
,  beiwohnen?"
430 DATA -3,-3, 2, 4,-3,-3, 1, 0, 0, 0      [1076]
440 DATA"Soll der Bitte der Gro~grundbesit [9557]
zer,  aufs{ssige Bauern h{rter zu bestraf
en,  stattgegeben werden?"
450 DATA  4, 1,-5,-1, 1, 2,-1, 1, 0, 0      [923]
460 DATA"Soll die amerikanische Gesellscha [9361]
ft Dry-Oil das Recht bekommen, die Erd|lvo
rr|- te des Landes ausbeuten zu d|rfen?"
470 DATA -2,-2,-1, 0,-2,-1, 1, 0, 0, 0      [1135]
480 DATA"Soll Frauen der Zugang zu |ffentl [4797]
ichen  [mtern gestattet werden?"
490 DATA -3,-4, 1, 4, 0, 0, 1, 0, 0, 0      [1371]
500 DATA"Soll Eurer Exzellenz ein neuer Pa [4666]
last  gebaut werden?"
510 DATA -1, 2, 0,-3,-1, 0, 0, 0, 0, 2      [1316]
520 DATA"Soll dem Geheimdienst verboten we [11547]
rden,  ohne richterlichen Beschlu~ das Tel
efon-netz anzuzapfen und Briefe zu |ffnen?
"
530 DATA  3, 0, 3, 5, 5,-6, 0, 0, 0, 0      [921]
540 DATA"Sollen an alle Kinder Luftballons [9895]
 mit  der Aufschrift 'Viva Excellenza' au
sge- geben werden?"
550 DATA  0, 0, 4, 3, 0, 3, 0, 0, 0, 3      [1014]
560 DATA"Soll die Kunstd|ngung erlaubt wer [4461]
den?"
570 DATA  4, 0, 2, 0, 0, 0, 0, 4, 0, 0      [811]
580 DATA"Soll den Gro~grundbesitzern verbo [11390]
ten  werden, ein Feld l{nger als zwei Ja
hre  hintereinander zu bestellen?"
590 DATA -3, 0,-2, 0, 0, 0, 0, 4, 0, 0      [743]
600 DATA"]berschwemmungen haben den Gro~gr [9910]
undbe- sitzern hohe Verluste bereitet. Sol
len  sie aus Staatsmitteln ausgeglichen we
r- den?"
610 DATA  6, 0, 2, 0, 0, 0, 0, 3, 0, 12     [1070]
620 DATA"Sollen alle Conduyaner ohne Auswe [6325]
is ver-haftet und zu Zwangsarbeit verurtei
lt  werden?"
630 DATA  4, 3, 0, 0, 4, 2,-10, 0, 0, 0     [931]
640 DATA"Soll eingef|hrt werden, da~ den A [7237]
rmen  des Landes eine warme Mahlzeit am T
ag  garantiert wird?"
650 DATA -3, 3, 3, 5,-3, 0, 0, 0, 0, 8      [829]
660 DATA"Soll unser Land eine neue Verfass [8591]
ung be-kommen, in der die Menschenrechte g
aran-tiert werden?"
670 DATA -2,-4, 5, 5,-4,-2, 2, 0, 0, 0      [1203]
680 DATA"Soll die Selbstverwaltung der D|r [13527]
fer ab-geschafft und statt der Dorf{lteste
n  Regionalkommissare eingesetzt werden?
"
690 DATA -3, 3,-5, 0, 0, 4, 0, 2, 0, 3      [484]
700 DATA"Soll es den Schulen zur Pflicht g [8767]
emacht werden, in jedem Klassenzimmer ein
Bild von Ihnen aufzuh{ngen?"
710 DATA  0, 3,-2,-2, 0, 3, 0, 0, 0, 1      [1532]
720 DATA"Geben Euer Exzellenz den Auftrag, [10589]
 ein  neues Verfahren zur Herstellung von
 Ke- rosin zu entwickeln?"
730 DATA  0, 2, 0, 2, 4, 0, 0, 0, 3, 9      [1499]
740 DATA"Sollen alle Raketen der Armee ver [8222]
schrot-tet und die Truppenzahl verringert
wer- den?"
750 DATA  0,-7, 0, 3, 0, 3, 8, 0, 0, 1      [966]
760 DATA"Soll unser Land eine Mannschaft z [7888]
u den  Olympischen Spielen schicken?"
770 DATA  0, 0, 3, 3, 0, 0,-2, 0, 0, 2      [961]
780 DATA"Soll den Gro~grundbesitzern das ' [8441]
Recht  der ersten Nacht' genommen werden?"
790 DATA -4, 0, 5, 1, 0, 0, 0, 0, 0, 0      [1318]
800 DATA"Soll der 4. M{rz - der Jahrestag  [12531]
der  Machtergreifung Eurer Exzellenz - z
um  allgemeinen Feiertag erkl{rt werden?"
810 DATA -3, 0, 4, 4,-4, 1,-1,-1,-1, 0      [1273]
820 DATA"Der Minister f|r Landwirtschaft u [17740]
nd For-sten, Oberst Coutinho, hat vor Geri
cht  einen Meineid geschworen, als er gefr
agtwurde, ob er Bestechungsgelder angenom-
 men hat. Soll er entlassen werden?"
830 DATA  3,-5, 2, 1, 1, 0, 1, 0, 0, 0      [734]
840 DATA"Wollen Euer Exzellenz eine allgem [5262]
eine  Pockenschutzimpfung einf|hren?"
850 DATA  2, 2, 2, 2, 2, 0, 0, 1, 1, 3      [807]
860 DATA"Sollen die traditionellen Ma~e ab [8200]
ge-  schafft und das metrische System ei
nge- f|hrt werden?"
870 DATA  0, 3,-2, 0, 3, 0, 0, 0, 2, 2      [867]
880 DATA"Soll dem Trinkwasser Fluor zur Ve [9792]
rhin-  derung der weitverbreiteten Zahnf{u
le  zugesetzt werden?"
890 DATA  2, 4, 0, 0, 1, 0, 0, 0, 0, 2      [1002]
900 DATA"Soll neben dem Staatsrundfunk ein [10145]
e pri- vate Radiostation den Betrieb aufne
hmen d|rfen?"
910 DATA  0,-1, 1, 2, 5,-3, 0, 0, 0, 0      [1135]
920 DATA"Soll die Alkoholsteuer abgeschaff [3771]
t wer- den?"
```

```
930 DATA -1,-3, 4, 4,-1,-1, 0,-1,-1, 2      [1540]
940 DATA"Soll der Guazuela-Peso aufgewerte  [3453]
t wer- den?"
950 DATA -2, 0, 0, 1,-4, 0, 3, 0,-1, 0      [1403]
960 DATA"Soll der Guazuela-Peso abgewertet  [3817]
 wer-  den?"
970 DATA  3, 0, 0, 0, 4, 0,-4, 1, 2, 2      [700]
980 DATA"Der B|rgermeister von Casella, ve  [16802]
rmut-  lich ein heimlicher Oppositioneller
     wurde vom Geheimdienst ermordet. Soll
   eine Untersuchungskommission eingesetzt
 werden?"
990 DATA  0, 0, 2, 4, 0,-6, 2, 0, 0, 1      [1555]
1000 DATA"Soll der |ffentliche Nahverkehr   [7413]
zum     Nulltarif angeboten werden?"
1010 DATA  0, 0, 3, 5,-4, 0, 0, 0,-1, 3     [1032]
1020 DATA"Sollen Ergebnisse der milit|risc   [10379]
hen For-schung der Industrie kostenlos zur
 Ver- f|gung gestellt werden?"
1030 DATA  0, 0, 0, 0, 5, 0, 0, 0, 4, 6     [1612]
1040 DATA @                                  [215]
```

```
10 'Guazuela v 2.2 vom 1. 3. 87              [1523]
20 'Poke-Plan: a=himem                       [800]
30 'a+1 bis a+4: inks 0 ... 3                [1897]
40 'a+5: level                               [484]
50 'a+6: highscores laden (ja=1, nein=0)     [3463]
60 'a+8 bis a+22: Name (15 Buchstaben)       [1398]
70 'a+23 bis a+54: Staedtekoordinaten st(i   [5707]
,j) - st(1,0) *LowB, HighB*, st(1,1) *LowB
, HighB*, st(2,0)...
80 'a+55 ... a+534: Grafikdaten              [1213]
90 MODE 1:FOR i=0 TO 3:INK i,0:NEXT:LOCATE   [4901]
 1,25:PRINT">":LOCATE 40,25:PRINT"<";
100 WINDOW#0,1,40,4,24:WINDOW#1,1,19,1,1:W   [10457]
INDOW#2,21,40,1,1:WINDOW#3,3,38,25,25:WIND
OW#4,21,40,2,2:WINDOW#5,1,19,2,2:WINDOW#6,
1,40,3,24:farbe0=0:farbe1=24:farbe2=20:far
be3=16
110 el$=CHR$(200):eg$=CHR$(200):e$=CHR$(20   [3213]
0):DIM schwierig$(5):name$="HURTADO"
120 DATA ]bung,Neuling,Provinzpolitiker,Er   [5073]
nsthafter Versuch,Profi,"Selbstm|rder"
130 RESTORE 120:FOR i=0 TO 5:READ schwieri   [2764]
g$(i):NEXT
140 PEN#1,2:PEN#2,3:PEN#3,0:PEN#4,1:PAPER#   [6103]
1,3:PAPER#2,2:PAPER#3,1:PAPER#4,3:PAPER#5,
2:PEN#5,0
150 GOSUB 230                                [935]
160 OPENOUT"d":MEMORY HIMEM-1:CLOSEOUT:a=H   [4217]
IMEM:MEMORY a-535:a=HIMEM
170 POKE a+5,1:POKE a+6,0:GOSUB 1320         [1888]
180 GOSUB 1200:GOSUB 1220                    [1274]
190 DATA 148,313,188,325,228,271,288,265,3   [3210]
54,251,382,301,396,329,456,289
200 RESTORE 190:FOR i=1 TO 8:FOR j=0 TO 1:   [6743]
READ st(i,j):POKE a+23+((i-1)*2+j)*2,st(i,
j)MOD 256:POKE a+23+((i-1)*2+j)*2+1,st(i,j
)\ 256:NEXT:NEXT
210 RUN"!gua"                                [436]
220 'Symbole                                 [369]
230 SYMBOL AFTER 0:SYMBOL 48,124,134,138,1   [3149]
46,162,194,124,0
240 SYMBOL 49,16,48,80,16,16,16,56,0         [2605]
250 SYMBOL 50,124,130,2,12,112,128,254,0     [1835]
260 SYMBOL 51,120,132,2,12,2,132,120,0       [2390]
270 SYMBOL 52,128,128,136,254,8,8,8,0        [2222]
280 SYMBOL 53,254,128,128,124,2,2,252,0      [2649]
290 SYMBOL 54,124,128,128,252,130,130,124,   [2477]
0
300 SYMBOL 55,254,2,4,8,16,16,16,0           [1646]
310 SYMBOL 56,124,130,130,124,130,130,124,   [2378]
0
320 SYMBOL 57,124,130,130,126,2,130,124,0    [2408]
330 SYMBOL 65,16,40,68,130,254,130,130,0     [2330]
340 SYMBOL 66,248,132,132,252,130,130,252,   [2359]
0
350 SYMBOL 67,126,128,128,128,128,128,126,   [1880]
0
360 SYMBOL 68,252,130,130,130,130,130,252,   [2392]
0
370 SYMBOL 69,254,128,128,240,128,128,254,   [2095]
0
380 SYMBOL 70,254,128,128,240,128,128,128,   [2567]
0
390 SYMBOL 71,126,128,128,142,130,130,124,   [2365]
0
400 SYMBOL 72,130,130,130,254,130,130,130,   [3106]
0
410 SYMBOL 73,16,16,16,16,16,16,16,0         [1645]
420 SYMBOL 74,16,16,16,16,16,16,96,0         [1854]
430 SYMBOL 75,136,144,160,192,160,144,136,   [2812]
0
440 SYMBOL 76,128,128,128,128,128,128,254,   [2329]
0
450 SYMBOL 77,130,198,170,146,130,130,130,   [2969]
0
460 SYMBOL 78,130,194,162,146,138,134,130,   [2354]
0
470 SYMBOL 79,124,130,130,130,130,130,124,   [1864]
0
480 SYMBOL 80,252,130,130,252,128,128,128,   [1697]
0
490 SYMBOL 81,124,130,130,130,130,146,124,   [2162]
4
500 SYMBOL 82,252,130,130,252,136,132,130,   [1711]
0
510 SYMBOL 83,126,128,128,124,2,2,252,0      [1802]
520 SYMBOL 84,254,16,16,16,16,16,16,0        [2520]
530 SYMBOL 85,130,130,130,130,130,130,124,   [2234]
0
540 SYMBOL 86,130,130,130,130,68,40,16,0     [2298]
550 SYMBOL 87,130,130,130,130,130,84,40,0    [2047]
560 SYMBOL 88,130,68,40,16,40,68,130,0       [2134]
570 SYMBOL 89,130,68,40,16,16,16,16,0        [2096]
580 SYMBOL 90,254,4,8,16,32,64,254,0         [1740]
590 SYMBOL 97,0,0,120,4,124,132,124,0        [1956]
600 SYMBOL 98,128,128,252,130,130,130,252,   [2240]
0
610 SYMBOL 99,0,0,126,128,128,128,126,0      [2114]
620 SYMBOL 100,2,2,126,130,130,130,126,0     [1933]
630 SYMBOL 101,0,0,124,130,254,128,126,0     [1864]
640 SYMBOL 102,24,32,32,48,32,32,32,0        [1722]
650 SYMBOL 103,0,0,124,130,130,126,2,252     [2160]
660 SYMBOL 104,128,128,252,130,130,130,130   [2098]
,0
670 SYMBOL 105,0,16,0,16,16,16,16,0          [1539]
680 SYMBOL 106,0,16,0,16,16,16,16,96         [1628]
690 SYMBOL 107,64,64,72,80,96,80,72,0        [1860]
700 SYMBOL 108,32,32,32,32,32,32,24,0        [1534]
710 SYMBOL 109,0,0,236,146,146,146,146,0     [2307]
720 SYMBOL 110,0,0,124,66,66,66,66,0         [1957]
730 SYMBOL 111,0,0,124,130,130,130,124,0     [1866]
740 SYMBOL 93,130,0,130,130,130,130,124,0    [2097]
750 SYMBOL 91,146,40,68,130,254,130,130,0    [1859]
760 SYMBOL 92,130,124,130,130,130,130,124,   [2281]
0
770 SYMBOL 123,0,132,120,4,124,132,124,0     [2700]
780 SYMBOL 124,0,130,124,130,130,130,124,0   [2393]
```

```
790 SYMBOL 112,0,0,252,130,130,252,128,128 [2242]
800 SYMBOL 113,0,0,126,130,130,126,2,2     [2147]
810 SYMBOL 114,0,0,88,96,64,64,64,0        [2464]
820 SYMBOL 115,0,0,126,128,124,2,252,0     [2447]
830 SYMBOL 116,32,32,112,32,32,32,24,0     [1710]
840 SYMBOL 117,0,0,66,66,66,66,62,0        [2384]
850 SYMBOL 118,0,0,0,130,68,40,16,0        [1916]
860 SYMBOL 119,0,0,130,130,146,170,68,0    [1990]
870 SYMBOL 120,0,0,68,40,16,40,68,0        [1834]
880 SYMBOL 121,0,0,130,68,40,16,32,64      [1799]
890 SYMBOL 122,0,0,124,8,16,32,124,0       [1936]
900 SYMBOL 58,0,0,0,16,0,16,0,0            [2170]
910 SYMBOL 44,0,0,0,0,0,8,16,32            [2044]
920 SYMBOL 46,0,0,0,0,0,0,16,0             [1822]
930 SYMBOL 63,120,4,4,56,32,0,32,0         [1727]
940 SYMBOL 33,16,16,16,16,16,0,16,0        [1523]
950 SYMBOL 125,0,0,66,0,66,66,62,0         [2008]
960 SYMBOL 126,60,66,66,92,66,66,92,64     [1593]
970 SYMBOL 238,129,195,102,24,24,36,66,129 [2409]
980 'Tasten umbelegen                      [859]
990 KEY DEF 17,1,125,93                    [746]
1000 KEY DEF 19,1,123,91                   [1140]
1010 KEY DEF 22,1,124,92                   [1264]
1020 KEY DEF 26,1,124,92,64                [765]
1030 KEY DEF 24,1,126,94,0                 [1480]
1040 RETURN                                [555]
1050 'Geldanzeige                          [1447]
1060 CLS#1:PRINT#1,USING fo$;ge!;:RETURN   [3570]
1070 'Warten auf Taste                     [733]
1080 WHILE INKEY$<>"":WEND:PRINT CHR$(7);: [4842]
MID$(eg$,1,1)=CHR$(200)
1090 MID$(eg$,1,1)=INKEY$:IF eg$=CHR$(200) [3260]
THEN 1090
1100 MID$(e$,1,1)=eg$:MID$(el$,1,1)=LOWER$ [3339]
(eg$):en=ASC(eg$)-48
1110 IF el$="j"OR el$="y"THEN ja=-1 ELSE j [1815]
a=0
1120 SOUND 1,0,2,5,,,31:PAPER#3,0:CLS#3:PA [3359]
PER#3,1:RETURN
1130 'Taste druecken                       [657]
1140 CLS#3:PRINT#3,TAB(7);"<TASTE> dr|cken [4198]
";:GOSUB 1080:RETURN
1150 'Zifferntaste druecken                [1996]
1160 CLS#3:PRINT#3,TAB(6);"Zifferntaste dr [5381]
|cken";:GOSUB 1080:RETURN
1170 'Ja/Nein-Abfrage                      [2122]
1180 CLS#3:PRINT#3,TAB(3);"<J> f|r Ja - <N [4795]
> f|r Nein";:GOSUB 1080:RETURN
1190 'Bitte warten                         [1984]
1200 CLS#3:PRINT#3,TAB(10);"BITTE WARTEN!" [2663]
;:RETURN
1210 'Grafikdaten einpoken                 [1161]
1220 RESTORE 1250:i=55                     [751]
1230 READ k:IF k=999 THEN 1240 ELSE POKE a [9855]
+i,k MOD 256:POKE a+i+1,k\256:READ k:POKE
a+i+2,k MOD 256:POKE a+i+3,k\256:i=i+4:GOT
O 1230
1240 RETURN                                [555]
1250 DATA 8,365,631,365,631,195,8,195,8,36 [10370]
5,52,343,62,347,76,345,78,341,76,333,72,32
9,74,325,84,327,86,323,98,333,104,349,130,
351,132,345,156,347,186,337,186,327,192,33
1,196,347,244,341,244,339,260,341,272,351,
286,337,300,337,310,325
1260 DATA 308,319,302,315,304,309,308,297, [10554]
306,285,304,273,308,259,310,249,298,245,28
4,255,266,275,252,269,256,243,244,219,244,
229,230,223,216,227,220,235,222,237,218,24
1,202,241,202,237,192,235,158,229,136,247
1270 DATA 114,241,98,261,100,283,108,287,1 [4374]
02,297,84,301,68,311,54,313,50,325,52,343
1280 DATA 312,323,314,335,312,343,324,347, [12723]
346,351,392,335,406,349,440,351,466,343,46
8,337,466,331,456,319,456,319,458,305,472,
305,474,301,486,287,500,289,512,307,518,31
5,510,333,502,339,502,351,508,359,522,361,
532,365
1290 DATA 552,365,554,353,556,335,544,323, [8776]
546,305,554,293,566,285,566,273,554,267,53
0,271,530,253,514,251,484,235,458,251,438,
253,420,261,406,277,392,273,380,257,374,23
9,362,225,344,225
1300 DATA 344,227,348,231,356,245,342,249, [2546]
326,239,300,233,288,253,999
1310 'Einstellungen                         [1831]
1320 FOR i=0 TO 5:CLS#i:NEXT:PRINT#2,"VORE [3106]
INSTELLUNGEN"
1330 LOCATE 10,1:PRINT CHR$(24);" G U A Z  [3105]
U E L A ";CHR$(24);
1340 LOCATE 10,4:PEN 2:PRINT"von Thomas Na [4930]
umann":LOCATE 10,6:PRINT"M|rz 1987 - v 2.2
":PEN 1
1350 PRINT:PRINT                           [743]
1360 LOCATE 1,9:PRINT"<0> Anleitung":PRINT [9144]
:PRINT"<1> Monitor: ";:PEN 2:IF farbe2=9 T
HEN PRINT"Farbmonitor" ELSE PRINT"Gr|nmoni
tor"
1370 PEN 1:PRINT:PRINT"<2> Level: ";:PEN 2 [4505]
:PRINT schwierig$(PEEK(a+5));CHR$(18)
1380 PEN 1:PRINT:PRINT"<3> High-Scores lad [6422]
en: ";:PEN 2:IF PEEK(a+6)=1 THEN PRINT"ja
 " ELSE PRINT"nein"
1390 PEN 1:PRINT:PRINT"<4> Name Eurer Exze [4134]
llenz: ";:PEN 2:PRINT name$
1400 PEN 1:PRINT:PRINT"<5> oder <ENTER>: S [9638]
tart":INK 0,farbe0:INK 1,farbe1:INK 2,farb
e2:INK 3,farbe3:BORDER farbe0:PEN 1:PAPER
0
1410 GOSUB 1160:IF el$=CHR$(13) OR en=5 TH [2863]
EN 5140
1420 IF en<0 OR en>4 THEN 1410             [1388]
1430 IF en<>0 THEN 5000                    [1517]
1440 CLS:PEN 2:PRINT"           ANLEITUNG" [2345]
:PEN 1
1450 PRINT                                 [361]
1460 PRINT"Sie sind Pr|sident von Guazuela [3739]
,  einem"
1470 PRINT"kleinen, ziemlich instabilen  S [4691]
taat  in"
1480 PRINT"Mittelamerika. Ihr Ziel  ist  e [4443]
s,  m|g-"
1490 PRINT"lichst lange zu regieren und da [2915]
bei  Ihr"
1500 PRINT"Nummernkonto in der Schweiz zu  [4830]
 f|llen."
1510 PRINT"Dabei m|ssen Sie alle Bev|lkeru [6260]
ngsgrup-"
1520 PRINT"pen ruhig halten.  Aber  das  i [3239]
st  noch"
1530 PRINT"nicht alles: Ihr kleiner Staat  [4001]
hat noch"
1540 PRINT"einen  argw|hnischen  Nachbar [3739]
n,  die"
1550 PRINT"Operettenmonarchie Conduyana."  [2697]
1560 PRINT                                 [361]
1570 PRINT"Das Spiel gliedert  sich  in  Q [3717]
uartale."
1580 PRINT"Jedes Quartal l|uft folgenderma [3205]
~en ab:"
1590 PRINT                                 [361]
1600 PEN 2:PRINT"1) Tagesfragen":PEN 1     [1909]
1610 PRINT                                 [361]
1620 PRINT"F|llen Sie Ihre Entscheidung  ] [4337]
ber  die"
1630 PRINT"gerade aktuelle  und  hei~  dis [5335]
kutierte"
```

```
0 6:hisc$(i)=SPACE$(15):NEXT:tt$=SPACE$(15
):GOSUB 9300:v$=STRING$(40,"-"):ba$=STRING
$(20,CHR$(233)):inv$=CHR$(24)
20 DIM gp$(7),sn$(8),mn$(4),ka$(5),pa$(5), [2981]
mch$(1),menue$(8),st(8,2),pk(120,1)
30 fo$="###,###,###.## $":xn$=CHR$(23)+CHR [6362]
$(1):xf$=CHR$(23)+CHR$(0):mch$(0)="USA":mc
h$(1)="UdSSR":ee$="Euer Exzellenz"
40 FOR i=1 TO 8:FOR j=1 TO 2:st(i,j)=PEEK( [14210]
him!+23+((i-1)*2+j-1)*2)+PEEK(him!+23+((i-
1)*2+j-1)*2+1)*256:NEXT:NEXT:RESTORE 100:F
OR i=1 TO 8:READ sn$(i):NEXT:RESTORE 110:F
OR i=1 TO 4:READ mn$(i):NEXT:RESTORE 120:F
OR i=1 TO 5:READ ka$(i):READ pa$(i):NEXT
50 RESTORE 130:FOR i=1 TO 7:READ gp$(i):NE [13405]
XT:RESTORE 140:FOR i=0 TO 8:READ menue$(i)
:NEXT:FOR i=0 TO 119:pk(i,0)=PEEK(him!+55+
i*4)+PEEK(him!+55+i*4+1)*256:pk(i,1)=PEEK(
him!+55+i*4+2)+PEEK(him!+55+i*4+3)*256:NEX
T
60 CLS#6:IF PEEK(him!+6)=1 THEN GOSUB 7800 [11052]
:GOSUB 9600:OPENIN"guahigh":FOR i=0 TO 5:L
INE INPUT#9,hisc$(i):INPUT#9,hisc(i):NEXT:
CLOSEIN:GOSUB 3800
70 DIM g!(7),p!(7),ma(7),ku!(5),ak(5),kn(4 [2450]
),si!(5),bu(1)
80 sk!=1-0.11*PEEK(him!+5):FOR i=1 TO 7:g! [11770]
(i)=60+INT(RND*90):p!(i)=30+INT(RND*50):NE
XT:kn(1)=1:kn(2)=2:kn(3)=4:kn(4)=55:bu(0)=
0:bu(1)=0:fe$=">>> Leider nicht m|glich. <
<<"
90 ge!=10000:f=INT(3/sk!):fw=0:un=0:zz=0:z [10311]
=0:sz!=6*sk!:vs=0:ed=0:rn=0:at=INT(3*sk!):
rv=INT(5*sk!):bs=0:fd=3:bv=5000+INT(RND*30
00*sk!):c=5000+INT(RND*3000):mo=4:jr=1964
100 DATA Ciudad de Guazuela,Incarcion,Case [4947]
lla,Tonuela,Anchasar,Diego,Porto Verde,Con
ducuor
110 DATA Fr|hling,Sommer,Herbst,Winter [3072]
120 DATA F. Gonzales,Christdemokr.,J. Feli [6176]
pe,Sozialisten,C. Turandot,Kommunisten,Gen
eral Todo,Nationalisten,Dr. Limosa,Liberal
e
130 DATA Gro~grundb.,Armee,Landarbeiter,Fa [3344]
brikarbeiter,Unternehmer,Geheimdienst,Cond
uyana
140 DATA Wirtschaft,Au~enpolitik,Milit|rpo [10169]
litik,"B|rse",Schweiz,Lage,Speichern,High-
Scores,N|chstes Quartal
290 'Vorspann [606]
300 FOR i=8 TO 22:MID$(hisc$(6),i-7,1)=CHR [5658]
$(PEEK(him!+i)):NEXT:GOSUB 4700:CLS:PRINT"
Altes Spiel laden?":GOSUB 7600:IF ja THEN
GOSUB 4900
490 'Hauptprogramm [1580]
500 RANDOMIZE TIME:CLS:GOSUB 7300:GOSUB 70 [3622]
00:IF bs>0 THEN kf!=0.5/sk!ELSE kf!=1
510 IF mo=2 THEN kf!=kf!+0.05*sk!ELSE IF m [3721]
o=4 THEN kf!=kf!-0.05*sk!
520 ku!(1)=p!(7)*0.5*(RND*0.2*sk!+0.9):ku! [12219]
(2)=p!(7)*0.4*(RND*0.2*sk!+0.9):ku!(3)=p!(
5)*0.6*(RND*0.2*sk!+0.9):ku!(4)=p!(1)*0.7*
(RND*0.2*sk!+0.9):x!=0:FOR i=1 TO 6:x!=x!+
g!(i):NEXT:x!=x!/60:ku!(5)=x!*5*(RND*0.2*s
k!+0.9)
530 FOR i=1 TO 5:ku!(i)=(ku!(i)+60+RND*10) [1700]
*kf!:NEXT
690 'Kontrollen [661]
700 IF zz<=fm THEN 900 ELSE IF df<mf THEN [2352]
df=df+1:GOSUB 9300:zz=0:GOTO 900
710 PRINT"Sie wurden von Ihrem Premier weg [6601]
en Al-  tersschw|che abgesetzt.":PRINT:GOT
O 6000
720 IF ge!>50000000 THEN PRINT ee$;" Premi [13114]
erminister beschlag-nahmt die Staatskasse
und findet Sie miteinem Teil daraus ab.":P
RINT:ge!=ge!*RND*0.5+ge!*0.2:GOSUB 7000:GO
TO 6000
730 IF (bv>9000 AND RND<0.2/sk!)OR RND<0.0 [10220]
1 OR bv>30000 THEN op=INT(RND*(bv/2)):PRIN
T"Die Pest fordert ";:PRINT USING"#,###,##
# ";op*1000;:PRINT"Todesopfer.":bv=bv-op:G
OSUB 7200
740 IF (c>9000 AND RND<0.2*sk!)OR RND<0.01 [11508]
 OR c>30000 THEN op=INT(RND*(c/2)):PRINT"I
n ";gp$(7);" fallen ";PRINT USING"#,###,##
# ";op;:PRINT"Menschen ":PRINT"der Cholera
 zum Opfer.":c=c-op:GOSUB 7200
890 'Tagesfragen [926]
900 CLS#2:PRINT#2,"TAGESFRAGEN":t=ta(zz):P [7948]
RINT fr$(t):PRINT:PRINT ee$;" m|ge entsche
iden!":PRINT
910 IF fp!(t,10)>0 THEN x!=(fp!(t,10)*(z+1 [15552]
5*sk!)/(15*sk!))*(0.8+RND*0.2)*1000:PRINT"
Kosten f|r eine positive Entscheidung:":PR
INT:PEN 2:PRINT USING"   --- ###,###.## $
---";x!:PEN 1 ELSE PRINT"Es entstehen kein
e Kosten.":x!=0
920 GOSUB 7600:PRINT:IF ja AND ge!<x!THEN [7458]
PRINT fe$:GOTO 930 ELSE IF ja THEN PRINT"O
.k.":ge!=ge!-x!:GOSUB 7000:GOTO 940 ELSE P
RINT"Nun gut!"
930 FOR i=1 TO 9:fp!(t,i)=-0.5*fp!(t,i):NE [3006]
XT
940 FOR i=1 TO 7:g!(i)=g!(i)+fp!(t,i)*10:N [5087]
EXT:p!(1)=p!(1)+fp!(t,8)*10:p!(5)=p!(5)+fp
!(t,9)*10:GOSUB 9400:GOSUB 7900:CLS
1090 'Wahlen [188]
1100 IF z MOD 15<>8 THEN 1400 ELSE CLS#2:P [14839]
RINT#2,"WAHLEN":PRINT"Der Tag der Wahrheit
 ist heranger|ckt,":PRINT ee$;"! Werden Si
e die f|lli-";"gen Wahlen abhalten?":GOSUB
 7600
1110 IF NOT ja THEN g!(1)=g!(1)+40:g!(2)=g [5019]
!(2)+50:g!(3)=g!(3)-70:g!(4)=g!(4)-90:g!(5
)=g!(5):g!(6)=g!(6)+30:g!(7)=g!(7)-20:GOSU
B 9400:GOSUB 7900:GOTO 1400
1120 LOCATE 1,5:PRINT"Es kandidieren f|r d [6734]
ie Pr|sidentschaft  unseres Landes:"
1130 FOR i=1 TO 5:LOCATE 1,7+i:PRINT ka$(i [6310]
);TAB(20);pa$(i):NEXT:PRINT:PRINT"und ";ee
$;"!":GOSUB 7200
1140 CLS:PEN 2:PRINT"Kandidat";TAB(15);"Pa [4435]
rtei";TAB(30);"Stimmant.":PRINT:PEN 1
1150 FOR i=1 TO 5:si!(i)=200-g!(i):NEXT:x! [5095]
=0:FOR i=1 TO 5:x!=x!+g!(i):NEXT:x!=x!/5:s
i!(0)=x!
1160 si!(1)=si!(1)*(1.5+RND*0.5):si!(2)=si [11124]
!(2)*(2+RND):si!(5)=si!(5)*(0.5+RND*0.25):
si!(0)=si!(0)*(2+RND)*sk!:x!=0:FOR i=0 TO
5:x!=x!+si!(i):NEXT:FOR i=0 TO 5:si!(i)=si
!(i)*100/x!:NEXT
1170 FOR i=1 TO 5:LOCATE 1,2+i:PRINT ka$(i [11369]
);TAB(15);pa$(i);TAB(30);:PRINT USING"##.#
";si!(i);:PRINT"%":NEXT:LOCATE 1,9:PRINT e
e$;TAB(30);:PRINT USING"##.#";si!(0);:PRIN
T"%":PRINT
1180 j=0:m=si!(0):FOR i=1 TO 5:IF si!(i)>m [3523]
 THEN j=i:m=si!(i)
1190 NEXT:IF j=0 THEN PRINT inv$;"Sie sind [10922]
 der Wahlsieger!";inv$;:PRINT:PRINT"Herzli
chen Gl|ckwunsch!":GOSUB 7200:GOTO 1400
1200 PRINT"Wahlsieger ist ";ka$(j);" von d [15338]
en":PRINT pa$(j):PRINT:PRINT"Wollen ";ee$;
" ihn als Volksfeind";"hinrichten lassen?"
:GOSUB 7600:IF NOT ja THEN PRINT:PRINT"Ihn
```

```
1640 PRINT"politische Frage. Je nachdem, w [5333]
ie  Ihre"
1645 GOSUB 1140:CLS                        [894]
1650 PRINT"Entscheidung ausf{llt, werden T [3395]
eile der"
1660 PRINT"Bev{lkerung  zustimmen,  ander [4748]
e  sich"
1670 PRINT"{rgern und  wieder  andere  ste [3684]
hen  ihr"
1680 PRINT"gleichg{ltig gegen{ber."        [2723]
1690 PRINT                                 [361]
1700 PEN 2:PRINT"2) Steuern":PEN 1         [2051]
1710 PRINT                                 [361]
1720 PRINT"Wenn  keine  Bev{lkerungsgrupp [5205]
e  eine"
1730 PRINT"Revolution anfacht oder  Conduy [3026]
ana  Sie"
1740 PRINT"angreift,  gibt  Ihnen  das  S [3582]
teueramt"
1750 PRINT"einen  Bericht.  Sie  k{nne [4142]
n  den"
1760 PRINT"Steuersatz {ndern, aber passen  [4120]
Sie auf,"
1770 PRINT"da~ Sie Ihr Volk nicht  so  aus [3958]
pressen,"
1780 PRINT"da~ es rebelliert!"             [2789]
1790 PRINT                                 [361]
1800 PEN 2:PRINT"3) Menue":PEN 1           [1687]
1810 PRINT                                 [361]
1820 PRINT"Sie  k{nnen  die  Wirtschaft [5330]
  durch"
1830 PRINT"Subventionen f{rdern,  sich  G [4712]
eld  von"
1840 PRINT"Schutzm{chten besorgen oder Ihr [4881]
e  Armee"
1850 PRINT"ausr{sten. Sie k{nnen au~erdem  [3968]
 an  der"
1855 GOSUB 1140:CLS                        [894]
1860 PRINT"B{rse Aktien kaufen und verkauf [4601]
en, Geld"
1870 PRINT"in die Schweiz transferieren  u [6549]
nd  Ihre"
1880 PRINT"Flucht dorthin vorbereiten. Mit [2459]
  'Lage'"
1890 PRINT"k{nnen Sie einen Lagebericht an [4493]
fordern,"
1900 PRINT"aber nur, wenn  Sie  es  sich   [3034]
mit  dem"
1910 PRINT"Geheimdienst  nicht  verdorben  [4293]
  haben."
1920 PRINT"Au~erdem k{nnen Sie sich die  a [2195]
ktuellen"
1930 PRINT"High-Scores ansehen und den  Sp [4075]
ielstand"
1940 PRINT"abspeichern."                   [2658]
1950 PRINT                                 [361]
1960 PEN 2:PRINT"4) Bildschirmaufteilung": [3403]
PEN 1
1970 PRINT                                 [361]
1980 PRINT"Am oberen Rande des  Bildschirm [4918]
s  sieht"
1990 PRINT"man in der ersten Zeile links o [5415]
ben  den"
2000 PRINT"aktuellen     Staatskassenstand [4143]
     und"
2010 PRINT"darunter Name und Level (0: sch [3369]
wach, 5:"
2020 PRINT"sehr   stark).   Rechts    oben [3770]
    wird"
2030 PRINT"angegeben,  in  welcher  Quart [5536]
alsphase"
2040 PRINT"bzw.  Menuepunkt  man  sich   b [4709]
efindet."
```

```
2050 PRINT"Darunter  steht  das  Quarta [2518]
1.  Das"
2055 GOSUB 1140:CLS                        [894]
2060 PRINT"Programm zeigt Ihnen in  der  u [5016]
ntersten"
2070 PRINT"Zeile an, welche Eingabe es  vo [2756]
n  Ihnen"
2080 PRINT"verlangt."                      [2003]
2090 PRINT                                 [361]
2100 PEN 2:PRINT"5) Besondere Bedienungshi [3856]
nweise":PEN 1
2110 PRINT                                 [361]
2120 PRINT"Statt <N> f]r 'Nein' k{nnen  Si [2945]
e  immer"
2130 PRINT"auch die  <ENTER>-Taste  dr{cke [5572]
n.  Wenn"
2140 PRINT"Sie  auf  der  B{rse  Aktien  v [3777]
erkaufen"
2150 PRINT"wollen,  m}ssen  Sie  auf  die [3948]
   Frage"
2160 PRINT"'Aktien  verkaufen:'  eine  Lee [2915]
reingabe"
2170 PRINT"machen (nur <ENTER>  dr{cken).  [4990]
 W{hrend"
2180 PRINT"eines Krieges k{nnen Sie  mit   [2950]
<Q>  wie"
2190 PRINT"'Quit'  aufgeben  und  Repa [4370]
rationen"
2200 PRINT"zahlen,  falls  gen{gend  Geld  [3595]
 in  der"
2210 PRINT"Staatskasse ist."               [2720]
2220 GOSUB 1140:CLS:GOTO 1330              [1519]
5000 IF en<>1 THEN 5030                    [1238]
5010 IF farbe2=9 THEN farbe2=20:farbe3=16  [3307]
ELSE farbe2=9:farbe3=6
5020 GOTO 1360                             [411]
5030 IF en<>2 THEN 5060                    [1251]
5040 IF PEEK(a+5)=5 THEN POKE a+5,0 ELSE P [3088]
OKE a+5,PEEK(a+5)+1
5050 GOTO 1360                             [411]
5060 IF en<>3 THEN 5090                    [1575]
5070 IF PEEK(a+6)=1 THEN POKE a+6,0 ELSE P [2104]
OKE a+6,1
5080 GOTO 1360                             [411]
5090 FOR i=9 TO 21:LOCATE 1,i:PRINT CHR$(1 [4745]
8);:NEXT:LOCATE 1,9
5100 PRINT"Den Namen Eurer Exzellenz bitte [5071]
:":PRINT:INPUT"",name$
5110 IF LEN(name$)>15 THEN PRINT:PRINT"Nic [7992]
ht mehr als 15 Buchstaben!":GOSUB 1140:GOT
O 5090
5120 name$=UPPER$(name$)                   [790]
5130 FOR i=9 TO 21:LOCATE 1,i:PRINT CHR$(1 [4243]
8);:NEXT:GOTO 1360
5140 POKE a+1,farbe0:POKE a+2,farbe1:POKE  [3806]
a+3,farbe2:POKE a+4,farbe3
5150 IF LEN(name$)<15 THEN name$=name$+SPA [3337]
CE$(15-LEN(name$))
5160 FOR i=8 TO 22:POKE a+i,ASC(MID$(name$ [3971]
,i-7,1)):NEXT:RETURN
```

```
10 DEFINT a-z:eg$=CHR$(200):e$=eg$:el$=eg$ [15380]
:df=0:him!=HIMEM:mf=1:fz=50:DIM fr$(fz),fp
!(fz,10),ta(fz),hisc$(6),hisc(6):FOR i=0 T
```

```
en bleibt nur die Flucht...":GOTO 6000
1210 PEN 0:PAPER 2:CLS:LOCATE 10,10:PRINT  [8147]
UPPER$(ka$(j)):LOCATE 10,12:PRINT"wurde hi
ngerichtet.":GOSUB 8300:PEN 1:PAPER 0
1220 g!(j)=g!(j)-60-30/sk!:FOR i=1 TO 5:g!  [5353]
(i)=g!(i)-30/sk!:NEXT:g!(6)=g!(6)+40:GOSUB
 9400:GOSUB 7900
1390 'Revolution?                           [867]
1400 FOR i=1 TO 7:IF g!(i)<30/sk!THEN ma(i  [3150]
)=ma(i)+1 ELSE IF g!(i)>50+50/sk!THEN ma(i
)=MAX(ma(i)-1,0)
1410 NEXT:IF rn>0 THEN 1700 ELSE GOSUB 840  [7150]
0:IF NOT(x!>at AND x!<rv AND RND<0.3/sk!)T
HEN 1430
1420 CLS#2:PRINT#2,"ATTENTAT!":CLS:PRINT"E  [14012]
in Attentat wurde auf Sie ver}bt.":PRINT:G
OSUB 8300:RANDOMIZE TIME:IF RND<0.05 THEN
PRINT"Es war erfolgreich.":PRINT:GOTO 6300
 ELSE PRINT"Gl}ck gehabt!":GOSUB 7200:GOTO
 1800
1430 IF NOT(x!>=rv AND RND<0.3/sk!)THEN 17  [5736]
00 ELSE CLS#2:PRINT#2,"REVOLUTION!":PAPER
2:PEN 0:CLS
1440 LOCATE 5,3:IF gr<>6 THEN PRINT"Rebell  [7205]
ion der ";gp$(gr)ELSE PRINT"Der ";gp$(6);"
 inszeniert":LOCATE 5,4:PRINT"eine Revolut
ion!"
1450 PRINT:ma(gr)=0:GOSUB 8300              [1368]
1460 x!=0:FOR k=1 TO 6:x!=x!+g!(k):NEXT:x!  [4295]
=x!-g!(7):x!=x!/5:h=x!-g!(gr)
1470 IF h>=100-RND*50*sk!THEN 1490 ELSE LO  [3815]
CATE 5,6:PRINT ee$;" sind abgesetzt!"
1480 IF NOT fw OR RND<0.1 THEN LOCATE 5,10  [7797]
:PRINT ee$;" wurden erschossen!":GOSUB 720
0:PEN 1:PAPER 0:CLS:GOTO 6300 ELSE GOSUB 7
200:PEN 1:PAPER 0:CLS:GOTO 6000
1490 LOCATE 5,6:PRINT ee$;" haben die":LOC  [7981]
ATE 5,7:PRINT"Revoluzzer niedergeworfen!":
g!(gr)=0:rn=5:PEN 1:PAPER 0:GOSUB 7200
1690 'Krieg?                                [651]
1700 IF c<4000 OR ma(7)<rv OR g!(7)>30/sk!  [5155]
OR RND>0.3/sk!OR p!(7)<50/sk!OR fd>=1 THEN
 1800
1710 CLS:op!=ge!*RND:PRINT gp$(7);" droht   [11738]
Ihnen mit Krieg!":PRINT:PRINT"M|chten Sie
ihn durch einen 'Kredit' ab-wenden (H|he:"
;ROUND(op!,2);"$)?"
1720 GOSUB 7600:IF NOT ja THEN GOSUB 8700   [4574]
ELSE ge!=ge!-op!:GOSUB 7000
1730 IF ed>0 THEN 6000                      [840]
1790 'Steuern                               [434]
1800 CLS#2:PRINT#2,"STEUERN":CLS:PRINT ee$  [14054]
;" Steueramt berichten:":PRINT:PRINT"Bev|l
kerung:";TAB(25);:PRINT USING LEFT$(fo$,11
);bv*1000:PRINT:PRINT"Zu versteuern:";TAB(
25);
1810 bp!=0:FOR i=1 TO 6:bp!=bp!+g!(i):NEXT  [14904]
:bp!=bp!+5*p!(1)+5*p!(5):bp!=bp!*100:bp!=b
p!*(0.9+RND*0.2)*bv/6000:PRINT USING fo$;b
p!:PRINT:PRINT"Steuersatz:";TAB(25);:PRINT
 USING LEFT$(fo$,13);sz!;:PRINT" %":PRINT:
PRINT"Einnahmen:";TAB(25);
1820 gw!=(bp!/100)*sz!:PRINT USING fo$;gw!  [8687]
:LOCATE 1,17:PRINT"Wollen ";ee$;" den Steu
ersatz {n-";"dern?":GOSUB 7600:IF NOT ja T
HEN 1850 ELSE as!=sz!
1830 LOCATE 1,19:PRINT CHR$(18);:INPUT"Neu  [6120]
er Steuersatz: ",sz!:IF sz!<0 OR sz!>99 TH
EN PRINT fe$:GOTO 1830
1840 FOR i=1 TO 5:g!(i)=g!(i)+as!*10-sz!*1  [7075]
2:NEXT:GOSUB 7900:IF sz!>20 AND sz!>as!THE
N rn=0:FOR i=1 TO 5:ma(i)=at+1:NEXT
1850 ge!=ge!+gw!:GOSUB 7000                 [725]
```

```
1990 'Menue                                 [1057]
2000 CLS#2:CLS:PRINT#2,"REGIERUNG":FOR i=0  [10137]
 TO 8:IF(bu(0)>0 AND bu(1)>0 AND i=1)OR(wh
>0 AND i=4)THEN 2010 ELSE PRINT USING"<#>
";i;:PRINT menue$(i):PRINT
2010 NEXT                                   [350]
2020 GOSUB 7500:IF en=8 OR el$=CHR$(13)THE  [6269]
N 5000 ELSE IF en=0 THEN GOSUB 3500:GOTO 2
000 ELSE IF en<1 OR en>7 THEN 2020
2030 ON en GOSUB 2200,2500,2700,3200,8100,  [3302]
4800,3800:IF ed THEN 6000
2040 GOTO 2000                              [363]
2190 'Aussenp.                              [809]
2200 CLS:CLS#2:PRINT#2,"AUSSENPOLITIK":IF   [10174]
bu(0)>0 AND bu(1)>0 THEN PRINT"B}ndnisse s
ind z. Z. nicht m}glich!":GOSUB 7200:GOTO
2330
2210 PRINT ee$;" haben ";:IF bu(0)<>-1 AND  [13839]
 bu(1)<>-1 THEN PRINT"kein B}ndnis";"abges
chlossen!"ELSE PRINT"ein B}ndnis mit":IF b
u(0)=-1 THEN PRINT"den USA abgeschlossen."
ELSE PRINT"der UdSSR abgeschlossen."
2220 PRINT:FOR i=0 TO 1:IF bu(i)=-1 THEN P  [9741]
RINT:PRINT"Wollen ";ee$;" es k}ndigen?":GO
SUB 7600:IF ja THEN bu(i)=5+INT(RND*5/sk!)
:ks=0:GOTO 2200
2230 NEXT:PRINT:IF bu(0)=-1 OR bu(1)=-1 TH  [13304]
EN 2270 ELSE PRINT"Wollen ";ee$;" ein B}nd
nis neu";"abschlie~en?":FOR i=0 TO 1:IF bu
(i)=0 THEN PRINT USING"<#>";i;:PRINT" mit
der ";mch$(i)
2240 NEXT:GOSUB 7500:FOR i=0 TO 1:IF bu(i)  [3016]
=0 AND en=i THEN bu(i)=-1
2250 NEXT:IF en<0 OR en>1 THEN GOTO 2000    [2218]
2260 ks=0                                   [288]
2270 IF ks>3 THEN 2330                      [1294]
2280 PRINT:FOR i=0 TO 1:IF bu(i)<>-1 THEN   [2969]
2320
2290 GOSUB 7700:IF q THEN 2320              [1533]
2300 IF i=0 THEN a=p!(5)ELSE a=p!(1)        [1268]
2310 IF a>100 THEN PRINT"Die ";mch$(i);" i  [16296]
st der Ansicht, da~ unser":PRINT"Land kein
e Unterst}tzung braucht.":GOSUB 7200 ELSE
us=INT(RND*10+3)*1000:ge!=ge!+us:GOSUB 700
0:PRINT:PRINT"Gew{hrt werden";us;"$":ks=ks
+1:GOSUB 7200
2320 NEXT                                   [350]
2330 RETURN                                 [555]
2490 'Mil.-pol.                             [729]
2500 CLS:CLS#2:PRINT#2,"MILIT[RPOLITIK":PR  [12602]
INT"Wollen Sie Waffen kaufen?":GOSUB 7600:
IF NOT ja THEN 2520 ELSE PRINT:INPUT"F}r w
ieviel $: ",x!:IF x!>ge!THEN PRINT:PRINT f
e$:GOSUB 7200:GOTO 2500
2510 k=2:GOSUB 8000                         [1161]
2520 IF fd>0 OR c<3000 THEN GOTO 2530 ELSE  [9608]
 PRINT:PRINT"Wollen Sie ";gp$(7);" }berfal
len?":GOSUB 7600:IF NOT ja THEN 2530 ELSE
GOSUB 8700
2530 RETURN                                 [555]
2690 'Boerse                                [334]
2700 CLS:CLS#2:PRINT#2,"B\RSE"              [1828]
2710 PEN 2:PRINT" Condu-";TAB(9);" CMC";TA  [11952]
B(17);" Holec";TAB(25);" Fruit";TAB(33);"S
taats-";:PRINT" copper";TAB(17);" Ltd.";TA
B(25);" Trade";TAB(33);"anleihen":PEN 1
2715 MOVE 10,200:DRAWR 0,125,1:MOVE 10,200  [4954]
:DRAWR 630,0:FOR i=50 TO 100 STEP 50:MOVE
6,200+i:DRAWR 8,0,2:NEXT
2720 FOR i=1 TO 5:a=i*128-100:b=INT(ku!(i)  [13078]
/2):MOVE a,200:DRAWR 0,b,1:DRAWR 60,0:DRAW
R 0,-b:DRAWR-60,0:FOR j=1 TO 29:MOVE a+j*2
,200:DRAWR 0,b,3:NEXT:MOVE a,b+200:DRAWR 3
```

```
0,30,2:DRAWR 60,0:DRAWR-30,-30:MOVE a+60,2
00:DRAWR 30,30:DRAWR 0,b
2730 LOCATE i*8-5,13:PRINT inv$;i;inv$:LOC [4458]
ATE i*8-7,15:PRINT USING"###.##$";ku!(i);:
NEXT
2740 LOCATE 1,17:PRINT CHR$(18):FOR i=1 TO [12147]
 5:LOCATE i*8-7,17:PEN 2:PRINT USING"###,#
##";ak(i):PEN 1:NEXT:LOCATE 1,19:PRINT CHR
$(18);"Welche Aktien kaufen?":GOSUB 7500:I
F en<1 OR en>5 THEN 2760
2750 LOCATE 1,19:PRINT CHR$(18);:LOCATE 1, [16491]
19:x!=MIN(32700-ak(en),INT(ge!/ku!(en))):P
RINT"Wieviele (max.";x!;"): ";:INPUT"",mx!
:IF mx!>x!THEN GOTO 2750 ELSE ge!=ge!-mx!*
ku!(en):GOSUB 7000:ak(en)=ak(en)+mx!:GOTO
2740
2760 LOCATE 1,19:PRINT CHR$(18);:LOCATE 1, [6741]
19:PRINT"Welche Aktien verkaufen?":GOSUB 7
500:IF en<1 OR en>5 THEN 2790
2770 LOCATE 1,19:PRINT CHR$(18);:LOCATE 1, [7783]
19:INPUT"Wieviele: ",x!:IF x!>ak(en)THEN G
OTO 2770 ELSE ak(en)=ak(en)-x!:ge!=ge!+x!*
ku!(en):GOSUB 7000
2780 GOTO 2740                             [391]
2790 RETURN                                [555]
3190 'Flucht                               [602]
3200 CLS:CLS#2:PRINT#2,"FLUCHTWEGE":PRINT" [11493]
Sie haben bisher";:PRINT USING fo$;su!:PRI
NT"in der Schweiz.":PRINT:PRINT"Wollen Sie
 Geld transferieren?":GOSUB 7600:PRINT
3210 IF NOT ja THEN 3230 ELSE INPUT"Betrag [6499]
: ",x!:IF x!>ge!THEN su!=su!+ge!:ge!=0 ELS
E ge!=ge!-x!:su!=su!+x!
3220 GOSUB 7000:PRINT                      [858]
3230 IF fw THEN 3260 ELSE f=3:PRINT"Wollen [8448]
 Sie einen Fluchtweg einrichten?":PRINT:ko
!=(z+10)/10*10000:PRINT"Kosten: ";:PRINT U
SING fo$;ko!
3240 GOSUB 7600:IF ja AND ge!>=ko!THEN ge! [4165]
=ge!-ko!:GOSUB 7000:fw=-1:PRINT"O.k.":GOTO
 3280
3250 IF ja AND ge!<ko!THEN PRINT fe$:GOSUB [2292]
 7200
3255 GOTO 3280                             [680]
3260 IF f>0 AND fw THEN PRINT ee$;" k|nnen [7965]
 noch nicht fl|ch-";"ten. Noch hat Ihnen d
ie Schweiz kein Vi-sum ausgestellt.":GOSUB
 7200:GOTO 3280
3270 PRINT"Wollen Sie fl|chten?":PRINT:GOS [3256]
UB 7600:IF ja THEN 6000
3280 RETURN                                [555]
3490 'Wirtsch.-pol.                        [671]
3500 CLS:CLS#2:PRINT#2,"WIRTSCHAFT":PRINT" [19276]
<1> Subventionen f|r die Agrarwirtschaft";
:PRINT:PRINT"<2> Subventionen f|r die Indu
strie":PRINT:IF NOT vs THEN PRINT"<3> Vers
taatlichung ausl{ndischer Firmen";:PRINT
3510 PRINT"<4> Schmuggel von High-Tech":PR [3930]
INT:PRINT"<5> Zur|ck"
3520 GOSUB 7500:IF el$=CHR$(13)THEN en=5 E [4251]
LSE IF en<1 OR en>5 THEN 3520
3530 IF en<>1 AND en<>2 THEN 3560 ELSE IF  [2994]
en=1 THEN k=1 ELSE k=5
3540 CLS:PRINT:INPUT"Wieviel: ",x!:IF x!>g [4026]
e!THEN PRINT:PRINT fe$:GOSUB 7200:GOTO 354
0
3550 GOSUB 8000:GOTO 3500                  [2452]
3560 IF en<>3 OR vs THEN 3600 ELSE ge!=ge! [4472]
+RND*30000*sk!+20000:GOSUB 7000:vs=-1
3570 IF bu(0)>=0 AND bu(1)>=0 THEN CLS:PRI [12366]
NT"Ausl{ndische M{chte distanzieren sich":
PRINT"von der Ma~nahme ";ee$;".":GOSUB 720
0:FOR i=0 TO 1:bu(i)=bu(i)+5+INT(RND*5/sk!
):NEXT:GOTO 3500
3580 CLS:FOR i=0 TO 1:IF bu(i)=-1 THEN PRI [8996]
NT"Die ";mch$(i);" sagen sich von unserem
";"Land los.":bu(i)=bu(i)+INT(RND*5/sk!)+5
3590 NEXT:GOSUB 7200:GOTO 3500             [975]
3600 IF en<>4 THEN 3620 ELSE CLS:PRINT:IF  [9419]
bu(0)<>-1 THEN PRINT"Leider haben ";ee$;"
Spione kei-";"nen Zugang zu High-Tech-Prod
ukten bekom-men.":GOSUB 7200:GOTO 3620
3610 PRINT ee$;" Spione haben Computer-";" [18109]
chips |ber die Botschaft in New York    ei
ngeschmuggelt.":PRINT:PRINT"Der Pr{sident
der USA ist dar|ber em-   p|rt.":bu(0)=5+I
NT(RND*5/sk!):p!(5)=p!(5)+INT(RND*80)+20:G
OSUB 7200
3620 RETURN                                [555]
3790 'High-Scores                          [1009]
3800 CLS:CLS#2:PRINT#2,"HIGH-SCORES":GOSUB [3412]
 9700:GOSUB 7200:RETURN
4690 'Name und Level                       [504]
4700 CLS#5:PRINT#5,hisc$(6);PEEK(him!+5);: [2918]
RETURN
4790 'Speichern                            [428]
4800 CLS#2:CLS:PRINT#2,"SPEICHERN":PRINT"W [5827]
irklich?":GOSUB 7600:IF NOT ja THEN 4840 E
LSE GOSUB 7800
4810 OPENOUT"!guasave":PRINT#9,df:PRINT#9, [13360]
zz:PRINT#9,z:PRINT#9,ge!:PRINT#9,su!:FOR i
=1 TO 7:PRINT#9,g!(i):PRINT#9,p!(i):NEXT:F
OR i=0 TO 1:PRINT#9,bu(i):NEXT:PRINT#9,mo:
PRINT#9,jr:PRINT#9,sz!
4820 PRINT#9,c:PRINT#9,bv:PRINT#9,f:PRINT# [7639]
9,fw:PRINT#9,rn:PRINT#9,fd:PRINT#9,bs:PRIN
T#9,vs:PRINT#9,sk!:FOR i=0 TO 5:PRINT#9,ku
!(i):PRINT#9,ak(i):NEXT
4830 PRINT#9,fm:FOR i=0 TO fm:PRINT#9,ta(i [3652]
):NEXT:CLOSEOUT
4840 RETURN                                [555]
4890 'Laden                                [384]
4900 CLS:GOSUB 9600:GOSUB 7800             [1931]
4910 OPENIN"!guasave":INPUT#9,dff:INPUT#9, [13522]
zz:zz=zz+1:INPUT#9,z:z=z+1:INPUT#9,ge!:INP
UT#9,su!:FOR i=1 TO 7:INPUT#9,g!(i):INPUT#
9,p!(i):NEXT:FOR i=0 TO 1:INPUT#9,bu(i):NE
XT:INPUT#9,mo:INPUT#9,jr:INPUT#9,sz!
4920 INPUT#9,c:INPUT#9,bv:INPUT#9,f:INPUT# [7780]
9,fw:INPUT#9,rn:INPUT#9,fd:INPUT#9,bs:INPU
T#9,vs:INPUT#9,sk!:FOR i=0 TO 5:INPUT#9,ku
!(i):INPUT#9,ak(i):NEXT
4930 INPUT#9,fm:FOR i=0 TO fm:INPUT#9,ta(i [2743]
):NEXT:CLOSEIN
4940 IF df<>dff THEN PRINT"Falsche Tagesfr [5898]
agen geladen!":GOSUB 7200:df=dff:GOSUB 930
0
4950 GOSUB 7000:GOSUB 7300:GOSUB 4700:RETU [2523]
RN
4990 'Zufall                               [1320]
5000 RANDOMIZE TIME:r!=RND:IF r!>0.45 THEN [4385]
 5400 ELSE CLS:CLS#2:PRINT#2,"EREIGNIS"
5010 IF r!>0.02 THEN 5050 ELSE PRINT"Ein E [13993]
rdbeben fordert zahlreiche Todes-":PRINT"o
pfer!":x!=MIN(99999,MAX(RND*10000,RND*ge!)
/sk!):PRINT"Es sind ";USING"##,###.## $";x
!;:PRINT" f|r die Soforthilfe":PRINT"erfor
derlich.":PRINT
5020 p!(1)=p!(1)-30:p!(5)=p!(5)-30:g!(6)=g [2902]
!(6)+10
5030 IF ge!<x!THEN PRINT"Leider haben Sie  [10486]
nicht soviel Geld -    Sie geben alles, ab
er zu wenig.":g!(3)=g!(3)-30:g!(4)=g!(4)-3
0:g!(1)=g!(1)-50:GOSUB 9400:GOSUB 7900:ge!
=0:GOSUB 7000:GOTO 5230
5040 ge!=ge!-x!:GOSUB 7000:PRINT ee$;" kom [9005]
```

```
men nicht darum herum,";:PRINT"dem P|bel d
as Geld vorzuwerfen!":PRINT:g!(3)=g!(3)-15
:g!(4)=g!(4)-15:GOSUB 9400:GOSUB 7900:GOTO
 5230
5050 IF r!>0.06 THEN 5080 ELSE i=INT(RND*5 [10079]
0)+5:PRINT"Der Guazuela-Peso mu~te um";i;"
% abge-":PRINT"wertet werden.":ge!=ge!-i*g
e!/100:GOSUB 7000
5060 IF su!>0 THEN PRINT:PRINT"Das betriff [8719]
t zum Gl|ck nicht Euer Exzel-lenz 'Notvorr
at' in der Schweiz!"
5070 GOTO 5230                             [397]
5080 IF r!>0.1 OR ak(1)+ak(2)+ak(3)+ak(4)+ [10195]
ak(5)<100 THEN 5090 ELSE PRINT"Eine Baisse
 wirkt sich verheerend auf   die B|rsenkur
se aus.":bs=2+INT(RND*4):GOTO 5230
5090 IF r!>0.15 THEN 5100 ELSE PRINT"Terro [12440]
risten haben eine der modernsten   Fabrike
n des Landes in die Luft ge-     sprengt."
:p!(5)=p!(5)-20:GOSUB 7900:GOTO 5230
5100 IF r!>0.17 THEN 5110 ELSE PRINT"Ein F [11512]
l|chenbrand hat mehr als ein Sechs-tel des
 Ackerlandes vor|bergehend un-   fruchtbar
 gemacht.":p!(1)=p!(1)-20:GOSUB 7900:GOTO
5230
5110 IF r!>0.19 THEN 5120 ELSE PRINT"Skand [10950]
al|se Zust|nde in der ";gp$(2);":":PRINT I
NT(RND*3)+2;"Generale und";INT(RND*8)+2;"O
bristen mu~ten gehen.":p!(2)=p!(2)-20:GOSU
B 7900:GOTO 5230
5120 IF r!>0.27 THEN 5140 ELSE PRINT"Die B [7972]
eziehungen zu ";gp$(7);" haben sich ";:IF
RND<0.5 THEN a=-1:PRINT"verschlechtert."EL
SE a=1:PRINT"verbessert."
5130 g!(7)=g!(7)+a*(INT(RND*50)+10):GOSUB  [2590]
7900:GOTO 5230
5140 IF r!>0.32 THEN 5150 ELSE PRINT"Leide [15845]
r haben die Conduyaner wieder un-  verdien
te, aber betr|chtliche wirt-     schaftlic
he Fortschritte gemacht.":p!(7)=p!(7)+RND*
50+10:GOSUB 7900:GOTO 5230
5150 IF r!>0.37 THEN 5160 ELSE PRINT"Erfre [13979]
ulicherweise haben die Conduyaner  R|cksch
l|ge auf wirtschaftlichem Gebiet hinnehmen
 m|ssen.":p!(7)=p!(7)-RND*50-10:GOSUB 7900
:GOTO 5230
5160 IF r!>0.42 THEN 5210                  [848]
5170 IF bu(0)=-1 THEN PRINT"Die USA bauen  [13418]
einen Milit|rst|tzpunkt   an der K|ste uns
eres Landes.":p!(2)=p!(2)+10:g!(2)=g!(2)+1
0:g!(7)=g!(7)-40:g!(6)=g!(6)+10:p!(5)=p!(5
)+10:g!(5)=g!(5)+10:GOSUB 7900:GOTO 5230
5180 IF bu(1)<>-1 THEN PRINT"Die Blockfrei [7316]
en kommen in Ciudad del    Guazuela zusamm
en.":GOTO 5230
5190 PRINT"Die UdSSR unterst|tzen die komm [11742]
unisti-  schen Berggguerilleros. Wollen Sie
 das   B|ndnis aufk|ndigen?":GOSUB 7600:IF
 ja THEN bu(1)=2+INT(RND*2):x!=-1 ELSE x!=
0.5
5200 g!(1)=g!(1)-x!*30:FOR i=3 TO 4:g!(i)= [7351]
g!(i)-x!*30:NEXT:g!(6)=g!(6)-x!*30:g!(7)=g
!(7)+x!*30:p!(1)=p!(1)-x!*30:GOSUB 9400:GO
SUB 7900:GOTO 5400
5210 IF su=0 OR r!>0.45 THEN 5400 ELSE PRI [11591]
NT ee$;" Hausbank in der Schweiz";"teilt m
it, da~ sie Euer Guthaben mit":i=RND*15*sk
!+1:PRINT USING"##.#";i;
5220 PRINT"% verzinst.":PRINT:PRINT"Altes  [9287]
Guthaben:";:PRINT USING fo$;su!:PRINT:su!=
su!+su!*i/100:PRINT"Neues Guthaben:";:PRIN
T USING fo$;su!:GOTO 5230
5230 GOSUB 7200                            [859]
```

```
5390 'Schalter weiterstellen               [2372]
5400 zz=zz+1:z=z+1:un=0:c=c+INT(RND*100):b [4869]
v=bv+INT(RND*100):FOR i=0 TO 1:IF bu(i)>0
THEN bu(i)=bu(i)-1
5410 NEXT:IF fw THEN f=MAX(f-1,0)          [1953]
5420 rn=MAX(rn-1,0):fd=MAX(0,fd-1):bs=MAX( [3330]
bs-1,0):wh=MAX(wh-1,0)
5440 IF p!(7)<100 THEN p!(7)=p!(7)+RND*20/ [2226]
sk!
5450 g!(7)=(0.9+RND*0.1)*g!(7)             [868]
5500 GOSUB 9500                            [844]
5510 GOTO 500                              [401]
5990 'Auswertung                           [1091]
6000 CLS#2:PRINT#2,"AUSWERTUNG"            [2041]
6010 IF ed=1 THEN PRINT"Sie fl|chteten aus [4385]
 Ihrer Hauptstadt!":PRINT
6020 IF ed=2 THEN GOTO 6300                [1881]
6030 IF NOT fw THEN PRINT"Leider ist die F [12873]
lucht mi~gl|ckt - Sie   wurden von Ihren e
igenen Leibw|chtern   erschossen!":GOSUB 8
300:GOTO 6300
6040 IF fw AND f>0 THEN PRINT"Die Flucht w [5540]
ar wohl etwas |berst|rzt!":pkt=1
6050 IF fw AND f=0 THEN PRINT"Wohlbehalten [4803]
 landen Sie in der Schweiz!":pkt=2
6060 PRINT:PRINT"Sie haben";jr-1965;"Jahre [13023]
 ";:PRINT"und";mo*3-3;"Monate ";"regiert."
:PRINT"Dabei haben Sie";:PRINT USING fo$;s
u!;:PRINT" ";"auf";:PRINT"die Seite schaff
en k|nnen.":PRINT
6070 IF pkt=1 AND ge!>0 THEN PRINT"Leider  [17884]
verlangen die Schweizer Beh|rden":PRINT US
ING fo$;0.4*su!;:PRINT" an Steuern.":PRINT
:PRINT"Es verbleiben";:PRINT USING fo$;su!
*0.6:PRINT"Bereiten Sie das n|chste Mal Ih
re Fluchtbesser vor!"
6080 GOSUB 7200:CLS:x!=(jr-1965)*4/+mo:x!= [8710]
INT((x!/sk!)*SQR(SQR(su!)))*10:PRINT:PEN 2
:PRINT"Ihr Score:";TAB(20);:PEN 1:PRINT x!
;"Punkte":PEN 1:PRINT
6090 hisc(6)=x!:FOR i=0 TO 5:FOR j=6 TO i+ [10665]
1 STEP-1:IF hisc(i)<hisc(j)THEN k=hisc(i):
hisc(i)=hisc(j):hisc(j)=k:MID$(tt$,1,15)=h
isc$(i):MID$(hisc$(i),1,15)=hisc$(j):MID$(
hisc$(j),1,15)=tt$
6100 NEXT:NEXT:GOSUB 9700                  [1536]
6290 'nochmal?                             [211]
6300 PRINT:PRINT"Wollen Sie nochmal spiele [14115]
n?":GOSUB 7600:IF ja THEN 6320 ELSE PRINT:
PRINT"High-Scores abspeichern?":GOSUB 7600
:IF ja THEN OPENOUT"!guahigh":FOR i=0 TO 5
:PRINT#9,hisc$(i):PRINT#9,hisc(i):NEXT:CLO
SEOUT
6310 END                                   [110]
6320 ERASE g!,p!,ma,ku!,ak,kn,si!,bu:GOSUB [3775]
 9800:GOTO 70
6890 'Q fuer Aufgabe                       [1356]
6900 CLS#3:PRINT#3," <Q> f|r Aufgabe, sons [4113]
t warten ":RETURN
6990 'Geldanzeige                          [1447]
7000 CLS#1:PRINT#1,USING fo$;ge!;:RETURN   [3570]
7090 'Warten auf Taste                     [733]
7100 WHILE INKEY$<>"":WEND:SOUND 55,100,1, [5097]
5:MID$(eg$,1,1)=CHR$(200)
7110 MID$(eg$,1,1)=INKEY$:IF eg$=CHR$(200) [3296]
THEN 7110
7120 MID$(e$,1,1)=eg$:MID$(el$,1,1)=LOWER$ [3339]
(eg$):en=ASC(eg$)-48
7130 IF el$="j"OR el$="y"THEN ja=-1 ELSE j [1815]
a=0
7140 SOUND 1,0,2,5,,,31:PAPER#3,0:CLS#3:PA [3359]
PER#3,1:RETURN
7190 'Taste druecken                       [657]
```

```
7200 CLS#3:PRINT#3,TAB(7);"<TASTE> dr}cken [3837]
";:GOSUB 7100:RETURN
7290 'Datumsanzeige                       [1489]
7300 IF mo=4 THEN mo=1:jr=jr+1 ELSE mo=mo+ [428]
1
7310 CLS#4:PRINT#4,mn$(mo);jr:RETURN      [2309]
7390 'Balkenskala                         [1215]
7400 LOCATE 19,h1:PEN 3:PRINT ba$:PEN 1:LO [4571]
CATE 19,h1:PRINT MID$(ba$,1,h2):RETURN
7490 'Zifferntaste druecken               [1996]
7500 CLS#3:PRINT#3,TAB(6);"Zifferntaste dr [4262]
}cken";:GOSUB 7100:RETURN
7590 'Ja/Nein-Abfrage                     [2122]
7600 CLS#3:PRINT#3,TAB(3);"<J> f}r Ja - <N [4289]
> f}r Nein";:GOSUB 7100:RETURN
7690 'Wollen Sie U?                       [1907]
7700 q=0:IF un THEN q=-1:GOTO 7710 ELSE PR [10242]
INT"Wollen Sie Unterst}tzung haben?":GOSUB
 7600:IF NOT ja THEN PRINT"Dann nicht!":FO
R ii=1 TO 1000:NEXT:q=-1 ELSE un=-1
7710 RETURN                               [555]
7790 'Bitte warten                        [1984]
7800 CLS#3:PRINT#3,TAB(10);"BITTE WARTEN!" [2663]
;:RETURN
7890 'Begradigen                          [1170]
7900 FOR h=1 TO 7:g!(h)=MAX(0,MIN(200,g!(h [4469]
))):p!(h)=MAX(0,MIN(200,p!(h))):NEXT:RETUR
N
7990 'Wirkung von Investitionen           [1465]
8000 x1!=x!/(((z+20)/20)*500*sk!):p!(k)=p! [6545]
(k)+x1!:g!(k)=g!(k)+x1!:ge!=ge!-x!:GOSUB 7
900:GOSUB 7000:RETURN
8090 'Stimmungsbild                       [1137]
8100 CLS:IF g!(6)<20 THEN PRINT"Der ";gp$( [8024]
6);" unterst}tzt Sie nicht":PRINT"mehr. Si
e erhalten daher keinen Lagebe- richt.":GO
SUB 7200:GOTO 8220
8110 ko!=30-g!(6)/10:ko!=ko!*100:ko!=ko!-( [3504]
ko!/4)*RND:ko!=ko!+ge!/(sk!*80)
8120 PRINT"Soll Ihnen der ";gp$(6);" einen [9567]
 Lage-":PRINT"bericht liefern?":PRINT:PRIN
T"Kosten:";:PRINT USING"##,### $";ko!:GOSU
B 7600:IF NOT ja THEN 8220
8130 IF ko!>ge!THEN PRINT:PRINT fe$:GOSUB  [3180]
7200:GOTO 8220 ELSE ge!=ge!-ko!
8140 CLS:GOSUB 7000:CLS#2:PRINT#2,"LAGEBER [4266]
ICHT":PEN 2:PRINT"Beliebtheit":PEN 1
8150 FOR i=1 TO 6:LOCATE 1,i+2:PRINT gp$(i [4301]
);":":NEXT:LOCATE 1,10:PRINT gp$(7);":"
8160 FOR i=1 TO 6:h1=i+2:h2=INT(g!(i)/10): [5466]
GOSUB 7400:NEXT:h1=10:h2=INT(g!(7)/10):GOS
UB 7400
8170 LOCATE 1,11:PEN 2:PRINT v$:PEN 1:LOCA [6412]
TE 1,12:PRINT"Ausr}stung ";gp$(2);":":h1=1
2:h2=INT(p!(2)/10):GOSUB 7400
8180 LOCATE 1,14:PEN 2:PRINT"Wirtschaftssi [3424]
tuation":PEN 1
8190 LOCATE 1,16:PRINT"Agrarwirtschaft":h1 [4491]
=16:h2=INT(p!(1)/10):GOSUB 7400
8200 LOCATE 1,17:PRINT"Industrie":h1=17:h2 [3675]
=INT(p!(5)/10):GOSUB 7400
8210 GOSUB 7200                           [859]
8220 RETURN                               [555]
8290 'MG-Feuer                            [746]
8300 FOR h1=1 TO 3:GOSUB 8310:NEXT:RETURN  [3770]
8310 h=kn(INT(RND*3)+1):FOR h2=1 TO INT(RN [3744]
D*20)+5:hr=INT(RND*15)+1
8320 FOR h3=31-hr TO 26-hr STEP-1:SOUND h, [7072]
0,1,15,,,h3:NEXT:SOUND 55,0,2,0:NEXT:FOR h
2=1 TO INT(RND*600)+200:NEXT:WHILE SQ(4)>1
27:WEND:RETURN
8390 'Feindlichste                        [1388]
8400 x!=0:gr=0:FOR i=1 TO 6:IF ma(i)>x!THE [2659]
N x!=ma(i):gr=i
8410 NEXT:RETURN                          [940]
8490 'Kriegslaerm                         [990]
8500 h4=1                                 [394]
8510 IF h4<=3 THEN ON INT(RND*4)+1 GOSUB 8 [6437]
310,8310,8530,8540:WHILE SQ(4)>127 OR SQ(2
)>127 OR SQ(1)>127:WEND:h4=h4+1:GOTO 8510
8520 RETURN                               [555]
8530 h=kn(INT(RND*3)+1):FOR h1!=1 TO 7 STE [6148]
P 0.3:SOUND h,2000,5,h1!,,,31-h1!*4:NEXT:S
OUND 55,2000,5,h1!,,,1:RETURN
8540 h=kn(INT(RND*3)+1):FOR h1!=1 TO 7 STE [6326]
P 0.2:SOUND h,h1!*6+10,h1!,h1!,,,h1!*4:NEX
T:FOR h1=1 TO 70:SOUND 55,0,1,MIN(7,h1/5),
,,31-(h1 MOD 10)-1:NEXT:RETURN
8550 RETURN                               [555]
8560 FOR h1=1 TO 3:SOUND kn(INT(RND*3)+1), [4695]
500,6,7:FOR h=500 TO 100 STEP-15
8570 SOUND kn(INT(RND*3)+1),h,3,7:NEXT:SOU [7563]
ND kn(INT(RND*3)+1),100,10,7:SOUND 55,0,30
:NEXT:RETURN
8690 'Krieg!!!                            [616]
8700 ed=0:GOSUB 6900:CLS:CLS#2:PRINT#2,"KR [4500]
IEG":GOSUB 9000:fd=INT(RND*6)+4
8710 WINDOW#0,1,40,14,23:GOSUB 9200        [1566]
8720 RANDOMIZE TIME:sla=1:IF RND<mg!THEN s [3071]
=5 ELSE s=4
8730 PRINT"Der Krieg beginnt...":GOSUB 856 [4885]
0:PRINT:PLOT-10,-10,1
8740 PRINT xn$;:a=st(s,1):b=st(s,2):GOSUB  [1861]
8900
8750 op=INT(RND*300)+5:CLS:PRINT"Gua:";:PR [8350]
INT USING" ###,###,###";bv*1000;:PRINT TAB
(21);"Con:";:PRINT USING" ###,###,###";c*1
000:PRINT
8751 PRINT"Die";sla;CHR$(8);". Schlacht -  [7166]
um ";UPPER$(sn$(s));" - ";"forderte ";:PRI
NT USING"###,### ";op*1000;:PRINT"Opfer."
8760 r!=RND:PRINT"Die Streitkr}fte von ";: [5500]
IF r!<mg!THEN PRINT"Guazuela";ELSE PRINT g
p$(7);
8770 PRINT" siegten.":PRINT:mx!=0.5*RND+0. [5879]
5:IF s<=4 THEN bv=bv-mx!*op:c=c-(1-mx!)*op
 ELSE c=c-mx!*op:bv=bv-(1-mx!)*op
8780 IF bv<4000 AND r!>=mg!THEN PRINT"Guaz [15160]
uela ist v}llig ausgeblutet. Der ge-gneris
che Milit}rgouverneur verurteilt  Sie dahe
r nach Ihrer Kapitulation zum   Tode.":ed=
2:GOTO 8860
8790 IF c<4000 AND r!<mg!AND sla>1 THEN PR [8243]
INT gp$(7);" kapituliert und zahlt Repara-
":PRINT"tionen.":GOTO 8830
8800 GOSUB 8500:GOSUB 9200:GOSUB 8900:PRIN [4070]
T xf$;:IF r!<mg!THEN s=s+1 ELSE s=s-1
8810 IF s=0 THEN PRINT"Die entscheidende S [9525]
chlacht wurde vom    Feind gewonnen. Sie
mu~ten fl}chten.":ed=1:GOTO 8860
8820 IF s<9 THEN 8840 ELSE PRINT"Sie haben [8612]
 den Feind niedergeworfen. Die Beute ist g
ro~!":p!(2)=RND*0.8*p!(2)*sk!
8830 PRINT:FOR i=1 TO 6:g!(i)=g!(i)+RND*60 [6614]
-RND*sla*5:NEXT:ge!=ge!+RND*60000+30000*sk
!:p!(7)=0:g!(7)=0:GOSUB 7900:GOSUB 7000:GO
TO 8860
8840 sla=sla+1:IF LOWER$(INKEY$)<>"q"THEN  [12913]
GOTO 8740 ELSE CLS:IF ge!<30000 THEN PRINT
"Der Staatsschatz reicht nicht hin, um  ";
gp$(7);" zufriedenzustellen.":GOSUB 7200:G
OSUB 6900:GOTO 8740
8850 mx!=MAX(30000,ge!*(0.4/sk!)):PRINT gp [9255]
$(7);"verlangt ";:PRINT USING fo$;mx!:PRIN
T"als Kriegsentsch}digung.":ge!=ge!-mx!:GO
SUB 7000
```

```
8860 GOSUB 7200:WINDOW 1,40,4,24:CLS#6:RET [2751]
URN
8890 'Schlachtort                          [595]
8900 TAG:MOVE a-8,b+8:PRINT CHR$(238);:TAG [2563]
OFF:RETURN
8990 'Bild                                 [788]
9000 FOR i=0 TO 3:INK i,0:NEXT:PLOT pk(0,0 [9969]
),pk(0,1),1:h1=1:h2=4:GOSUB 9900:PLOT pk(5
,0),pk(5,1):h1=6:h2=64:GOSUB 9900:PLOT pk(
65,0),pk(65,1):h1=66:h2=90:GOSUB 9900:PLOT
 pk(91,0),pk(91,1):h1=92:h2=119:GOSUB 9900
9010 PLOT 148,313,2:PLOT 150,313:PLOT 150, [6442]
311:PLOT 148,311:PLOT 188,325:PLOT 190,325
:PLOT 190,323:PLOT 188,323:PLOT 228,271
9020 PLOT 230,271:PLOT 230,269:PLOT 228,26 [12657]
9:PLOT 288,263:PLOT 290,263:PLOT 290,265:P
LOT 288,265:PLOT 354,249,3:PLOT 354,251:PL
OT 356,251:PLOT 356,249:PLOT 456,289:PLOT
458,289:PLOT 458,287:PLOT 456,287:PLOT 396
,329
9030 PLOT 398,329:PLOT 398,327:PLOT 396,32 [5454]
7:PLOT 382,301:PLOT 384,301:PLOT 384,299:P
LOT 382,299
9040 TAG:PLOT-1,-1,2:PEN 2:MOVE 116,287:PR [8160]
INT"GUAZUELA";:FOR i=1 TO 4:MOVE st(i,1)+8
,st(i,2):PRINT LEFT$(sn$(i),3);:NEXT
9050 PLOT-1,-1,3:MOVE 416,240:PRINT"CONDUY [7218]
ANA";:FOR i=5 TO 8:MOVE st(i,1)+8,st(i,2):
PRINT LEFT$(sn$(i),3);:NEXT:TAGOFF
9060 PEN 1:FOR i=0 TO 3:INK i,PEEK(him!+i+ [4292]
1):NEXT:RETURN
9190 'Motivation                           [1222]
9200 mc=(7*p!(7)+200-g!(7))*c/5000:mg!=0:F [7741]
OR i=1 TO 6:mg!=mg!+g!(i):NEXT:mg!=(mg!/6)
```

```
*bv/7000:mg!=mg!+4*p!(2)+3*sk!*p!(2):mm=mg
!+mc:mg!=mg!/mm:IF mg!>0.5 THEN mg!=mg!+(1
-mg!)/2 ELSE mg!=mg!/2
9210 RETURN                                [555]
9290 'Fragen laden                         [383]
9300 IF z>0 THEN GOSUB 9600:GOSUB 7800     [1579]
9310 d$="!guafrag."+STR$(df):d$=d$:OPENIN  [11287]
d$:i=0:WHILE NOT EOF:LINE INPUT#9,fr$(i):F
OR j=1 TO 10:INPUT#9,fp!(i,j):NEXT:i=i+1:W
END:fm=i-1:CLOSEIN:GOSUB 9800:RETURN
9390 'Zufallsminderung                     [1672]
9400 FOR i=1 TO 7:g!(i)=INT(g!(i)*0.95+RND [4048]
*0.1*g!(i)):p!(i)=INT(p!(i)*0.95+RND*0.1*g
!(i)):NEXT:RETURN
9490 'Ueberdruss                           [625]
9500 FOR i=1 TO 7:IF g!(i)>100 THEN g!(i)= [2445]
g!(i)-RND*0.1*(g!(i)-100)
9510 IF p!(i)>100 THEN p!(i)=p!(i)-RND*0.1 [2791]
*(p!(i)-100)
9520 NEXT:RETURN                           [940]
9590 'Ladenanzeige                         [506]
9600 CLS#2:PRINT#2,"LADEN":RETURN          [1493]
9690 'High-Score-Ausgabe                   [1253]
9700 PRINT:PRINT"DIE BESTEN PR[SIDENTEN":P [8561]
RINT LEFT$(v$,22):PRINT:FOR ii=0 TO 5:PRIN
T hisc$(ii);TAB(20);hisc(ii);"Punkte":NEXT
:RETURN
9790 'Mischen                              [663]
9800 RANDOMIZE TIME:FOR i=0 TO fm:ta(i)=i: [10158]
NEXT:FOR i=0 TO fm-1:k=INT(RND*(fm-i)+1)+i
:t=ta(i):ta(i)=ta(k):ta(k)=t:NEXT:RETURN
9890 'HilfsUP zu Bild                      [1203]
9900 FOR i=h1 TO h2:DRAW pk(i,0),pk(i,1):N [2705]
EXT:RETURN
```

**Sie wollen uns ein Programm zusenden? Folgende Regeln sollten Sie beachten:**

1) Schicken Sie Ihre Programme auf Kassette oder Diskette. Dabei mehrmals hintereinander abspeichern.

2) Legen Sie eine ausführliche Programmbeschreibung bei, aus der eindeutig der Sinn und Zweck des Programmes hervorgeht. Eine Variablenliste ist sehr wichtig, in ihr sollten alle im Programm definierten Variablen mit ihrer Verwendung aufgeführt sein.
Beispiel: n$ = Nachname
g = glücklicher usw.

3) Wenn Sie gleichzeitig Besitzer eines Druckers sind, schicken Sie auf jeden Fall ein Programmlisting mit.

4) Vergessen Sie Ihren Absender nicht! Vollständige Anschrift, wenn möglich auch die Telefonnummer, ersparen unnötige Verzögerungen bei eventuellen Nachfragen unsererseits.

5) Behalten Sie auf jeden Fall eine Kopie Ihres Programmes, unvorhersehbare Umstände, wie z.B. Bandriß der Kassette, werden somit nicht zur Katastrophe.

6) Wenn Sie sogar mehrere Programme geschrieben haben und diese veröffentlichen wollen, nehmen Sie bitte für jedes Programm eine separate Kassette (mehrmals abspeichern!).

*Damit erleichtern Sie unsere Arbeit wesentlich und eine schnellere Bearbeitung wird möglich.

7) Versehen Sie das Programm mit den sehr wichtigen REM-Statements. Das fördert nicht nur die Übersicht für uns und unsere Leser, sondern unterstützt zusätzlich das strukturierte Programmieren. Merke: Auch erfahrene Programmierer verwenden REM-Zeilen, das Programm sollte schließlich jeder verstehen und anwenden können.

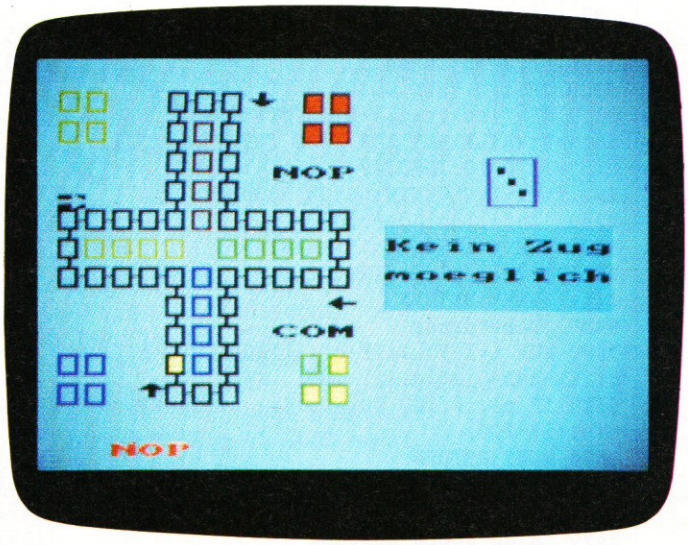
8) Vermeiden Sie möglichst Variablennamen, wo leicht l und i oder 0 und o verwechselt werden können. Variablen sollten grundsätzlich in Kleinbuchstaben geschrieben werden und möglichst verständlich sein. Die Variable butter sagt mehr aus, als nur bu.

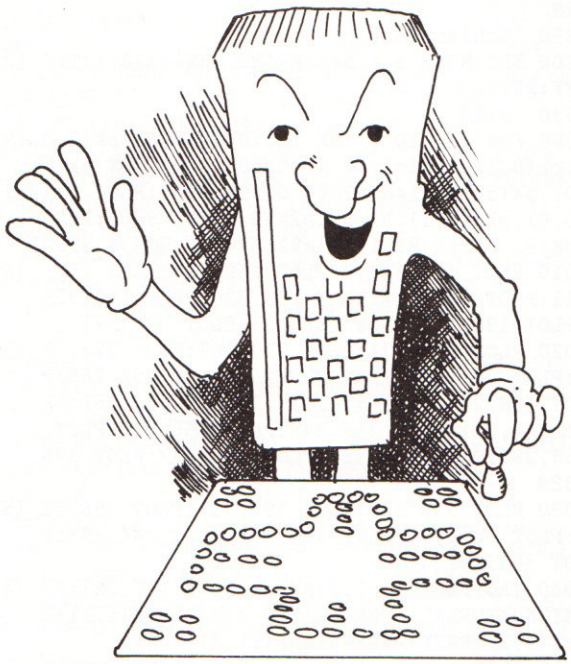
9) Bieten Sie das Programm niemals verschiedenen Verlagen gleichzeitig an! Mit der Veröffentlichung und dem daraus resultierenden Honorar, gehen die Urheberrechte an den Verlag über! Sollte Ihr Programm dann noch in einer anderen Zeitschrift abgedruckt werden, liegt ein Verstoß gegen das Wettbewerbsrecht vor.

10) Jedes bei uns ordnungsgemäß eingegangene Programm wird ausführlich begutachtet und getestet. Sie erhalten von uns dann umgehend Bescheid, ob Ihr Programm veröffentlicht wird oder nicht. Wenn Sie das Programm nach unserer Begutachtung zurückhaben wollen, legen Sie bitte der Einsendung einen frankierten und adressierten Rückumschlag bei. Die Kosten einer eventuellen Rücksendung sind in jedem Fall vom Einsender zu tragen.

**Falls wir uns für den Abdruck Ihres Programmes entschieden haben, setzen wir uns unverzüglich mit Ihnen in Verbindung und besprechen Ihr verdientes Honorar!**
**Übrigens wäre es nett, wenn Sie uns auch ein paar persönliche Daten schreiben würden (z.B. Ihr Alter, Tätigkeit, wie und wann kam die Idee zu dem Programm, was alles machen Sie mit dem CPC usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre „Schneider CPC international" auf Ihre Beiträge.**

**Einsendungen bitte an: DMV – Daten & Medien Verlagsges.mbH, Fuldaer Straße 6, 3440 Eschwege**



# Mensch ärgere den CPC nicht!

**Das folgende Programm ist dem bekannten Gesellschaftsspiel nachempfunden und ist nach unserer Meinung hervorragend von Werner Schmidt umgesetzt worden.**

Nach der Eingabe des Listings sollte man es vorsichtshalber vor dem ersten Start speichern,denn bei einem Fehler in den Datazeilen kann das Programm abstürzen. Nach einer kurzen Wartezeit,in der die Daten eingelesen werden, erscheint das Titelbild,welches man mit der Leertaste wieder verlassen kann. Nach dem Aufbau des Spielfeldes wird man nach den Namen der Mitspieler gefragt. Nur ENTER bedeutet,daß diese Farbe nicht mitspielt. Eingabe von COMPUTER bewirkt, daß der Computer diesen Spieler übernimmt. Das Programm führt alle Züge, die sich als Pflichtzüge erweisen, automatisch aus. Kann der Spieler mehr als einen Stein bewegen, so fängt einer der möglichen Steine an zu blinken.

Mit der Leertaste läßt sich nun ein Stein auswählen. Wenn der richtige blinkt, wird mit der Entertaste der Zug ausgeführt. Am Ende einen Zuges ist es möglich durch Drücken der E-Taste das Spiel vorzeitig abzubrechen.

(Werner Schmidt)

## für 464-664-6128

```
10 FOR n=0 TO 11:READ a:INK n,a:NEXT:INK 1 [7170]
2,13,15:INK 13,13,0:BORDER 13:MODE 0:LOCAT
E 5,18:PRINT"bitte warten":SPEED INK 15,15
20 DATA 13,0,6,18,2,24,3,9,1,12,4,10       [1382]
30 MEMORY 39999:FOR n=40000 TO 40274:READ  [4390]
a:POKE n,a:NEXT:CALL 40000
40 DATA 1,73,156,33,84,156,195,209,188,78  [1943]
50 DATA 156,195,88,156,80,82,73,78,212,0   [2058]
60 DATA 0,0,0,0,254,3,192,221,94,2         [2070]
70 DATA 221,126,4,221,70,1,221,78,0,197    [1931]
80 DATA 203,67,202,130,156,205,173,156,209 [2283]
,229
90 DATA 6,8,205,156,156,225,1,80,0,9       [1429]
100 DATA 6,4,205,156,156,201,29,205,173,15 [2164]
6
110 DATA 209,229,1,0,32,9,6,4,205,156      [1217]
120 DATA 156,225,1,80,0,9,6,8,205,156      [1681]
130 DATA 156,201,197,6,3,26,119,35,19,16   [2013]
140 DATA 250,1,253,7,9,193,16,240,201,6    [1440]
150 DATA 8,14,40,22,0,29,33,0,0,41         [1467]
160 DATA 203,33,210,190,156,25,16,247,95,2 [1773]
5
170 DATA 17,255,191,25,201,62,2,50,16,164  [1024]
180 DATA 205,31,157,58,16,164,30,27,1,72   [1499]
190 DATA 161,205,103,156,58,16,164,198,3,3 [1940]
0
200 DATA 24,1,220,160,205,103,156,58,16,16 [2261]
4
210 DATA 198,3,30,30,1,36,161,205,103,156  [1088]
220 DATA 58,16,164,198,6,30,27,1,0,161     [1645]
230 DATA 205,103,156,205,27,187,254,32,200 [2213]
,205
240 DATA 25,189,205,25,189,58,16,164,60,50 [1610]
250 DATA 16,164,254,73,194,204,156,205,31, [1585]
157
260 DATA 195,199,156,58,16,164,61,30,27,1  [1336]
270 DATA 252,161,205,103,156,58,16,164,198 [2320]
,2
280 DATA 30,24,1,252,161,205,103,156,58,16 [1901]
290 DATA 164,198,2,30,30,1,252,161,205,103 [1650]
300 DATA 156,58,16,164,198,5,30,27,1,252   [2028]
310 DATA 161,205,103,156,201               [886]
320 FOR n=41000 TO 41467:READ a:POKE n,a:N  [3828]
EXT:FOR n=41468 TO 41503:POKE n,0:NEXT
330 DATA 192,192,192,192,192,192,128,0,64, [5443]
128,0,64,128,0,64,128,0,64,128,0,64,128,0,
64,128,0,64,128,0,64,192,192,192,192,192,1
92
340 DATA 60,60,60,60,60,60,40,0,20,40,0,20 [4301]
,40,0,20,40,0,20,40,0,20,40,0,20,40,0,20,4
```

```
0,0,20,60,60,60,60,60,60
350 DATA 252,252,252,252,252,252,168,0,84, [6221]
168,0,84,168,0,84,168,0,84,168,0,84,168,0,
84,168,0,84,168,0,84,252,252,252,252,252,2
52
360 DATA 3,3,3,3,3,3,2,0,1,2,0,1,2,0,1,2,0 [2520]
,1,2,0,1,2,0,1,2,0,1,2,0,1,3,3,3,3,3,3
370 DATA 195,195,195,195,195,195,130,0,65, [4856]
130,0,65,130,0,65,130,0,65,130,0,65,130,0,
65,130,0,65,130,0,65,195,195,195,195,195,1
95
380 DATA 192,192,192,192,192,192,132,12,72 [7906]
,132,12,72,132,12,72,132,12,72,132,12,72,1
32,12,72,132,12,72,132,12,72,192,192,192,1
92,192,192
390 DATA 192,192,192,192,192,192,196,204,2 [8102]
00,196,204,200,196,204,200,196,204,200,196
,204,200,196,204,200,196,204,200,196,204,2
00,192,192,192,192,192,192
400 DATA 192,192,192,192,192,192,144,48,96 [7346]
,144,48,96,144,48,96,144,48,96,144,48,96,1
44,48,96,144,48,96,144,48,96,192,192,192,1
92,192,192
410 DATA 192,192,192,192,192,192,208,240,2 [7384]
24,208,240,224,208,240,224,208,240,224,208
,240,224,208,240,224,208,240,224,208,240,2
24,192,192,192,192,192,192
420 DATA 60,60,60,60,60,60,44,12,28,44,12, [4622]
28,44,12,28,44,12,28,44,12,28,44,12,28,44,
12,28,44,12,28,60,60,60,60,60,60
430 DATA 252,252,252,252,252,252,236,204,2 [7784]
20,236,204,220,236,204,220,236,204,220,236
,204,220,236,204,220,236,204,220,236,204,2
20,252,252,252,252,252,252
440 DATA 3,3,3,3,3,3,18,48,33,18,48,33,18, [4874]
48,33,18,48,33,18,48,33,18,48,33,18,48,33,
18,48,33,3,3,3,3,3,3
450 DATA 195,195,195,195,195,195,210,240,2 [7603]
25,210,240,225,210,240,225,210,240,225,210
,240,225,210,240,225,210,240,225,210,240,2
25,195,195,195,195,195,195
460 DIM p(40,2),nam$(4),pin(40),w$(6),sp(4 [6322]
,4),sza(4),szb(4),h(4,4,2),l(4),hei(4,4),d
ar(4),mus(4),hm(4,4,2),hil(4)
470 FOR n=1 TO 4:FOR m=1 TO 4:FOR t=1 TO 2 [3712]
:READ hm(n,m,t):NEXT t,m,n
480 DATA 21,5,21,9,21,13,21,17,37,21,33,21 [4174]
,29,21,25,21,21,37,21,33,21,29,21,25,5,21,
9,21,13,21,17,21
490 FOR n=1 TO 40:FOR m=1 TO 2:READ p(n,m) [3494]
:NEXT m,n
500 DATA 25,1,25,5,25,9,25,13,25,17,29,17, [6967]
33,17,37,17,41,17,41,21,41,25,37,25,33,25,
29,25,25,25,25,29,25,33,25,37,25,41,21,41
510 DATA 17,41,17,37,17,33,17,29,17,25,13, [4626]
25,9,25,5,25,1,25,1,21,1,17,5,17,9,17,13,1
7,17,17,17,13,17,9,17,5,17,1,21,1
520 FOR n=1 TO 4:FOR m=1 TO 4:FOR t=1 TO 2 [3717]
:READ h(n,m,t):NEXT t,m,n
530 DATA 37,1,41,1,37,5,41,5,37,37,41,37,3 [3832]
7,41,41,41,1,37,5,37,1,41,5,41,1,1,5,1,1,5
,5,5
540 FOR n=1 TO 4:READ l(n):NEXT             [1313]
550 DATA 40,10,20,30                        [406]
560 SYMBOL 244,0,0,0,0,0,0,24,24:SYMBOL 24 [3782]
5,24,24,0,0,0,0,0,0
570 SYMBOL 246,192,192,192,192,0,0,0,0:SYM [3008]
BOL 247,0,0,0,0,3,3,3,3
580 SYMBOL 248,192,192,192,192,0,0,24,24:S [3497]
YMBOL 249,24,24,0,0,3,3,3,3
590 SYMBOL 250,195,195,195,195,0,0,0,0:SYM [4044]
BOL 251,0,0,0,0,195,195,195,195
600 SYMBOL 252,195,195,195,195,0,0,24,24:S [4379]
YMBOL 253,24,24,0,0,195,195,195,195
610 SYMBOL 254,195,195,195,195,0,0,195,195 [3914]
:SYMBOL 255,195,195,0,0,195,195,195,195
620 FOR n=1 TO 6:w$(n)=CHR$(242+2*n)+CHR$( [4976]
10)+CHR$(8)+CHR$(243+2*n):NEXT:w$(0)=CHR$(
32)+CHR$(10)+CHR$(8)+CHR$(32)
630 'Titelbild                              [797]
640 MODE 0:INK 2,6,18:INK 3,18,2:INK 4,2,2 [2400]
4:INK 5,24,6
650 LOCATE 5,1:PRINT"M E N S C H":LOCATE 5 [6897]
,3:PRINT"aergere den":LOCATE 8,5:PRINT"C P
 C":LOCATE 8,7:PRINT"nicht"
660 LOCATE 1,24:PRINT CHR$(164);" WERNER S [8178]
CHMIDT":LOCATE 17,25:PRINT"1/87";:PEN 12:L
OCATE 2,22:PRINT"Leertaste druecken":PEN 1
670 CALL &9CC7                              [788]
680 MODE 0:WINDOW#1,13,20,10,15:WINDOW#2,1 [2690]
,20,25,25
690 INK 2,6:INK 3,18:INK 4,2:INK 5,24:RAND [3374]
OMIZE TIME
700 'Spielfeld                              [77]
710 FOR n=1 TO 40:|PRINT,p(n,1),p(n,2),410 [2455]
00:NEXT
720 FOR n=375 TO 375-3*32 STEP -32:PLOT 20 [4029]
0,n,1:DRAWR 0,-8:NEXT
730 FOR n=216 TO 216+3*32 STEP 32:PLOT n,2 [3312]
59:DRAWR 8,0:NEXT
740 PLOT 328,247:DRAWR 0,-8:PLOT 328,215:D [2491]
RAWR 0,-8
750 FOR n=216 TO 216+3*32 STEP 32:PLOT n,1 [2416]
97::DRAWR 8,0:NEXT
760 FOR n=183 TO 183-3*32 STEP -32:PLOT 20 [1911]
0,n:DRAWR 0,-8:NEXT
770 PLOT 192,69:DRAWR -8,0:PLOT 160,69:DRA [1428]
WR -8,0
780 FOR n=183 TO 183-3*32 STEP -32:PLOT 14 [2319]
0,n:DRAWR 0,-8:NEXT
790 FOR n=124 TO 124-3*32 STEP -32:PLOT n, [3796]
197:DRAWR -8,0:NEXT
800 PLOT 14,207:DRAWR 0,8:PLOT 14,239:DRAW [1985]
R 0,8
810 FOR n=124 TO 124-3*32 STEP -32:PLOT n, [3758]
259:DRAWR -8,0:NEXT
820 FOR n=375 TO 375-3*32 STEP -32:PLOT 14 [3437]
0,n:DRAWR 0,-8:NEXT
830 PLOT 148,389:DRAWR 8,0:PLOT 180,389:DR [1876]
AWR 8,0
840 FOR n=5 TO 17 STEP 4:|PRINT,21,n,41036 [1969]
:NEXT
850 FOR n=25 TO 37 STEP 4:|PRINT,n,21,4107 [3231]
2:NEXT
860 FOR n=25 TO 37 STEP 4:|PRINT,21,n,4110 [3486]
8:NEXT
870 FOR n=5 TO 17 STEP 4:|PRINT,n,21,41144 [1303]
:NEXT
880 FOR n=37 TO 41 STEP 4:FOR m=1 TO 5 STE [4262]
P 4:|PRINT,n,m,41036:NEXT m,n
890 FOR n=37 TO 41 STEP 4:FOR m=37 TO 41 S [4559]
TEP 4:|PRINT,n,m,41072:NEXT m,n
900 FOR n=1 TO 5 STEP 4:FOR m=37 TO 41 STE [3706]
P 4:|PRINT,n,m,41108:NEXT m,n
910 FOR n=1 TO 5 STEP 4:FOR m=1 TO 5 STEP  [5354]
4:|PRINT,n,m,41144:NEXT m,n
920 LOCATE 8,1:PRINT CHR$(241);:LOCATE 11, [4624]
15:PRINT CHR$(242);:LOCATE 4,21:PRINT CHR$
(240);:LOCATE 1,8:PRINT CHR$(243);
930 PLOT 500,328,10:DRAWR 52,0:DRAWR 0,-52 [3406]
:DRAWR -52,0:DRAWR 0,52
940 LOCATE 17,6:PRINT w$(0):anf=0:zah=0     [2335]
950 PAPER#1,11:FOR n=1 TO 4:CLS#1:LOCATE#1 [5078]
,3,2:PRINT#1,n;CHR$(8);"."
960 LOCATE#1,1,3:PRINT#1,"Spieler"          [2468]
970 LOCATE#1,1,5:LINE INPUT#1,nam$(n)       [2027]
980 IF nam$(n)<>"" THEN anf=anf+1           [1761]
990 NEXT                                    [350]
1000 FOR n=1 TO 4:IF nam$(n)<>"" THEN 1010 [3533]
 ELSE NEXT:GOTO 950
1010 CLS#1                                  [373]
1020 FOR n=1 TO 4:nam$(n)=UPPER$(nam$(n)): [6239]
```

```
IF LEN(nam$(n))>8 THEN nam$(n)=LEFT$(nam$(
n),8)
1030 NEXT                                    [350]
1040 IF nam$(1)<>"" THEN FOR n=1 TO 4:|PRI [5684]
NT,h(1,n,1),h(1,n,2),41324:NEXT n:LOCATE 9
,6:PRINT LEFT$(nam$(1),3);
1050 IF nam$(2)<>"" THEN FOR n=1 TO 4:|PRI [5548]
NT,h(2,n,1),h(2,n,2),41360:NEXT n:LOCATE 9
,17:PRINT LEFT$(nam$(2),3);
1060 IF nam$(3)<>"" THEN FOR n=1 TO 4:|PRI [7263]
NT,h(3,n,1),h(3,n,2),41396:NEXT n:LOCATE 1
,17:PRINT LEFT$(nam$(3),3);
1070 IF nam$(4)<>"" THEN FOR n=1 TO 4:|PRI [7184]
NT,h(4,n,1),h(4,n,2),41432:NEXT n:LOCATE 1
,6:PRINT LEFT$(nam$(4),3);
1080 FOR n=1 TO 4:FOR m=0 TO 4:hei(n,m)=0: [5961]
sp(n,m)=0:sza(n)=4:szb(n)=0:NEXT m,n
1090 FOR n=1 TO 40:pin(n)=0:NEXT             [2193]
1100 'Hauptroutine                           [873]
1110 FOR n=1 TO 4                            [788]
1120 IF nam$(n)="" OR hei(n,0)=4 THEN 1440 [2391]
1130 CLS#2:PEN#2,n+1:LOCATE#2,3,1:PRINT#2, [3199]
nam$(n)
1140 FOR m=0 TO 4:mus(m)=0:dar(m)=0:NEXT m [1373]
1150 IF szb(n)=0 THEN FOR m=1 TO 2:GOSUB 1 [3402]
830:IF w=6 THEN 1170 ELSE NEXT m
1160 GOSUB 1830                              [901]
1170 IF pin(10*(n-1)+1)=n AND sza(n)>0 THE [2054]
N GOSUB 1450:GOTO 1210
1180 IF w=6 THEN GOSUB 1540:IF mus(0)>0 TH [1635]
EN 1210
1190 GOSUB 1570                              [897]
1200 GOSUB 1640                              [819]
1210 IF nam$(n)="COMPUTER" THEN 1230         [2826]
1220 IF mus(0)=1 OR dar(0)=1 THEN LOCATE#1 [8340]
,1,3:PRINT#1,"Zugzwang" ELSE IF mus(0)>1 O
R dar(0)>1 THEN LOCATE#1,2,2:PRINT#1,"waeh
le";:LOCATE#1,2,4:PRINT#1,"Stein";
1230 IF mus(0)=1 THEN FOR m=1 TO 4:IF mus( [5328]
m)=1 THEN stein=m:GOTO 1300 ELSE NEXT m
1240 IF dar(0)=1 THEN FOR m=1 TO 4:IF dar( [2051]
m)=1 THEN stein=m:GOTO 1300 ELSE NEXT m
1250 IF dar(0)=0 THEN LOCATE#1,1,2:PRINT#1 [6921]
,"Kein Zug";:LOCATE#1,1,4:PRINT#1,"moeglic
h";:FOR t=1 TO 1000:NEXT:CLS#1:GOTO 1430
1260 IF nam$(n)="COMPUTER" THEN GOSUB 2580 [3604]
:GOTO 1300
1270 IF INKEY$<>"" THEN 1270                 [936]
1280 IF mus(0)>0 THEN GOSUB 1880:GOTO 1300 [2986]
1290 GOSUB 2020                              [871]
1300 IF sp(n,stein)=0 THEN |PRINT,h(n,stei [6495]
n,1),h(n,stein,2),41000+n*36:t=10*(n-1)+1:
FOR x=400 TO 50 STEP -50:SOUND 4,x,8:NEXT
x:GOTO 1390
1310 IF sp(n,stein)<=l(n) AND sp(n,stein)+ [3560]
w>l(n) THEN 2270
1320 IF sp(n,stein)<0 THEN 2430              [2006]
1330 t=sp(n,stein):pin(t)=0                  [1253]
1340 FOR m=1 TO w                            [797]
1350 IF pin(t)=0 THEN |PRINT,p(t,1),p(t,2) [4777]
,41000 ELSE |PRINT,p(t,1),p(t,2),41144+pin
(t)*36
1360 t=t+1:IF t>40 THEN t=t-40               [519]
1370 |PRINT,p(t,1),p(t,2),41144+n*36:SOUND [3301]
 2,200,20
1380 FOR z=1 TO 500:NEXT z,m:GOTO 1400       [2918]
1390 |PRINT,p(t,1),p(t,2),41144+n*36         [2098]
1400 sp(n,stein)=t:IF pin(t)>0 THEN GOSUB  [2774]
2160
1410 pin(t)=n                                [406]
1420 CLS#1:IF zah=anf OR UPPER$(INKEY$)="E [3789]
" THEN 2530
1430 IF w=6 THEN 1140                        [565]
1440 NEXT n:GOTO 1110                        [1279]
1450 'Feld frei machen                       [1163]
1460 t=10*(n-1)+1+w:mus(0)=1                 [1216]
```

```
1470 IF pin(t)<>n THEN FOR m=1 TO 4:IF sp( [7375]
n,m)=10*(n-1)+1 THEN mus(m)=1:RETURN ELSE
NEXT m
1480 z=t:t=t+w:IF t>40 THEN t=t-40           [3863]
1490 IF pin(t)<>n THEN FOR m=1 TO 4:IF sp( [5756]
n,m)=z THEN mus(m)=1:RETURN ELSE NEXT m
1500 z=t:t=t+w:IF t>40 THEN t=t-40           [3863]
1510 IF pin(t)<>n THEN FOR m=1 TO 4:IF sp( [5756]
n,m)=z THEN mus(m)=1:RETURN ELSE NEXT m
1520 FOR m=1 TO 4:IF sp(n,m)=t THEN mus(m) [2713]
=1
1530 NEXT m:RETURN                           [461]
1540 'Sechs                                  [383]
1550 IF sza(n)>0 THEN sza(n)=sza(n)-1:szb( [7054]
n)=szb(n)+1:mus(0)=1:FOR m=1 TO 4:IF sp(n,
m)=0 THEN mus(m)=1:GOTO 1560 ELSE NEXT m
1560 RETURN                                  [555]
1570 'Rausschmeissen ?                       [832]
1580 FOR m=1 TO 4                            [564]
1590 IF sp(n,m)>0 THEN t=sp(n,m)+w ELSE GO [3367]
TO 1630
1600 IF sp(n,m)<=l(n) AND t>l(n) THEN 1630 [2343]
1610 IF t>40 THEN t=t-40                     [1494]
1620 IF pin(t)<>n AND pin(t)>0 THEN mus(0) [2807]
=mus(0)+1:mus(m)=1
1630 NEXT m:RETURN                           [461]
1640 'erlaubte Zuege                         [1175]
1650 FOR m=1 TO 4                            [564]
1660 IF sp(n,m)<0 THEN 1780                  [1174]
1670 IF sp(n,m)<=l(n) AND sp(n,m)+w>l(n) T [3330]
HEN 1730
1680 IF sp(n,m)=0 THEN 1720                  [1293]
1690 t=sp(n,m)+w:IF t>40 THEN t=t-40         [1534]
1700 IF pin(t)=n THEN 1720                   [1230]
1710 dar(0)=dar(0)+1:dar(m)=1                [765]
1720 NEXT m:RETURN                           [461]
1730 t=sp(n,m)+w-l(n)                        [674]
1740 IF t>4 THEN 1720                        [748]
1750 FOR z=1 TO t                            [883]
1760 IF hei(n,z)=1 THEN 1720                 [325]
1770 NEXT z:GOTO 1710                        [831]
1780 t=-sp(n,m)                              [930]
1790 IF t+w>4 THEN 1720                      [262]
1800 FOR z=1 TO w                            [880]
1810 t=t+1:IF hei(n,t)=1 THEN 1720           [1319]
1820 NEXT z:GOTO 1710                        [831]
1830 'Wuerfeln                               [420]
1840 FOR z=INT(RND*15)+5 TO 1 STEP -1        [1355]
1850 w=INT(RND*6)+1:LOCATE 17,6:PRINT w$(w [3330]
):SOUND 1,170-5*z,5
1860 FOR t=1 TO 200-9*z:NEXT t,z             [1427]
1870 FOR t=1 TO 800:NEXT:RETURN              [1550]
1880 'Pflichtzuege                           [1524]
1890 FOR t=1 TO 4:IF mus(t)=1 THEN m=t:GOT [3549]
O 1900 ELSE NEXT t
1900 IF sp(n,m)>0 THEN |PRINT,p(sp(n,m),1) [4782]
,p(sp(n,m),2),41000
1910 IF sp(n,m)<0 THEN |PRINT,hm(n,ABS(sp( [4599]
n,m)),1),hm(n,ABS(sp(n,m)),2),41000+n*36
1920 FOR t=1 TO 200:NEXT t                   [1264]
1930 IF sp(n,m)>0 THEN |PRINT,p(sp(n,m),1) [5053]
,p(sp(n,m),2),41144+n*36
1940 IF sp(n,m)<0 THEN |PRINT,hm(n,ABS(sp( [4599]
n,m)),1),hm(n,ABS(sp(n,m)),2),41288+n*36
1950 FOR t=1 TO 200:NEXT t                   [1264]
1960 a$=INKEY$                               [278]
1970 IF a$=CHR$(13) THEN stein=m:CLS#1:RET [1747]
URN
1980 IF a$<>CHR$(32) THEN 1900               [1593]
1990 m=m+1:IF m>4 THEN m=1                   [1166]
2000 IF mus(m)<>1 THEN 1990                  [1018]
2010 GOTO 1900                               [383]
2020 'Zugauswahl                             [1458]
2030 FOR t=1 TO 4:IF dar(t)=1 THEN m=t:GOT [2753]
O 2040 ELSE NEXT t
```

```
2040 IF sp(n,m)>0 THEN |PRINT,p(sp(n,m),1) [4782]
,p(sp(n,m),2),41000
2050 IF sp(n,m)<0 THEN |PRINT,hm(n,ABS(sp( [4599]
n,m)),1),hm(n,ABS(sp(n,m)),2),41000+n*36
2060 FOR t=1 TO 200:NEXT                   [1050]
2070 IF sp(n,m)>0 THEN |PRINT,p(sp(n,m),1) [5053]
,p(sp(n,m),2),41144+n*36
2080 IF sp(n,m)<0 THEN |PRINT,hm(n,ABS(sp( [4599]
n,m)),1),hm(n,ABS(sp(n,m)),2),41288+n*36
2090 FOR t=1 TO 200:NEXT                   [1050]
2100 a$=INKEY$                             [278]
2110 IF a$=CHR$(13) THEN stein=m:CLS#1:RET [1747]
URN
2120 IF a$<>CHR$(32) THEN 2040             [1964]
2130 m=m+1:IF m>4 THEN m=1                 [1166]
2140 IF dar(m)<>1 THEN 2130                [622]
2150 GOTO 2040                             [355]
2160 'Rausschmeissen                       [959]
2170 FOR m=1 TO 10                         [582]
2180 SOUND 4,10*m+80,10                    [1322]
2190 |PRINT,p(t,1),p(t,2),41144+pin(t)*36  [2537]
2200 FOR z=1 TO 100:NEXT                   [1697]
2210 |PRINT,p(t,1),p(t,2),41144+n*36       [2098]
2220 FOR z=1 TO 100:NEXT z,m               [2339]
2230 FOR m=1 TO 4:IF sp(pin(t),m)=t THEN 2 [4119]
240 ELSE NEXT m
2240 |PRINT,h(pin(t),m,1),h(pin(t),m,2),41 [3345]
288+pin(t)*36
2250 sza(pin(t))=sza(pin(t))+1:szb(pin(t)) [3594]
=szb(pin(t))-1
2260 sp(pin(t),m)=0:RETURN                 [1397]
2270 'ins Haus gehen                       [203]
2280 t=sp(n,stein):z=t+w-l(n):pin(t)=0     [1373]
2290 FOR m=1 TO w-z                        [1033]
2300 IF pin(t)=0 THEN |PRINT,p(t,1),p(t,2) [4777]
,41000 ELSE |PRINT,p(t,1),p(t,2),41144+pin
(t)*36
2310 t=t+1                                 [254]
2320 |PRINT,p(t,1),p(t,2),41144+n*36:SOUND [3301]
 2,200,20
2330 FOR x=1 TO 500:NEXT x,m               [1217]
2340 IF pin(t)=0 THEN |PRINT,p(t,1),p(t,2) [4777]
,41000 ELSE |PRINT,p(t,1),p(t,2),41144+pin
(t)*36
2350 |PRINT,hm(n,1,1),hm(n,1,2),41288+n*36 [4672]
:SOUND 2,200,20:FOR x=1 TO 500:NEXT
2360 z=z-1:FOR m=1 TO z                    [911]
2370 |PRINT,hm(n,m,1),hm(n,m,2),41000+n*36 [3091]
2380 |PRINT,hm(n,m+1,1),hm(n,m+1,2),41288+ [4329]
n*36:SOUND 2,200,20
2390 FOR x=1 TO 500:NEXT x,m               [1217]
2400 hei(n,0)=hei(n,0)+1:hei(n,m)=1:sp(n,s [4955]
tein)=-m:IF hei(n,0)=4 THEN zah=zah+1
2410 IF 5-hei(n,0)=-sp(n,stein) THEN szb(n [2258]
)=szb(n)-1
2420 GOTO 1420                             [399]
2430 'im Haus setzen                       [1369]
2440 t=-sp(n,stein):hei(n,t)=0             [1052]
2450 FOR m=1 TO w                          [797]
2460 |PRINT,hm(n,t,1),hm(n,t,2),41000+n*36 [2961]
2470 t=t+1                                 [254]
2480 |PRINT,hm(n,t,1),hm(n,t,2),41288+n*36 [3925]
:SOUND 2,200,20
2490 FOR x=1 TO 500:NEXT x,m               [1217]
2500 hei(n,t)=1:sp(n,stein)=-t             [2055]
2510 IF 5-hei(n,0)=-sp(n,stein) THEN szb(n [2258]
)=szb(n)-1
2520 GOTO 1420                             [399]
2530 'Spielende                            [324]
2540 LOCATE 17,6:PRINT w$(0):CLS#2:LOCATE# [4764]
2,2,1:PEN#2,13:PRINT#2,"S P I E L E N D E"
2550 LOCATE#1,2,2:PRINT#1,"neues":LOCATE#1 [5478]
,2,4:PRINT#1,"Spiel ?":LOCATE#1,2,6:PRINT#
1,"(J/N)"
2560 IF UPPER$(INKEY$)="J" THEN 630        [2190]
2570 IF UPPER$(INKEY$)="N" THEN MODE 2:LIS [2343]
T ELSE 2560
2580 'Computerzug                          [1507]
2590 IF mus(0)<2 THEN 2620                 [988]
2600 t=INT(RND*4)+1                        [840]
2610 IF mus(t)=1 THEN stein=t:RETURN ELSE  [2703]
2600
2620 FOR m=1 TO 4                          [564]
2630 IF dar(m)=1 AND sp(n,m)>0 AND sp(n,m) [6652]
<=l(n) AND sp(n,m)+w>l(n) THEN stein=m:RET
URN
2640 NEXT m                                [371]
2650 FOR m=1 TO 4                          [564]
2660 IF dar(m)=1 AND sp(n,m)<0 THEN stein= [3140]
m:RETURN
2670 NEXT m                                [371]
2680 FOR m=1 TO 4                          [564]
2690 IF dar(m)=1 AND INT((sp(n,m)+9)/10)=( [6676]
sp(n,m)+9)/10 AND nam$(INT((sp(n,m)+9)/10)
)<>"" AND sza(INT((sp(n,m)+9)/10))>0 THEN
stein=m:RETURN
2700 NEXT m                                [371]
2710 FOR m=0 TO 4:hil(m)=0:NEXT            [1792]
2720 FOR m=1 TO 4                          [564]
2730 IF dar(m)=1 THEN hil(m)=sp(n,m):IF hi [4125]
l(m)<l(n) THEN hil(m)=hil(m)+40
2740 NEXT                                  [350]
2750 hil(0)=MAX(hil(1),hil(2),hil(3),hil(4 [2247]
))
2760 m=1                                   [184]
2770 IF hil(0)=hil(m) THEN 2780 ELSE m=m+1 [1038]
:GOTO 2770
2780 stein=m:RETURN                        [1360]
```






# SCHLUMPIS HORROR-TRIP

Dichter Nebel in den Gassen, hin und wieder hört man entfernte Schritte. Die Luft ist erfüllt von dem Fischgeruch des nahen Hafens. Ratternd läßt gerade der Krämer von gegenüber das Ladengitter ins Schloß fallen. Wer ist in dieser unfreundlichen Nacht noch unterwegs ...? Na wer schon? Schlumpi der stadtbekannte Penner natürlich. Voll bis obenhin torkelt er grölend durch die Gassen auf der Suche nach einem Schlafplatz. Ein Kumpel von ihm hat von einem unbewohnten Haus in einem abgelegenen Stadtteil gesprochen, das einen geräumigen Keller haben soll. Dieses Haus sucht Schlumpi nun. Den Kragen hochgestellt und eine Flasche Fusel in der Rechten versucht Schlumpi die Kälte zu vertreiben, denn er hat noch einen langen Weg vor sich. Ein vorbeikommender Passant hätte diese zwielichtige Gestalt mit dem abgetragenen Mantel vielleicht für eine Ausgeburt eines schlechten Romanes gehalten, aber Schlumpi ist nicht auf Eleganz bedacht und auch sonst nur dem Fusel zugetan. Der Fuselbestand dezimierte sich von Straßenecke zu Straßenecke, bis schließlich nur noch vier Flaschen übrigblieben den Durst zu stillen. Vollkommen genug, mag der geneigte Leser vielleicht denken, nicht so Schlumpi, das jähe Ende vor Augen, beschleunigte er seinen Schritt und gelangte somit auch bald zu seinem ersehnten Ziel. Die Tür zum Keller mit Elan aufstoßend, ein altes Seemannslied auf den Lippen betrat (fiel) unser mutiger Held in den Keller. Dichte Finsternis umgab ihn, und er wollte schon wieder gehen, doch mußte er zu seinem Leidwesen bemerken, daß die Tür ins Schloß gefallen und von innen nicht mehr zu öffnen war. Sich der im Dunkeln lauernden Gefahr stellend, torkelte Schlumpi schließlich weiter in den Keller und suchte einen Lichtschalter. Diesen hatte er auch bald gefunden und entsprechend der komplizierten Anleitung umgelegt. Doch was mußten seine vom Alkohol getrübten Augen jetzt bemerken. Eine wildgewordene Billardkugel und ein Höhrerklapperndes Telefon bewegten sich wie von Geisterhand in diesem mysteriösen Bau umher ... Hier setzt der Spieler ein. Er übernimmt Schlumpis Part. Die Statusanzeige in der ersten Zeile gibt den laufenden Fuselbestand (LIQ.:) und die aktuelle Bild- nummer (SCREEN:) an. Schlumpi wird mit dem Joystick bewegt, dabei gelten folgende Bewegungen:

rechts.....Schlumpi rechts
links.....Schlumpi links
Feuer & rechts.....Sprung nach rechts
Feuer & links.....Sprung nach links

Der Spieler muß nun Schlumpi über die Plattformen zu dem großen, goldenen Schlüssel steuern, um die Tür zum nächsten Raum zu öffnen. Die Schlüsselaufnahme quittiert das Programm mit einem Ton. Nur wenn der Schlüssel vorher eingesammelt wurde, kann Schlumpi zur Ausgangstür gehen und hoffen, daß sie den Weg zum folgenden Raum freigibt!!! Eine Berührung mit der Billardkugel oder mit dem Telefon zieht unserem Held kostbaren Fusel ab. Sie sollten sich auch vor den gelbgezackten Feldern in acht nehmen, da eine Berührung mit ihnen ebenfalls eine Flasche Fusel kostet

## Eigene Bilder

»SCHLUMPIS HORROR-TRIP« ist typisches Plattform-Spiel. Es traten insofern Schwierigkeiten auf, daß der Programmablauf in BASIC wesentlich langsamer ist und die Anzahl der Gegner somit auf zwei beschränkt werden mußte. Die Ablegung der einzelnen Räume im Programm verbrauchte bei meinen ersten Versionen viel Platz, bis ich schließlich auf Datazeilen gekommen bin. Somit ist es auch möglich eigene Bilder anzuhängen oder vorhandene zu verändern.

Die Zahlenbelegungen der Objekte:

1.....Mauermittelstück
2.....Mauerende
3.....Maueranfang
6.....Scherbenhaufen
8.....einzelnes Mauerstück

Bei dem Erstellen neuer Bilder müssen Sie darauf achten, daß das Programm die Bilder in Dreierschritten aufbaut, d.h. daß nur jeweils die dritte Zeile Objekte enthalten darf. Die erste Zeile ist für die Statusanzeigen reserviert! Geben Sie nach der letzten Datazeile »DATA 999« ein, um dem Programm mitzuteilen, daß das Bild zu Ende ist. Jetzt werden die Bahnen der Gegner, die Position der Türen, die des Schlüssels und die Schlumpis in Datazeilen abgelegt. Beispiel: DATA 1,24,19,24,16,2,16,21,1,2,20,2 3,20,3,2,2. Das obige Beispiel läßt die Kugel in Zeile 24 von Spalte 1 bis Spalte 19 hin- und zurückrollen. Das Telefon fliegt in Spalte 16 von Zeile 2 bis Zeile 21 und zurück. Der obere Teil der Eingangstür befindet sich in Spalte 1 und Zeile 2. Die Ausgangstür ist in Spalte 20 und Zeile 23 und der Schlüssel ist in Spalte 20 und Zeile 3. Schlumpis Startposition ist in Spalte 2 und Zeile 2. Bei dem Ablegen der Türen und Schlumpis Positon muß darauf geachtet werden, daß nur der obere Teil angegeben wird! Nur das Telefon kann hoch und runter fliegen! Jetzt müssen Sie nur noch unter der Routine BILDSCHIRMAUFBAU die Zeile des neuen Bildes und unter der Routine BILD GESCHAFFT die Nummer des letzten Bildes angeben...

Abschließend bleibt mir nur noch, Ihnen viel Vergnügen beim Spielen und gute Ideen bei dem Kreieren neuer Bilder zu wünschen.

(Marc Kühl)

**für 464-664-6128**

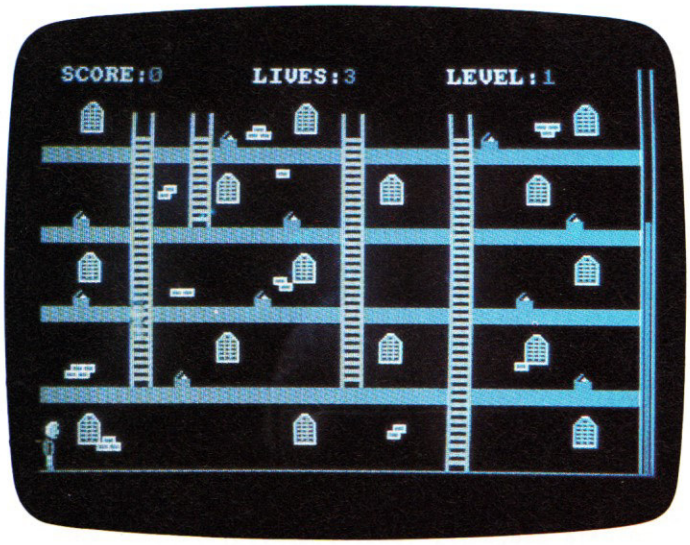
```
10 '*****************************          [1051]
20 '*                           *          [175]
30 '*  SCHLUMPIS HORROR-TRIP    *          [1745]
40 '*                           *          [175]
50 '*    written  2/87 by       *          [926]
60 '*                           *          [175]
70 '*       Marc Kuehl          *          [1086]
80 '*                           *          [175]
90 '*****************************          [1051]
```

```
100 SYMBOL AFTER 164                       [1452]
110 ' Maennchen links                      [1355]
120 SYMBOL 165,24,124,60,60,24,36,78,94    [1352]
130 SYMBOL 166,126,60,36,44,52,58,30,62    [2423]
140 SYMBOL 167,24,124,60,60,24,36,36,36    [1996]
150 SYMBOL 168,52,60,36,84,90,101,99,231   [2193]

160 ' Maennchen rechts                     [1072]
170 SYMBOL 169,24,62,60,60,24,36,114,122   [2823]
180 SYMBOL 170,126,60,36,52,44,92,120,124  [2738]
190 SYMBOL 171,24,62,60,60,24,36,36,36     [1974]
200 SYMBOL 172,44,60,36,42,90,126,198,231  [2185]
210 'Mauer                                 [1048]
220 SYMBOL 173,119,119,0,221,221           [1242]
```

```
230 SYMBOL 174,118,119,1,223,222                [1784]
240 SYMBOL 175,119,247,128,221,93               [1480]
250 'Kugel                                      [80]
260 SYMBOL 176,24,126,255,226,226,255,126,      [2242]
24
270 SYMBOL 177,24,126,255,201,201,255,126,      [2792]
24
280 SYMBOL 178,24,126,255,147,147,255,126,      [2616]
24
290 SYMBOL 179,24,126,255,39,39,255,126,24      [1926]
300 SYMBOL 180,24,60,126,102,195,195,195,2      [2148]
55
310 SYMBOL 181,207,239,255,255,255,255,255      [2353]
,255
320 SYMBOL 182,146,219,255,255,255              [1896]
330 SYMBOL 183,0,0,0,96,144,159,147,99          [2470]
340 SYMBOL 184,118,247,129,223,94               [2052]
350 SYMBOL 185,0,102,255,231,60,102,231,25      [2269]
5
360 SYMBOL 186,102,255,231,0,60,102,231,25      [2303]
5
370 'Bett                                       [740]
380 SYMBOL 187,0,0,191,192,255,255,255,128      [2155]
390 SYMBOL 188,1,15,177,65,255,255,255,1        [2466]
400 ' Anfangstext                               [1311]
410 MODE 1:INK 0,0:BORDER 0:INK 1,23:INK 2      [3282]
,11:INK 3,6:PEN 1
420 '                                           [117]
430 LOCATE 4,1:PEN 2:PRINT CHR$(24);"           [4399]
 SCHLUMPIS HORROR TRIP       ";CHR$(24);:P
EN 1
440 LOCATE 4,3:PRINT"Sie, Schlumpi , ein s      [4293]
tadtbekannter"
450 LOCATE 4,4:PRINT"Penner ziehen auf der      [4620]
  Suche  nach"
460 LOCATE 4,5:PRINT"einem  Schlafplatz         [5442]
durch   einen"
470 LOCATE 4,6:PRINT"geraeumigen  Keller.       [3993]
Doch wie  das"
480 LOCATE 4,7:PRINT"Leben so spielt  wird      [3587]
  eben dies'er"
490 LOCATE 4,8:PRINT"Keller von  einigen        [5998]
zwielichtigen"
500 LOCATE 4,9:PRINT"Gestalten  heimgesuch      [4844]
t.  Zu  allem"
510 LOCATE 4,10:PRINT"Ungluecк  laesst sic      [3240]
h  die Aussen-"
520 LOCATE 4,11:PRINT"tuer auch  nicht  me      [3426]
hr oeffnen ..."
530 LOCATE 4,13:PRINT"Sie sollen nun Schlu      [3358]
mpi durch alle"
540 LOCATE 4,14:PRINT"5  Raeume sicher zu       [4318]
seinem Schlaf-"
550 LOCATE 4,15:PRINT"platz bringen."           [2475]
560 LOCATE 4,16:PRINT"Dazu benoetigen Sie       [3450]
den goldenen!?"
570 LOCATE 4,17:PRINT"Schluessel , der jew      [3264]
eils  die Tuer"
580 LOCATE 4,18:PRINT"zum  naechsten  Raum      [4591]
  oeffnet  ..."
590 LOCATE 4,19:PRINT"Aber  Vorsicht , Sch      [4900]
lumpi verliert"
600 LOCATE 4,20:PRINT"bei jeder  Beruehrun      [3538]
g mit  den ge-"
610 LOCATE 4,21:PRINT"zackten Feldern eine      [4065]
 Flasche Fusel."
620 LOCATE 4,22:PRINT"           Viel Gluec     [1933]
k"
630 PEN 3:LOCATE 4,25:PRINT">>>>>>>>>>>>>       [2411]
SPACE  <<<<<<<<<<<<<"
640 ant$=""                                     [181]
650 WHILE ant$="":ant$=INKEY$:WEND              [1782]
660 IF ant$=" " THEN 670 ELSE ant$="":GOTO      [2565]
 650
670 fusel=4                                     [378]
680 DIM feld(21,26):bild=4                      [583]
690 ENV 1,127,-1,10                             [477]
700 MODE 0:INK 0,0:BORDER 0:INK 1,23:INK 2      [4799]
,11:INK 3,6:INK 4,11:INK 5,24:INK 6,3
710 i=165:gegi=176:rausch=4:schluessel=0        [3610]

720 GOSUB 1880                                  [867]
730 xgeg=vonx:go=1:ygeg=vony:go=0               [2338]
740 xxgeg=vonxx:yygeg=vonyy                     [2800]
750 LOCATE 1,1:PEN 1:PRINT"LIQ.:";fusel-1       [2119]
760 LOCATE 11,1:PRINT"SCREEN:";bild             [1281]
770 LOCATE x,y:PRINT CHR$(i);                   [1451]
780 LOCATE x,y+1:PRINT CHR$(i+1);               [1015]
790 ON JOY(0) GOSUB 1870,1870,1870,1150,18      [5751]
70,1870,1870,1260,1870,1870,1870,1870,1870
,1870,1870,1870,1870,1870,1870,1370,1870,1
870,1870,1620
800 IF geh=4 THEN IF i<>165 THEN i=167 ELS      [1197]
E i=165
810 IF geh=8 THEN IF i<>169 THEN i=171 ELS      [976]
E i=169
820 IF feld(x,y+2)<>-1 AND feld(x,y+2)<>-5      [9096]
 THEN IF y<>25 THEN LOCATE x,y:PRINT " ";:
LOCATE x,y+1:PRINT " ";:y=y+1:LOCATE x,y:P
RINT CHR$(i);:LOCATE x,y+1:PRINT CHR$(i+1)
;
830 IF feld(x,y+2)=-3 OR feld(x,y+1)=-3 TH      [2481]
EN 2800
840 IF x=xgeg AND y=ygeg THEN 2800              [2267]
850 IF x=xgeg AND y+1=ygeg THEN 2800            [1958]
860 IF x=xxgeg AND y+1=yygeg THEN 2800          [2687]
870 IF x=xxgeg AND y=yygeg THEN 2800            [2344]
880 IF feld(x,y+1)=-4 THEN ton=1:schluesse      [2777]
l=1
890 IF ton=1 THEN ENT -1,10,1,1,10,-1,1:EN      [6001]
V 1,127,-1,15:SOUND 1,70,180,15,1,1:ton=0:
feld(x,y+1)=0
900 IF feld(x+1,y)=-5 AND schluessel=1 THE      [2466]
N 2890
910 IF gegi=176 THEN geggo=1                    [876]
920 IF gegi=179 THEN geggo=-1                   [2219]
930 IF gegii=185 THEN gegii=186 ELSE gegii      [4307]
=185
940 IF geggo=1 THEN gegi=gegi+1                 [1698]
950 IF geggo=-1 THEN gegi=gegi-1                [1084]
960 IF xgeg=vonx THEN xgo=1 ELSE IF xgeg=b      [3781]
isx THEN xgo=-1
970 IF ygeg=vony THEN ygo=1 ELSE IF ygeg=b      [4104]
isy THEN ygo=1
980 IF vony=bisy THEN ygo=0                     [1255]
990 IF vonx=bisx THEN xgo=0                     [1277]
1000 IF xgo=-1 THEN LOCATE XGEG,YGEG:PRINT      [2920]
 " ";:XGEG=XGEG-1
1010 IF xgo=1 THEN LOCATE XGEG,YGEG:PRINT       [3583]
" ";:XGEG=XGEG+1
1020 IF ygo=-1 THEN LOCATE XGEG,YGEG:PRINT      [3477]
 " ";:yGEG=yGEG-1
1030 IF ygo=1 THEN LOCATE XGEG,YGEG:PRINT       [3877]
" ";:yGEG=yGEG+1
1040 LOCATE xgeg,ygeg:PEN 4:PRINT CHR$(geg      [3318]
i);:PEN 1
1050 IF xxgeg=vonxx THEN xxgo=1 ELSE IF xx      [4359]
geg=bisxx THEN xxgo=-1
1060 IF yygeg=vonyy THEN yygo=1 ELSE IF yy      [4330]
geg=bisyy THEN yygo=-1
1070 IF vonyy=bisyy THEN yygo=0                 [1571]
1080 IF vonxx=bisxx THEN xxgo=0                 [2227]
1090 IF xxgo=-1 THEN LOCATE xxgeg,yygeg:PR      [3213]
INT " ";:xxgeg=xxgeg-1
1100 IF xxgo=1 THEN LOCATE xxgeg,yygeg:PRI      [3420]
NT " ";:xxgeg=xxgeg+1
1110 IF yygo=-1 THEN LOCATE xxgeg,yygeg:PR      [5096]
INT " ";:yygeg=yygeg-1
1120 IF yygo=1 THEN LOCATE xxgeg,yygeg:PRI      [3527]
NT " ";:yygeg=yygeg+1
1130 LOCATE xxgeg,yygeg:PEN 9:PRINT CHR$(g      [3548]
egii);:PEN 1
1140 GOTO 790                                   [415]
1150 'links                                     [1103]
1160 PEN 1                                      [549]
1170 geh=4                                      [188]
1180 IF feld(x-1,y+1)=-4 THEN 1200              [819]
1190 IF x=1 OR feld(x-1,y)<>0 OR feld(x-1,      [3175]
y+1)<>0 OR feld(x,y+2)=0 THEN RETURN
1200 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR      [2212]
INT " ";
1210 x=x-1:IF i=165 THEN i=167 ELSE i=165       [1758]

1220 IF rausch=4 THEN rausch=1 ELSE rausch      [1627]
=4
1230 SOUND 4,3822,5,15,1,1,rausch               [2272]
1240 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+      [2922]
1:PRINT CHR$(i+1);
1250 RETURN                                     [555]
1260 'rechts                                    [723]
1270 PEN 1                                      [549]
1280 geh=8                                      [200]
1290 IF feld(x+1,y+1)=-4 THEN 1310              [1675]
1300 IF x=20 OR feld(x+1,y)<>0 OR feld(x+1      [4374]
```

```
,y+1)<>0 OR feld(x,y+2)=0 THEN RETURN
1310 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1320 x=x+1:IF i=169 THEN i=171 ELSE i=169  [2362]
1330 IF rausch=4 THEN rausch=1 ELSE rausch [1627]
=4
1340 SOUND 4,3822,5,15,1,1,rausch          [2272]
1350 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1360 RETURN                                [555]
1370 'hoch+links                           [911]
1380 PEN 1                                 [549]
1390 geh=4:i=165                           [234]
1400 IF y=1 OR feld(x,y-1)<>0 OR FELD(X,Y+ [2920]
2)=0 THEN RETURN
1410 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1420 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1430 GOSUB 2830                            [909]
1440 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1450 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1460 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1470 GOSUB 2830                            [909]
1480 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1490 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1500 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1510 GOSUB 2830                            [909]
1520 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1530 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1540 IF x<>1 AND FELD(x-1,y)<>-1 THEN x=x- [2808]
1
1550 GOSUB 2830                            [909]
1560 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1570 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1580 IF x<>1 AND FELD(x-1,y)<>-1 THEN x=x- [2808]
1
1590 GOSUB 2830                            [909]
1600 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1610 RETURN                                [555]
1620 'hoch+rechts                          [598]
1630 PEN 1                                 [549]
1640 geh=8:i=169                           [815]
1650 IF y=1 OR feld(x,y-1)<>0 OR FELD(X,Y+ [2920]
2)=0 THEN RETURN
1660 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1670 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1680 GOSUB 2830                            [909]
1690 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1700 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1710 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1720 GOSUB 2830                            [909]
1730 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1740 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1750 IF FELD(x,y-1)<>-1 AND y<>2 THEN y=y- [2141]
1
1760 GOSUB 2830                            [909]
1770 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1780 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1790 IF x<>20 AND FELD(x+1,y)<>-1 THEN x=x [2709]
+1
1800 GOSUB 2830                            [909]
1810 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1820 LOCATE x,y:PRINT " ";:LOCATE x,y+1:PR [2212]
INT " ";
1830 IF x<>20 AND FELD(x+1,y)<>-1 THEN x=x [2709]
+1
1840 GOSUB 2830                            [909]
1850 LOCATE x,y:PRINT CHR$(i);:LOCATE x,y+ [2922]
1:PRINT CHR$(i+1);
1860 RETURN                                [555]
1870 RETURN                                [555]
1880 'Bildschirmaufbau                     [1166]
1890 IF bild=1 THEN RESTORE 2210           [657]
1900 IF bild=2 THEN RESTORE 2330           [1026]
1910 IF bild=3 THEN RESTORE 2450           [281]
1920 IF bild=4 THEN RESTORE 2570           [1191]
1930 IF bild=5 THEN RESTORE 2690           [1064]
1940 PEN 3                                 [547]
1950 b=1                                   [315]
1960 FOR a=1 TO 20                         [916]
1970 READ feld                             [1052]
1980 IF feld<>999 THEN 2040                [1020]
1990 IF bild=1 THEN RESTORE 2310:GOTO 2130 [1575]
2000 IF bild=2 THEN RESTORE 2430:GOTO 2130 [2304]

2010 IF bild=3 THEN RESTORE 2550:GOTO 2130 [1237]

2020 IF bild=4 THEN RESTORE 2670:GOTO 2130 [1924]

2030 IF bild=5 THEN RESTORE 2790:GOTO 2130 [1670]

2040 IF feld=1 THEN LOCATE a,b:PEN 3:PRINT [2527]
 CHR$(173);:feld(a,b)=-1
2050 IF feld=2 THEN LOCATE a,b:PEN 3:PRINT [3203]
 CHR$(174);:feld(a,b)=-1
2060 IF feld=3 THEN LOCATE a,b:PEN 3:PRINT [2754]
 CHR$(175);:feld(a,b)=-1
2070 IF feld=6 THEN LOCATE a,b:PEN 5:PRINT [5719]
 CHR$(182);:feld(a,b)=-3
2080 IF feld=7 THEN LOCATE a,b:PEN 6:PRINT [4275]
 CHR$(183);:feld(a,b)=-4
2090 IF feld=8 THEN LOCATE a,b:PEN 3:PRINT [3972]
 CHR$(184);:feld(a,b)=-1
2100 IF feld=0 THEN LOCATE a,b:PRINT " ";  [1991]
2110 NEXT                                  [350]
2120 b=b+3:GOTO 1960                       [739]
2130 '                                     [117]
2140 READ vonx,vony,bisx,bisy:READ vonxx,v [4796]
onyy,bisxx,bisyy
2150 READ xtuer1,ytuer1:PEN 6:feld(xtuer1, [4797]
ytuer1)=-1:LOCATE xtuer1,ytuer1:PRINT CHR$
(180);:LOCATE xtuer1,ytuer1+1:PRINT CHR$(1
81);
2160 READ xtuer2,ytuer2:PEN 6:feld(xtuer2, [5866]
ytuer2)=-5:LOCATE xtuer2,ytuer2:PRINT CHR$
(180);:LOCATE xtuer2,ytuer2+1:PRINT CHR$(1
81);
2170 READ schx,schy:PEN 5:feld(schx,schy)= [5312]
-4:LOCATE schx,schy:PRINT CHR$(183);
2180 READ x,y                              [469]
2190 RETURN                                [555]
2200 ' bild 1                              [891]
2210 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2220 DATA 1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [1600]
3,1,1,1
2230 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0, [2489]
0,0,0,0
2240 DATA 0,0,6,2,0,0,0,0,0,0,0,0,0,0,0,0, [2061]
8,6,6,6
2250 DATA 0,0,0,0,8,0,0,0,0,0,0,0,0,0,8,0, [2182]
0,0,0,0
2260 DATA 0,0,0,0,0,6,2,0,0,0,3,6,1,6,0,0, [2661]
3,1,1,1
2270 DATA 0,0,0,0,0,0,0,6,2,6,0,0,0,0,8,0, [1985]
0,0,0,0
2280 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6, [2146]
0,0,0,0
2290 DATA 1,1,1,1,1,1,6,1,1,1,1,1,1,6,1,1, [2093]
1,1,1,1
2300 DATA 999                              [156]
2310 DATA 7,15,13,15,16,2,16,21,1,2,20,23, [2076]
20,3,2,2
2320 ' bild 2                              [894]
2330 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2340 DATA 0,0,0,0,0,0,0,3,1,1,1,1,1,1,1,1, [2733]
1,2,0,0
2350 DATA 0,3,1,6,1,2,0,0,0,0,0,0,0,0,0,0, [1561]
0,0,0,0
2360 DATA 2,0,0,0,0,0,0,0,0,0,0,0,0,8,0,0, [1468]
0,0,3,1
```

```
2370 DATA 0,0,3,6,0,3,6,1,2,0,3,1,6,6,6,2, [2900]
0,0,0,0
2380 DATA 2,0,0,0,8,0,0,0,0,0,0,0,0,0,0,0, [2309]
0,3,1,1
2390 DATA 0,3,6,2,0,0,0,0,0,0,0,0,0,0,0,8, [1584]
0,0,0,0
2400 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2104]
0,3,6,1
2410 DATA 1,1,6,1,1,6,1,6,1,1,6,1,1,1,1,1, [2741]
1,1,1,1
2420 DATA 999                               [156]
2430 DATA 1,17,5,17,6,12,16,12,1,23,20,8,2 [1793]
,18,2,23
2440 ' bild 3                               [897]
2450 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2460 DATA 0,0,0,0,0,0,0,0,0,8,0,0,0,0,0,0, [1954]
0,0,0,0
2470 DATA 0,0,0,8,0,8,0,8,0,0,3,1,6,2,0,0, [2358]
0,0,0,0
2480 DATA 1,2,0,0,6,0,6,0,6,6,0,0,0,0,0,0, [2011]
0,0,0,0
2490 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0, [2472]
3,6,1,1
2500 DATA 0,0,0,0,0,0,0,0,3,1,1,6,1,1,1,2, [1527]
6,6,0,0
2510 DATA 0,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0, [1847]
0,0,0,0
2520 DATA 0,0,0,0,0,6,1,6,0,0,0,0,0,0,0,0, [2031]
0,0,0,0
2530 DATA 1,1,1,1,1,1,1,1,1,1,1,1,6,1,1,1, [1753]
1,1,1,1
2540 DATA 999                               [156]
2550 DATA 4,6,14,6,9,15,16,15,1,8,20,23,20 [2600]
,12,2,8
2560 ' bild 4                               [884]
2570 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2580 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2590 DATA 0,0,0,0,0,3,1,6,0,3,1,2,0,8,0,6, [2459]
0,0,0,0
2600 DATA 0,0,0,8,0,0,0,6,1,6,0,0,6,0,0,0, [2610]
0,0,0,0
2610 DATA 0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2163]
0,3,1,1
2620 DATA 0,0,0,8,0,0,0,0,0,0,0,0,0,0,3,1, [2198]
6,0,0,0
2630 DATA 0,8,0,0,0,0,0,0,0,3,6,1,6,1,6,1, [1702]
6,1,6,1
2640 DATA 0,0,6,8,0,0,0,0,0,0,0,0,0,0,0,0, [1827]
0,0,0,0
2650 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1, [1690]
1,1,1,1
2660 DATA 999                               [156]
2670 DATA 11,18,14,18,6,6,14,6,1,23,20,11, [2036]
10,18,2,23
2680 ' bild 5                               [887]
2690 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2700 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, [2396]
0,0,0,0
2710 DATA 0,0,0,3,1,2,0,0,0,3,2,0,3,1,6,1, [2071]
6,2,0,3
2720 DATA 0,8,0,0,0,0,3,6,1,6,6,6,0,3,1,2, [1702]
0,0,3,1
2730 DATA 0,0,0,8,0,0,0,0,0,0,3,1,1,6,6,6, [1991]
1,1,6,0
2740 DATA 1,2,0,0,0,0,6,6,3,2,0,0,0,0,0,0, [1994]
0,0,0,6
2750 DATA 0,0,0,0,0,0,0,0,0,0,3,1,6,6,0,0, [2551]
0,0,0,0
2760 DATA 0,0,0,0,0,0,0,6,6,1,1,1,6,6,0,0, [2600]
0,0,0,0
2770 DATA 6,1,6,1,6,1,1,1,1,1,1,1,1,1,1,1, [1998]
1,1,1,1
2780 DATA 999                               [156]
2790 DATA 4,6,20,6,1,24,19,24,1,14,20,23,1 [1869]
1,21,2,14
2800 'In eine Falle getreten                [481]
2810 IF fusel<>1 THEN fusel=fusel-1:SOUND   [3194]
2,3822,100,15,1,1,31:GOTO 700 ELSE 2950

2820 'Ueberpruefen der Position             [1626]
2830 IF feld(x,y+2)=-3 OR feld(x,y+1)=-3 T [2481]
HEN 2800
2840 IF x=xgeg AND y+1=ygeg THEN 2800       [1958]
2850 IF x=xxgeg AND y+1=yygeg THEN 2800     [2687]

2860 IF x=xxgeg AND y=yygeg THEN 2800       [2344]
2870 IF feld(x,y+1)=-4 THEN schluessel=1    [1306]
2880 RETURN                                 [555]
2890 'Bild geschafft                        [1449]
2900 IF bild=5 THEN 3130                    [465]
2910 ENT -3,2,-5,1,2,4,1:SOUND 2,400,150,1 [6638]
5,1,3:SOUND 4,300,150,15,1,3:SOUND 1,200,1
50,15,1,3:bild=bild+1:ii=1
2920 FOR i=1 TO 20:feld(i,ii)=0:NEXT        [1043]
2930 IF ii<>25 THEN ii=ii+1:GOTO 2920       [1706]
2940 schluessel=0:GOTO 700                  [1234]
2950 ' Fusel alle !!!                       [1207]
2960 RESTORE 3020                           [747]
2970 ENT -1,10,1,10                         [867]
2980 READ ton,dauer                         [1018]
2990 IF ton=999 OR dauer=999 THEN 3050      [1813]
3000 SOUND 4,ton*2,dauer*2,15,,1:SOUND 2,t [4036]
on,dauer*2,15,,1
3010 GOTO 2980                              [311]
3020 DATA 358,50,0,1,358,25,358,13,0,1,358 [2353]
,13,0,1,358,50,301,25,301,13,319,13
3030 DATA 0,1,319,25,319,13,358,13,0,1,358 [2998]
,25,358,13,379,13,358,50
3040 DATA 999,999                           [203]
3050 MODE 1:PEN 1                           [1036]
3060 PEN 1:LOCATE 7,8:PRINT"Pleite auf der [3888]
 ganzen Linie!"
3070 LOCATE 5,9:PRINT"Alle 3 Flaschen Fuse [3498]
l sind weg!"
3080 ant$="":PEN 3:LOCATE 12,12:PRINT"Noch [3067]
 einmal (j/n)?"
3090 ant$=INKEY$:IF ant$<>"" THEN 3100 ELS [2030]
E 3090
3100 IF UPPER$(ant$)="J" THEN CLEAR:GOTO 1 [3092]
00
3110 IF UPPER$(ant$)="N" THEN CLS:END       [1803]

3120 GOTO 3090                              [630]
3130 ' Geschafft !!!                        [1643]
3140 RESTORE 3200                           [743]
3150 READ ton,dauer                         [1018]
3160 IF ton=999 THEN FOR a=1 TO 1500:NEXT: [3683]
MODE 0:GOTO 3220
3170 SOUND 2,ton*2,dauer,15,1,1:SOUND 4,to [3374]
n,dauer,15,1,1
3180 GOTO 3150                              [451]
3190 '                                      [117]
3200 DATA 358,60,268,45,0,2,268,15,0,2,268 [7404]
,120,358,60,239,45,284,15,268,120,358,60,2
68,45,201,15,0,2,201,60,213,45,239,15,268,
60,239,2,268,2,284,45,268,15,239,120
3210 DATA 999,999                           [203]
3220 PEN 3:LOCATE 1,25:PRINT STRING$(20,CH [2168]
R$(173));
3230 PEN 5:LOCATE 10,24:PRINT CHR$(187);CH [3369]
R$(188);
3240 i=169                                  [505]
3250 FOR x=1 TO 9                           [705]
3260 FOR a=1 TO 250:NEXT                    [1482]
3270 IF x=1 THEN 3290                       [687]
3280 LOCATE x-1,23:PRINT " ";:LOCATE x-1,2 [1751]
4:PRINT " ";
3290 PEN 1:LOCATE x,23:PRINT CHR$(i);:LOCA [4420]
TE x,24:PRINT CHR$(i+1);
3300 SOUND 4,3822,5,15,1,1,rausch           [2272]
3310 IF rausch=1 THEN rausch=4 ELSE rausch [1971]
=1
3320 IF i=169 THEN i=171 ELSE i=169         [1418]
3330 'LOCATE x,23:PRINT " ";:LOCATE x,24:P [2769]
RINT " ";
3340 NEXT                                   [350]
3350 FOR a=1 TO 1000:NEXT:INK 1,0:INK 3,0: [3401]
INK 5,0
3360 FOR a=1 TO 1000:NEXT:INK 1,23:INK 3,6 [2465]
:INK 5,5
3370 MODE 1:PEN 3:LOCATE 11,12:PRINT"Noch  [3286]
einmal (j/n)?"
3380 ant$=""                                [181]
3390 WHILE ant$="":ant$=INKEY$:WEND         [1782]
3400 IF UPPER$(ant$)="J" THEN CLEAR:GOTO 1 [3092]
00
3410 IF UPPER$(ant$)="N" THEN CLS:END       [1803]
3420 ant$="":GOTO 3390                      [381]
```




# RUNNER

Fritzchen-008 erhält folgenden Auftrag:
»Sammeln Sie die Schätze in der alten Burg!«
Für jeden Schatz, den Fritzchen aufnimmt, werden ihm 10 Punkte gutgeschrieben. Der Schatz ist erst dann aufgenommen, wenn ein Ton zu hören ist. Sind alle Schätze aufgenommen, bekommt Fritzchen nochmals eine bestimmte Punktezahl, die von der Zeit abhängt.
Allerdings kommen ihm beim Aufsammeln Kugeln entgegen, denen er durch Hüpfen ausweichen kann. Mit jeder geschafften Spielrunde wird die Anzahl der Kugeln erhöht.
Eine weitere Schwierigkeit stellt die begrenzte Zeit dar. Sie wird durch einen Balken am rechten Spielfeldrand angezeigt.
Gesteuert wird wahlweise mit Joystick oder Tastatur:

a – nach oben
z – nach unten
; – nach rechts
: – nach links
<SPACE> – Sprung

Während des Spielverlaufs können noch folgende Tasten betätigt werden:

e – zurück zum Menue
<ENTER> – Stoppen des Spiels

Im Programmteil »FARBENEINSTELLUNG« kann mit den Cursortasten <oben> und <unten> oder dem Joystick die Randfarbe verändert werden. Durch Drücken von <COPY> oder <FIRE> werden nacheinander Rand und Farben 1 – 3 festgesetzt.

## Hinweise zum Abtippen

1. Hauptprogramm abtippen und mit »RUN« starten. Kassettenbesitzer müssen jetzt »REC« und »PLAY« drücken und der Objectcode wird automatisch abgespeichert.
2. Anschließend das 2. Programm abtippen und hinter dem Hauptprogramm wie unter 1. abspeichern. Die Datei muß unbedingt den Namen »CODE.BIN« tragen, da sie sonst nicht nachgeladen werden kann.
3. Nun noch zurückspulen und mit 'RUN "RUNNER' starten.

(S. & D. Gradl)

## für 464-664-6128

```
10  REM"******************************"     [1771]
20  REM"***  -->  R U N N E R  <--  ***"     [1761]
30  REM"******************************"     [1771]
40  REM"***                        ***"     [1117]
50  REM"***  written  &  designed  ***"     [2799]
60  REM"***          by...         ***"     [1271]
70  REM"***      Dietmar & Sven    ***"     [2384]
80  REM"***        G R A D L       ***"     [1177]
90  REM"***------------------------***"     [1535]
100 REM"***  (c) Februar 1987      ***"     [1819]
110 REM"******************************"     [1771]
120 FOR x=&8500 TO &96A8 STEP 20             [1795]
130 : sum=0                                  [680]
140 : FOR y=0 TO 19                          [1055]
150 : : READ a$:zahl=VAL("&"+a$)             [2454]
160 : : sum=sum+zahl                         [694]
170 : POKE x+y,zahl                          [1094]
180 : NEXT y                                 [431]
190 : READ chksum$                           [656]
200 : chksum=VAL("&"+chksum$)                [1115]
210 : zeile=(x-&8500)/2+250                  [1967]
220 : IF chksum=sum THEN PRINT"Zeile";zeil   [4655]
e;"OK.":GOTO 240
230 : PRINT"Fehler in Zeile:";zeile;"."      [4260]
240 NEXT x                                   [356]
245 SAVE"runner.bin",b,&8500,&11B2,&8500     [2362]
250 DATA 3E,FF,32,00,B8,21,0F,0F,22,E9,B4,   [4612]
3A,88,A6,FE,00,28,0B,21,82,0761
260 DATA 95,11,77,BC,01,0F,00,ED,B0,21,00,   [3068]
00,22,0E,9F,06,08,21,91,95,05CB
270 DATA 11,00,9C,CD,77,BC,21,00,A0,CD,83,   [2884]
BC,CD,7A,BC,97,CD,96,BB,3E,0A70
280 DATA 1A,32,01,75,3E,14,32,02,75,3E,0D,   [3489]
32,03,75,3E,01,21,72,95,CD,04E6
290 DATA BF,BC,3E,01,21,76,95,CD,BC,BC,97,   [2968]
32,10,76,3E,01,32,00,76,C3,0824
300 DATA 7B,92,3E,01,32,32,9F,3E,03,32,25,   [3435]
9F,21,00,00,22,50,9F,31,00,04E9
310 DATA C0,CD,5C,96,3E,01,CD,0E,BC,3E,01,   [4335]
CD,DE,BB,11,72,02,21,00,00,07A0
320 DATA E5,CD,EA,BB,21,90,01,11,00,00,CD,   [3076]
F9,BB,11,7F,02,E1,CD,EA,BB,0A80
330 DATA 11,00,00,21,90,01,CD,F9,BB,3E,03,   [3088]
CD,DE,BB,06,03,11,76,02,C5,0742
340 DATA D5,21,00,00,CD,EA,BB,21,80,02,11,   [3784]
00,00,CD,F9,BB,E1,23,23,54,0818
350 DATA 5D,C1,10,E7,3E,03,CD,DE,BB,21,00,   [4040]
00,11,00,00,E5,CD,EA,BB,11,0856
360 DATA 80,02,E1,CD,F9,BB,06,10,21,39,86,   [3750]
C5,7E,32,03,9F,23,7E,32,04,07C8
370 DATA 9F,23,7E,32,05,9F,23,7E,32,06,9F,   [3875]
97,32,07,9F,32,08,9F,23,7E,0677
380 DATA 32,09,9F,23,7E,32,0A,9F,23,22,0C,   [3734]
9F,2A,03,9F,ED,5B,05,9F,06,0604
390 DATA 08,D5,C5,E5,CD,EA,BB,2A,07,9F,ED,   [3249]
5B,09,9F,CD,F9,BB,E1,23,23,0B61
400 DATA C1,D1,10,E9,2A,0C,9F,C1,10,B1,C3,   [3093]
93,86,42,00,00,00,AE,01,42,07F1
410 DATA 00,C8,01,A8,00,92,00,00,00,5E,00,   [3525]
92,00,78,00,C6,00,92,00,58,051B
420 DATA 01,56,00,92,00,C8,01,A8,00,E2,00,   [3204]
00,00,5E,00,E2,00,78,00,C6,05BA
430 DATA 00,E2,00,58,01,56,00,E2,00,C8,01,   [4007]
A8,00,32,01,00,00,5E,00,32,04A7
440 DATA 01,78,00,26,00,32,01,B8,00,86,00,   [3838]
32,01,C8,01,A8,00,32,01,58,043F
450 DATA 01,56,00,3E,02,CD,DE,BB,06,04,21,   [2979]
50,87,C5,7E,32,10,9F,23,7E,06C4
460 DATA 32,11,9F,23,7E,32,12,9F,23,7E,32,   [2924]
13,9F,23,7E,32,14,9F,23,7E,0612
470 DATA 32,15,9F,23,7E,32,20,9F,23,E5,CD,   [4743]
FD,86,2A,10,9F,23,23,22,10,0721
480 DATA 9F,CD,FD,86,2A,10,9F,11,12,00,19,   [3156]
22,10,9F,CD,FD,86,2A,10,9F,07FE
490 DATA 23,23,22,10,9F,CD,FD,86,2A,10,9F,   [4046]
```

```
11,16,00,ED,52,22,10,9F,CD,0744
500 DATA 15,87,E1,C1,10,A3,C3,6C,87,ED,5B, [3477]
10,9F,2A,12,9F,CD,EA,BB,2A,0A15
510 DATA 14,9F,11,00,00,CD,F9,BB,C9,78,32, [2691]
20,9F,2A,12,9F,23,23,23,23,06DE
520 DATA 23,23,22,12,9F,ED,5B,10,9F,CD,EA, [4499]
BB,21,00,00,11,16,00,CD,F9,0790
530 DATA BB,2A,12,9F,23,23,22,12,9F,ED,5B, [3415]
10,9F,CD,EA,BB,21,00,00,11,074A
540 DATA 16,00,CD,F9,BB,3A,20,9F,47,10,C2, [3120]
C9,B0,01,02,00,60,01,29,60,070F
550 DATA 00,52,00,10,01,1F,A0,00,F2,00,70, [2580]
00,0B,40,01,52,00,10,01,1F,0352
560 DATA 3E,17,32,52,9F,21,1F,88,18,07,2A, [3424]
10,9F,78,32,52,9F,CD,F6,87,071D
570 DATA ED,53,40,9F,ED,43,42,9F,22,10,9F, [4353]
CD,9A,87,2A,10,9F,3A,52,9F,08F3
580 DATA 47,10,DF,C3,7B,88,3E,01,CD,DE,BB, [2954]
ED,5B,40,9F,2A,42,9F,CD,EA,0A8A
590 DATA BB,21,03,88,22,36,9F,06,04,78,32, [4317]
38,9F,CD,DB,87,3E,03,CD,DE,0804
600 DATA BB,2A,40,9F,23,23,EB,2A,42,9F,23, [3238]
23,CD,EA,BB,21,13,88,22,36,07CC
610 DATA 9F,06,03,78,32,38,9F,CD,DB,87,C9, [3528]
18,04,78,32,38,9F,2A,36,9F,07BD
620 DATA CD,F6,87,22,36,9F,60,69,CD,F9,BB, [4261]
3A,38,9F,47,10,E8,C9,7E,5F,0A81
630 DATA 23,7E,57,23,7E,4F,23,7E,47,23,C9, [1724]
0C,00,00,00,00,00,06,00,F4,04C2
640 DATA FF,00,00,00,00,FA,FF,08,00,00,00, [2145]
00,00,02,00,F8,FF,00,00,3C,0535
650 DATA 00,14,00,49,00,14,00,42,00,1B,00, [3548]
72,01,20,00,79,01,27,00,1A,021C
660 DATA 00,5B,00,27,00,5B,00,21,00,61,00, [3015]
2D,00,61,00,F4,01,64,00,8C,03D2
670 DATA 00,AD,00,99,00,AD,00,FA,00,B9,00, [1734]
01,01,B2,00,FD,00,25,01,08,0585
680 DATA 02,54,01,15,02,54,01,0F,02,4E,01, [3788]
DC,00,4A,01,E9,00,4A,01,E3,0461
690 DATA 00,51,01,7E,00,0E,01,85,00,15,01, [4350]
21,A0,89,06,0F,22,86,9F,78,0498
700 DATA 32,88,9F,CD,F6,87,ED,53,70,9F,ED, [4846]
43,72,9F,22,86,9F,CD,A4,88,0B73
710 DATA 2A,86,9F,3A,88,9F,47,10,DF,C3,DC, [3235]
89,3E,01,CD,DE,BB,ED,5B,70,0A6B
720 DATA 9F,2A,72,9F,CD,EA,BB,21,38,89,22, [1812]
36,9F,3E,06,32,38,9F,CD,E4,0923
730 DATA 88,3E,03,CD,DE,BB,2A,70,9F,23,23, [2703]
EB,2A,72,9F,23,23,CD,EA,BB,098C
740 DATA 21,50,89,22,36,9F,3E,06,32,38,9F, [2625]
CD,E4,88,18,1B,18,04,78,32,0670
750 DATA 38,9F,2A,36,9F,CD,F6,87,22,36,9F, [4252]
60,69,CD,F9,BB,3A,38,9F,47,09B9
760 DATA 10,E8,C9,3E,02,CD,DE,BB,21,68,89, [3349]
22,80,9F,3E,07,32,82,9F,18,086A
770 DATA 04,78,32,82,9F,2A,80,9F,CD,F6,87, [3494]
22,80,9F,60,69,CD,ED,BB,2A,0A0B
780 DATA 80,9F,CD,F6,87,22,80,9F,60,69,CD, [4063]
F9,BB,3A,82,9F,47,10,DA,C9,0B49
790 DATA 16,00,00,00,00,00,14,00,F8,FF,08, [3260]
00,FA,FF,00,00,F8,FF,F8,FF,0810
800 DATA 00,00,EC,FF,12,00,00,00,00,00,12, [3258]
00,FA,FF,06,00,FA,FF,00,00,0607
810 DATA FA,FF,FA,FF,00,00,EE,FF,02,00,04, [3350]
00,0E,00,00,00,00,00,04,00,05F7
820 DATA F2,FF,00,00,00,00,04,00,0E,00,00, [3680]
00,00,00,04,00,F2,FF,00,00,03F8
830 DATA 02,00,04,00,0A,00,00,00,FE,FF,02, [2483]
00,00,00,EC,FF,FA,FF,00,00,05F3
840 DATA 00,00,14,00,0E,01,1A,00,30,02,1A, [2305]
00,28,00,1A,00,BE,00,6A,00,01F3
850 DATA 68,01,6A,00,FE,01,6A,00,28,00,BA, [3595]
00,0E,01,BA,00,30,02,BA,00,04D3
860 DATA BE,00,0A,01,68,01,0A,01,FE,01,0A, [2630]
01,28,00,50,01,0E,01,50,01,0320
870 DATA 30,02,50,01,3E,03,CD,DE,BB,21,FF, [3100]
89,06,0A,C5,CD,F6,87,E5,60,0937
880 DATA 69,CD,EA,BB,CD,27,8A,3E,03,CD,DE, [3741]
BB,E1,C1,10,EA,C3,89,8A,91,0C03
890 DATA 00,52,00,22,00,A2,00,22,00,F2,00, [3601]
D0,01,42,01,30,02,52,00,F4,04B6
900 DATA 01,A2,00,05,01,F2,00,C0,00,42,01, [3849]
26,02,F2,00,E6,00,A2,00,21,0561
910 DATA 3D,8A,06,0B,C5,CD,F6,87,E5,60,69, [3813]
CD,F9,BB,E1,C1,10,F2,CD,69,0BF0
920 DATA 8A,10,00,00,00,00,00,02,00,F0,FF, [3347]
00,00,00,00,02,00,10,00,00,029D
930 DATA 00,00,00,02,00,F0,FF,00,00,00,00, [3493]
02,00,06,00,06,00,02,00,00,0201
940 DATA 00,FA,FF,FA,FF,3E,01,CD,DE,BB,21, [4160]
00,00,11,06,00,E5,CD,ED,BB,0A29
950 DATA E1,11,04,00,CD,F9,BB,11,FE,FF,21, [3272]
02,00,CD,F9,BB,C9,3E,01,CD,09FE
960 DATA 90,BB,21,C7,8A,7E,23,CD,5A,BB,B7, [3413]
20,F8,CD,3D,96,3E,03,CD,90,0A4D
970 DATA BB,26,22,2E,01,CD,75,BB,3A,32,9F, [3791]
C6,30,CD,5A,BB,26,15,2E,01,077C
980 DATA CD,75,BB,3A,25,9F,C6,30,CD,5A,BB, [4764]
2A,50,9F,CD,A3,91,18,22,20,0947
990 DATA 53,43,4F,52,45,3A,20,20,20,20,20, [3080]
20,20,4C,49,56,45,53,3A,20,0473
1000 DATA 20,20,20,20,20,20,4C,45,56,45,4C [3886]
,3A,00,21,90,01,22,14,9F,97,0490
1010 DATA CD,DE,BB,3E,02,32,60,9F,32,66,9F [4537]
,32,6C,9F,32,72,9F,97,32,61,08B8
1020 DATA 9F,32,67,9F,32,6D,9F,32,73,9F,3E [3422]
,07,32,62,9F,32,65,9F,3E,11,0756
1030 DATA 32,68,9F,32,6B,9F,3E,1B,32,6E,9F [3489]
,32,71,9F,3E,25,32,74,9F,32,0729
1040 DATA 77,9F,3E,2F,32,7A,9F,3E,4A,32,63 [3790]
,9F,32,69,9F,32,6F,9F,32,75,07AB
1050 DATA 9F,32,78,9F,3E,4C,32,64,9F,32,6A [2930]
,9F,32,70,9F,32,76,9F,32,79,0815
1060 DATA 9F,3E,0A,32,27,9F,21,00,A2,97,06 [3808]
,45,77,23,10,FC,21,90,A2,97,0714
1070 DATA 06,DC,77,23,10,FC,3E,2C,32,01,9F [2923]
,5F,97,32,00,9F,21,00,A0,22,066E
1080 DATA FE,9F,CD,5B,8D,21,5C,AC,11,F6,8E [3452]
,01,00,81,CD,D7,BC,3E,07,CD,0A04
1090 DATA 5A,BB,3E,03,32,57,9F,CD,C9,91,CD [4887]
,09,BB,FE,0B,28,3F,FE,61,28,092D
1100 DATA 3B,FE,0A,28,44,FE,7A,28,40,FE,08 [4468]
,CA,FA,8B,FE,3A,CA,FA,8B,FE,0B69
1110 DATA 09,CA,27,8C,FE,3B,CA,27,8C,FE,20 [3380]
,CA,55,8C,FE,58,CA,55,8C,FE,0B04
1120 DATA 65,20,06,CD,D3,8C,C3,7B,92,FE,45 [2801]
,28,F6,FE,0D,CC,78,96,18,B7,0A9C
1130 DATA CD,F5,8D,CD,DA,8C,D6,02,CD,F4,8C [4081]
,18,AA,CD,44,8E,CD,DA,8C,C6,0D01
1140 DATA 02,CD,F4,8C,18,9D,CD,DF,8D,CD,90 [4872]
,8E,3A,00,9F,A7,CA,97,8B,CD,0B61
1150 DATA 27,8D,D6,02,32,00,9F,CD,56,8D,21 [3499]
,CF,A0,CD,45,8D,21,00,30,CD,085A
1160 DATA A4,96,21,8A,A0,CD,45,8D,C3,97,8B [4557]
,CD,DF,8D,CD,90,8E,3A,00,9F,0B06
1170 DATA FE,4A,CA,97,8B,CD,27,8D,C6,02,32 [3573]
,00,9F,CD,56,8D,21,45,A0,CD,09D1
1180 DATA 45,8D,21,00,30,CD,A4,96,21,00,A0 [4180]
,CD,45,8D,C3,97,8B,3A,01,9F,0849
1190 DATA 5F,3A,00,9F,F5,D5,3A,FE,9F,A7,20 [3668]
,07,3E,01,32,22,9F,18,0D,FE,07FC
1200 DATA 8A,28,05,D1,F1,C3,97,8B,97,32,22 [3613]
,9F,21,00,A2,22,FE,9F,D1,F1,0A2C
1210 DATA F5,D5,CD,5B,8D,D1,7B,D6,02,5F,F1 [3399]
,F5,D5,CD,56,8D,3A,22,9F,A7,0C0F
1220 DATA 20,05,21,CF,A0,18,03,21,45,A0,22 [4327]
,FE,9F,D1,F1,F5,D5,CD,5B,8D,09D6
1230 DATA 21,00,20,2B,E5,CD,09,BB,E1,38,04 [4252]
,7D,B4,20,F4,D1,F1,21,00,A2,08C9
1240 DATA 22,FE,9F,CD,5B,8D,3A,22,9F,A7,3A [3737]
,01,9F,5F,3A,00,9F,CA,0C,8C,088A
1250 DATA C3,3A,8C,21,5C,AC,CD,DD,BC,C9,21 [3039]
,00,A2,22,FE,9F,3A,01,9F,5F,099C
1260 DATA 3A,00,9F,CD,5B,8D,21,00,A2,22,FE [4085]
,9F,3A,01,9F,C9,32,01,9F,5F,07E4
1270 DATA 3A,00,9F,CD,56,8D,3A,03,9F,FE,14 [3350]
```

```
,20,0B,21,59,A1,22,03,9F,22,06A3
1280 DATA FE,9F,18,09,21,14,A1,22,03,9F,22 [2324]
,FE,9F,3A,01,9F,5F,3A,00,9F,0729
1290 DATA CD,5B,8D,CD,CA,8D,C9,21,00,A2,22 [1896]
,FE,9F,3A,01,9F,5F,3A,00,9F,0936
1300 DATA CD,5B,8D,21,00,A2,22,FE,9F,3A,01 [3444]
,9F,5F,3A,00,9F,C9,22,FE,9F,08D1
1310 DATA 3A,01,9F,5F,3A,00,9F,CD,5B,8D,CD [3005]
,CA,8D,C9,01,12,7E,18,03,01,0761
1320 DATA 77,1A,21,76,8D,70,23,71,67,6B,CD [2291]
,88,8D,06,03,0E,17,ED,5B,FE,07E1
1330 DATA 9F,DD,68,DD,45,E5,1A,77,C5,C1,13 [2918]
,CD,BE,8D,10,F6,E1,CD,AC,8D,0C1A
1340 DATA 0D,20,EC,C9,45,CB,85,4C,26,00,54 [4135]
,5D,29,29,19,29,29,29,59,19,05F2
1350 DATA ED,5B,C9,B1,19,7C,E6,07,67,3E,C0 [2233]
,84,CB,18,30,02,C6,20,67,C9,0958
1360 DATA 7C,C6,08,67,E6,38,C0,7C,D6,40,67 [3589]
,7D,C6,50,6F,D0,18,02,2C,C0,0960
1370 DATA 24,7C,E6,07,C0,7C,D6,08,67,C9,21 [1679]
,D3,8D,06,06,CD,FC,91,C9,00,0987
1380 DATA F4,01,02,06,0E,08,07,07,36,07,3F [3626]
,3A,01,9F,FE,04,C8,FE,0E,C8,0615
1390 DATA FE,18,C8,FE,22,C8,FE,2C,C8,E1,C3 [2027]
,97,8B,3A,00,9F,FE,0C,28,0F,0A98
1400 DATA FE,14,28,1E,FE,28,28,07,FE,36,28 [3112]
,29,C3,97,8B,3A,01,9F,FE,23,0812
1410 DATA DA,17,8E,E1,C3,97,8B,FE,05,D0,E1 [3368]
,C3,97,8B,3A,01,9F,FE,0F,DA,0B9F
1420 DATA 2A,8E,E1,C3,97,8B,FE,05,D0,E1,C3 [4124]
,97,8B,3A,01,9F,FE,2D,DA,3D,0B33
1430 DATA 8E,E1,C3,97,8B,FE,05,D0,E1,C3,97 [3012]
,8B,3A,00,9F,FE,0C,28,0F,FE,0B05
1440 DATA 14,28,1D,FE,28,28,07,FE,36,28,27 [4402]
,C3,97,8B,3A,01,9F,FE,22,38,0748
1450 DATA 04,E1,C3,97,8B,FE,04,D0,E1,C3,97 [3129]
,8B,3A,01,9F,FE,0E,38,04,E1,0A65
1460 DATA C3,97,8B,FE,04,D0,E1,C3,97,8B,3A [3033]
,01,9F,FE,2C,38,04,E1,C3,97,0AF8
1470 DATA 8B,FE,04,D0,E1,C3,97,8B,21,40,A2 [3973]
,7E,FE,FF,C0,06,18,21,2D,A2,0A6F
1480 DATA 97,77,23,10,FB,3E,03,CD,90,BB,2A [1861]
,50,9F,11,0A,00,19,22,50,9F,06F3
1490 DATA CD,A3,91,21,7A,95,06,04,CD,FC,91 [3995]
,21,D0,07,2B,7C,B5,20,FB,0E,0912
1500 DATA 3F,3E,07,CD,34,BD,3A,27,9F,3D,32 [3765]
,27,9F,C0,2A,50,9F,ED,5B,14,07AC
1510 DATA 9F,19,22,50,9F,3A,32,9F,3C,FE,0A [3837]
,20,02,3E,09,32,32,9F,E1,3E,06A3
1520 DATA FF,32,10,76,CD,39,91,C3,76,85,C5 [2936]
,D5,E5,F5,CD,98,96,3A,00,76,0B2B
1530 DATA 3D,32,00,76,C2,07,91,3E,07,32,00 [4336]
,76,3A,57,9F,3D,32,57,9F,20,05E1
1540 DATA 08,CD,1C,91,3E,03,32,57,9F,3A,32 [3437]
,9F,3D,32,13,9F,CD,AC,90,22,0742
1550 DATA 16,9F,CD,BA,90,2A,16,9F,7E,23,23 [2546]
,5E,CD,EA,8F,3A,13,9F,A7,28,08CE
1560 DATA 06,3D,32,13,9F,18,E1,CD,BA,90,2A [4330]
,16,9F,7E,5F,23,56,BA,38,05,0763
1570 DATA CA,B6,8F,18,30,A7,CA,D6,90,FE,0C [3659]
,CA,15,90,FE,14,CA,2E,90,FE,0B3F
1580 DATA 28,CA,15,90,FE,36,CA,3F,90,73,2B [2959]
,7B,D6,02,77,23,23,5E,23,22,07B5
1590 DATA 16,9F,CD,E5,8F,CD,0F,91,20,C1,C3 [3106]
,EE,90,FE,4C,CA,D6,90,FE,0C,0C09
1600 DATA CA,15,90,FE,14,CA,2E,90,FE,28,CA [2674]
,15,90,FE,36,CA,3F,90,73,2B,0A09
1610 DATA 7B,C6,02,77,23,23,5E,23,22,16,9F [4707]
,CD,E5,8F,CD,0F,91,20,90,C3,0879
1620 DATA EE,90,FE,0C,CA,15,90,FE,14,CA,2E [2483]
,90,FE,28,CA,15,90,FE,36,CA,0B24
1630 DATA 3F,90,73,2B,56,23,23,7E,C6,02,77 [2477]
,5F,7A,23,22,16,9F,CD,E5,8F,07DA
1640 DATA CD,0F,91,C2,43,8F,C3,EE,90,01,12 [4028]
,7E,18,03,01,77,1A,21,05,90,0736
1650 DATA 70,23,71,67,6B,CD,88,8D,06,02,0E [3445]
,0B,ED,5B,FA,9F,DD,68,DD,45,0921
1660 DATA E5,1A,77,13,CD,BE,8D,10,F8,E1,CD [2716]
,AC,8D,0D,20,EE,C9,23,7E,2B,0A40
1670 DATA FE,07,28,3D,FE,11,28,39,FE,1B,28 [3450]
,35,FE,25,28,49,DA,CA,8F,C3,08DA
1680 DATA A3,90,23,7E,2B,FE,07,28,24,FE,11 [3404]
,28,38,DA,CA,8F,C3,A3,90,23,090B
1690 DATA 7E,2B,FE,07,28,13,FE,11,28,0F,FE [2786]
,1B,28,0B,FE,25,28,07,FE,2F,06FA
1700 DATA 28,1B,C3,CA,8F,CD,87,90,28,04,38 [3064]
,39,30,2D,ED,5F,CB,47,CA,6D,08D2
1710 DATA 8F,CB,4F,CA,9E,8F,C3,CA,8F,CD,87 [4221]
,90,DA,6D,8F,CA,7D,90,D2,9E,0CBD
1720 DATA 8F,ED,5F,CB,47,C2,6D,8F,C3,9E,8F [2229]
,2A,16,9F,7E,23,56,BA,C9,ED,0AE1
1730 DATA 5F,CB,47,C2,CA,8F,C3,9E,8F,ED,5F [3916]
,CB,47,CA,6D,8F,C3,CA,8F,CD,0C89
1740 DATA 87,90,DA,6D,8F,D2,9E,8F,21,60,9F [4016]
,3A,13,9F,5F,87,83,5F,16,00,08D6
1750 DATA 19,C9,01,90,A2,3A,13,9F,57,CB,27 [3088]
,CB,27,CB,27,5F,CB,23,83,CB,08C9
1760 DATA 22,92,6F,26,00,09,22,FA,9F,C9,F5 [3410]
,23,ED,5F,CB,47,20,04,3E,07,07B5
1770 DATA 18,02,3E,11,77,2B,F1,A7,CA,9E,8F [2804]
,C3,6D,8F,97,32,13,9F,21,9F,0894
1780 DATA A1,22,FA,9F,CD,AC,90,7E,23,23,5E [3716]
,CD,EA,8F,CD,0F,91,20,F1,CD,0B18
1790 DATA 9F,96,F1,E1,D1,C1,C9,3A,32,9F,57 [2130]
,3A,13,9F,3C,32,13,9F,BA,C9,0A53
1800 DATA 2A,14,9F,2B,2B,22,14,9F,7C,B5,CA [3381]
,B5,91,11,76,02,CD,EA,BB,21,0865
1810 DATA 00,00,11,06,00,CD,F9,BB,C9,CD,D3 [4534]
,8C,01,01,00,11,40,01,21,C8,06CA
1820 DATA 00,C5,09,EB,09,EB,CD,EA,BB,C1,C5 [3888]
,60,69,29,11,00,00,D5,E5,CD,0A2F
1830 DATA 9B,91,CD,F9,BB,E1,D1,D5,E5,CD,9B [3754]
,91,EB,CD,F9,BB,E1,D1,D5,E5,0FEA
1840 DATA CD,F9,BB,D1,E1,CD,F9,BB,E1,23,23 [3529]
,44,4D,C5,E5,3A,10,76,A7,3E,0BBB
1850 DATA 00,32,10,76,11,69,95,20,03,11,60 [3199]
,95,CD,69,92,E1,C1,7D,FE,43,0818
1860 DATA 20,A9,7C,A7,C0,18,A4,7C,2F,67,7D [2996]
,ED,44,6F,C9,E5,21,01,08,CD,093C
1870 DATA 75,BB,E1,CD,00,B9,CD,79,EE,CD,03 [4929]
,B9,C9,CD,D3,8C,3A,25,9F,3D,0B84
1880 DATA 28,4A,F5,CD,39,91,F1,32,25,9F,C3 [3601]
,76,85,3A,32,9F,21,5D,9F,11,08DC
1890 DATA 03,00,47,3A,00,9F,19,4E,B9,28,08 [4919]
,0D,0D,B9,28,03,10,F1,C9,23,055E
1900 DATA 23,3A,01,9F,4E,0C,B9,28,C8,0D,0D [2586]
,B9,28,C3,0D,0D,B9,28,BE,2B,06A2
1910 DATA 2B,10,D8,C9,7E,23,4E,C5,CD,34,BD [4579]
,23,C1,10,F5,C9,3E,01,CD,90,099C
1920 DATA BB,21,04,0C,E5,CD,75,BB,CD,20,92 [3184]
,E1,23,7D,FE,07,20,F2,18,0C,0909
1930 DATA 06,0F,C5,3E,8F,CD,5A,BB,C1,10,F7 [2623]
,C9,3E,01,CD,96,BB,3E,03,CD,0985
1940 DATA 90,BB,21,05,0D,CD,75,BB,21,5B,92 [3696]
,E5,11,57,95,CD,69,92,21,00,0854
1950 DATA 40,CD,A4,96,E1,7E,23,CD,5A,BB,B7 [3967]
,20,EA,CD,96,BB,C3,65,94,2A,0B70
1960 DATA 20,47,41,4D,45,20,4F,56,45,52,20 [3508]
,2A,00,21,B2,AD,06,09,1A,77,0500
1970 DATA 23,13,10,FA,21,B2,AD,CD,AA,BC,C9 [3839]
,3E,01,CD,0E,BC,CD,99,95,3E,09CB
1980 DATA 03,CD,90,BB,21,09,08,CD,75,BB,3E [3691]
,C2,CD,5A,BB,CD,3E,93,3E,C3,09CB
1990 DATA CD,5A,BB,3E,0A,F5,26,08,6F,CD,75 [4305]
,BB,3E,95,CD,5A,BB,F1,F5,26,0A7A
2000 DATA 1F,6F,CD,75,BB,3E,95,CD,5A,BB,F1 [4303]
,3C,FE,11,20,E1,21,11,08,CD,0984
2010 DATA 75,BB,3E,C1,CD,5A,BB,CD,3E,93,3E [3940]
,C0,CD,5A,BB,3E,01,CD,90,BB,0AE6
2020 DATA 21,0B,0C,CD,75,BB,21,08,93,CD,FF [4441]
,92,21,0D,0C,CD,75,BB,21,1A,07C1
2030 DATA 93,CD,FF,92,21,0F,0C,CD,75,BB,21 [3298]
,20,93,CD,FF,92,3E,02,CD,90,09F9
2040 DATA BB,18,4B,7E,23,CD,5A,BB,B7,20,F8 [2623]
,C9,46,41,52,42,45,4E,45,49,0875
2050 DATA 4E,53,54,45,4C,4C,55,4E,47,00,53 [2895]
```

```
,50,49,45,4C,00,45,4E,44,45,0555
2060 DATA 00,0A,0D,52,41,48,4D,45,4E,00,0A [3871]
,0A,0D,53,43,48,52,45,49,42,03F3
2070 DATA 46,41,52,42,45,00,06,16,C5,3E,9A [3450]
,CD,5A,BB,C1,10,F7,C9,21,0B,07B8
2080 DATA 0A,E5,CD,75,BB,3E,F3,CD,5A,BB,CD [3084]
,06,BB,E1,E5,F5,26,0A,CD,75,0BBA
2090 DATA BB,3E,20,CD,5A,BB,F1,E1,FE,F1,28 [4509]
,21,FE,0B,28,1D,FE,F0,28,0E,0A77
2100 DATA FE,0A,28,0A,FE,E0,28,1E,FE,58,28 [4326]
,1A,18,CB,2D,2D,7D,FE,0B,30,07E9
2110 DATA C4,2E,0B,18,C0,2C,2C,7D,FE,0F,28 [3779]
,B9,38,B7,2E,0F,18,B3,7D,FE,080A
2120 DATA 0B,28,08,FE,0D,C4,00,00,C3,66,85 [3728]
,3E,01,CD,0E,BC,21,01,0A,E5,069F
2130 DATA CD,75,BB,3E,03,CD,90,BB,21,08,93 [2805]
,CD,FF,92,E1,2C,CD,75,BB,06,0A80
2140 DATA 11,3E,02,CD,90,BB,3E,3D,CD,5A,BB [3575]
,10,F9,21,04,01,CD,75,BB,3E,0830
2150 DATA 01,CD,90,BB,21,25,93,CD,FF,92,CD [3726]
,2B,94,78,32,00,75,3E,01,F5,092F
2160 DATA 21,2E,93,CD,FF,92,F1,F5,C6,30,CD [3344]
,5A,BB,CD,2B,94,21,00,75,F1,0B11
2170 DATA 5F,16,00,19,70,3C,FE,04,28,02,18 [3394]
,DF,3A,00,75,47,4F,CD,38,BC,0663
2180 DATA 3E,01,21,01,75,46,48,E5,F5,CD,32 [2422]
,BC,F1,E1,3C,23,FE,04,20,F1,093D
2190 DATA C3,7B,92,06,00,C5,48,CD,38,BC,CD [3234]
,06,BB,FE,58,28,2A,FE,0B,28,090B
2200 DATA 10,FE,F0,28,17,FE,0A,28,13,FE,E0 [2705]
,28,1A,FE,F1,20,E5,C1,05,CB,0A25
2210 DATA 78,28,DA,3E,1A,47,18,D5,C1,04,78 [3890]
,FE,1B,20,CE,06,00,18,CA,C1,07F3
2220 DATA C9,3E,01,CD,0E,BC,CD,5C,96,CD,99 [4471]
,95,3E,01,CD,90,BB,21,0B,09,08E5
2230 DATA CD,75,BB,21,05,95,CD,FF,92,2A,50 [3033]
,9F,CD,F4,94,3E,01,CD,90,BB,0ADB
2240 DATA 21,0D,09,CD,75,BB,21,11,95,CD,FF [2570]
,92,2A,50,9F,ED,5B,0E,9F,A7,090E
2250 DATA ED,52,CB,7C,28,03,EB,18,06,2A,50 [3415]
,9F,22,0E,9F,CD,F4,94,3E,03,0838
2260 DATA CD,90,BB,21,13,03,CD,75,BB,21,1D [4368]
,95,CD,FF,92,21,15,06,CD,75,08FB
2270 DATA BB,21,41,95,CD,FF,92,CD,3D,96,06 [3436]
,64,C5,CD,09,BB,C1,10,F9,CD,0B07
2280 DATA 06,BB,FE,6D,CA,6A,96,FE,4D,CA,6A [3358]
,96,FE,20,CA,66,85,FE,58,CA,0BFE
2290 DATA 66,85,18,E7,E5,3E,02,CD,90,BB,E1 [3198]
,CD,00,B9,CD,79,EE,CD,03,B9,0B4B
2300 DATA C9,59,6F,75,72,20,53,63,6F,72,65 [3119]
,3A,00,48,69,67,68,2D,53,63,0731
2310 DATA 6F,72,65,3A,00,50,72,65,73,73,20 [3554]
,5B,53,50,41,43,45,5D,20,6F,0660
2320 DATA 72,20,5B,46,49,52,45,5D,20,74,6F [3476]
,20,73,74,61,72,74,2E,2E,2E,064B
2330 DATA 00,50,72,65,73,73,20,5B,4D,5D,20 [2989]
,74,6F,20,6D,65,6E,75,65,2E,069D
2340 DATA 2E,2E,00,01,00,00,01,C2,00,0F,07 [3680]
,00,01,00,01,96,00,00,0F,07,01E4
2350 DATA 00,01,01,00,00,32,00,00,00,00,01 [3323]
,64,FC,01,01,0C,FF,04,00,19,02BF
2360 DATA 01,00,08,07,07,3E,DF,8B,A8,DF,8B [3973]
,A8,DF,8B,A8,DF,8B,A8,DF,8B,0A07
2370 DATA A8,63,6F,64,65,2E,62,69,6E,21,19 [2813]
,01,CD,75,BB,3E,01,CD,90,BB,0839
2380 DATA 21,17,96,CD,FF,92,3E,01,CD,DE,BB [4516]
,21,00,00,54,5C,D5,E5,CD,F0,0A19
2390 DATA BB,A7,28,31,E1,D1,D5,E5,E5,EB,29 [2678]
,29,11,80,00,19,EB,E1,D5,29,0ABD
2400 DATA 29,11,2C,01,19,D1,CD,EA,BB,11,FE [4715]
,FF,21,00,00,E5,CD,F9,BB,D1,0A29
2410 DATA 21,02,00,D5,E5,CD,F9,BB,D1,E1,CD [3319]
,F9,BB,E1,D1,13,13,7B,FE,61,0C43
2420 DATA 38,BE,11,00,00,23,23,7D,FE,11,38 [3599]
,B4,3A,01,75,47,4F,3E,01,CD,0617
2430 DATA 32,BC,21,19,01,CD,75,BB,21,1E,96 [3646]
,CD,FF,92,C9,52,75,6E,6E,65,092A
2440 DATA 72,00,A4,20,31,39,38,37,20,62,79 [3208]
,20,44,69,65,74,6D,61,72,20,0610
2450 DATA 26,20,53,76,65,6E,20,47,72,61,64 [2814]
,6C,00,3A,01,75,47,4F,3E,01,0571
2460 DATA CD,32,BC,3A,02,75,47,4F,3E,02,CD [2518]
,32,BC,3A,03,75,47,4F,3E,03,0686
2470 DATA CD,32,BC,C9,21,D9,B1,36,14,11,DA [4003]
,B1,01,20,00,ED,B0,C9,21,08,08C5
2480 DATA 01,CD,75,BB,3E,14,CD,5A,BB,C3,83 [3605]
,92,CD,98,96,CD,06,BB,FE,0D,0A9E
2490 DATA 28,0C,FE,65,CA,92,96,FE,45,CA,92 [2638]
,96,18,ED,CD,9F,96,C9,31,00,0ABF
2500 DATA C0,C3,7B,92,21,00,00,22,FE,B1,C9 [3105]
,21,5C,AC,18,F7,2B,7C,B5,20,08FF
2510 DATA FB,C9,00,00,00,00,00,00,00,00,00 [3891]
,00,00,00,00,00,00,00,00,00,01C4

10 REM"**************************          [1708]
20 REM"*** Shape's fuer Runner ***          [1787]
30 REM"**************************          [1708]
40 FOR x=&A000 TO &A1B5 STEP 20             [1938]
50 : sum=0                                  [680]
60 : FOR y=0 TO 19                          [1055]
70 : : READ a$:zahl=VAL("&"+a$)             [2454]
80 : : sum=sum+zahl                         [694]
90 : POKE x+y,zahl                          [1094]
100 : NEXT y                                [431]
110 : READ chksum$                          [656]
120 : chksum=VAL("&"+chksum$)               [1115]
130 : zeile=(x-&A000)/2+180                 [1707]
140 : IF chksum=sum THEN PRINT"Zeile";zeil [4431]
e;"OK.":GOTO 160
150 : PRINT"Fehler in Zeile:";zeile;"."     [4260]
160 NEXT x                                  [356]
170 SAVE"code.bin",b,&A000,&1B8             [1984]
180 DATA 00,70,80,00,C3,0C,10,D3,08,10,87, [3296]
08,10,87,0C,10,87,0C,00,0F,049E
190 DATA 00,00,03,08,11,88,00,33,EE,00,33, [3378]
BB,00,33,BB,00,33,BB,00,33,04C2
200 DATA BB,00,33,BB,00,11,8A,00,00,C2,00, [3203]
00,C2,00,00,0E,00,00,0E,00,03E4
210 DATA 00,0E,00,00,FF,00,00,FF,00,00,70, [2640]
80,00,C3,0C,10,D3,08,10,87,054D
220 DATA 08,10,87,0C,10,87,0C,00,0F,00,00, [2610]
03,08,33,EE,00,77,FF,00,FF,04FE
230 DATA FF,35,EE,FF,35,EE,FF,35,EE,00,44, [2478]
60,0F,00,60,0F,00,00,0F,08,079F
240 DATA 00,0F,0C,CF,07,0C,CF,08,0C,CF,08, [3314]
0C,CC,00,EE,00,00,EE,10,E0,065B
250 DATA 00,03,3C,00,01,BC,80,01,1E,80,03, [3341]
1E,80,03,1E,80,00,0F,00,01,036D
260 DATA 0C,00,00,11,88,00,77,CC,00,DD,CC, [2315]
00,DD,CC,00,DD,CC,00,DD,CC,088C
270 DATA 00,DD,CC,00,15,88,00,34,00,00,34, [3920]
00,00,07,00,00,07,00,00,07,02C3
280 DATA 00,00,FF,00,00,FF,00,10,E0,00,03, [3027]
3C,80,01,BC,80,01,1E,80,03,058C
290 DATA 1E,80,03,1E,80,00,0F,00,01,0C,00, [3457]
00,11,88,00,77,CC,00,FF,EE,0524
300 DATA E2,FF,FF,E2,FF,77,22,00,77,00,0F, [3664]
60,00,0F,60,01,0F,00,03,0F,06D1
310 DATA 00,03,0E,3F,03,01,3F,03,01,3F,77, [3068]
00,33,77,00,00,00,70,80,60,0347
320 DATA F0,C0,60,F0,C0,EE,F0,C0,EE,F0,C0, [3478]
EE,70,80,FF,00,00,77,FF,8E,0DDD
330 DATA 77,FF,8E,33,FF,CC,33,FF,CC,11,FF, [3237]
88,11,FF,88,11,EE,88,03,0E,0AC8
340 DATA 00,03,0E,00,03,06,00,03,06,00,77, [3290]
06,00,77,06,00,00,06,00,00,0123
350 DATA 77,00,00,77,00,10,E0,00,30,F0,60, [3457]
30,F0,60,30,F0,77,30,F0,77,080C
360 DATA 10,E0,77,00,00,FF,71,FF,EE,71,FF, [4167]
EE,33,FF,CC,33,FF,CC,11,FF,0C2E
370 DATA 88,11,FF,88,11,77,88,00,07,0C,00, [2855]
07,0C,00,06,0C,00,06,0C,00,037A
380 DATA 06,0C,00,06,EE,00,06,EE,00,EE,00, [2975]
00,EE,00,00,00,00,00,00,00,03D6
390 DATA 00,03,0C,07,0E,0F,0F,0F,0F,0F,0F, [3860]
0F,0F,07,0E,03,0C,00,00,00,00C0
```

# MEMORY

**Die Spielregeln für Memory dürften wohl jedem bekannt sein. Die Spieler drehen nacheinander zwei der verdeckten Karten um. Sind diese Karten ein Paar, darf man sie behalten und es nochmal versuchen, andernfalls müssen die Karten wieder verdeckt abgelegt werden und der nächste Spieler ist an der Reihe. Diese Version des Spieles ist für einen Spieler ausgelegt, der gegen den Computer antreten muß.**

### Nach dem Start des Programms...

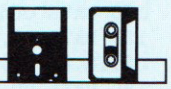
erfolgt die Frage nach dem Beginner und dem Schwierigkeitsgrad (0=leicht, 9=schwer). Wenn der Spieler am Zug ist, erscheint ein blinkender Cursor, der mit dem Joystick bewegt werden kann. Durch Druck auf dem Feuerknopf wird die entsprechende Karte umgedreht. Mit der H-Taste läßt sich das Spiel vorzeitig beenden und alle restlichen Karten werden umgedreht.

### Das Abtippen

Das erste Listing wird nach dem Abtippen abgespeichert, während die beiden anderen gestartet (sicherheitshalber sollten auch die Listings 2 & 3 vor dem Starten gesichert werden) und der damit erzeugte Code abgespeichert wird.

(Werner Schmidt/SH)

## für 464-664-6128



```
10 MEMORY 33999:LOAD"!mg":LOAD"!mm":FOR n= [3739]
40656 TO 40912:POKE n,0:NEXT
20 FOR n=0 TO 14:READ a:INK n,a:NEXT:INK 1 [3561]
5,23,8:BORDER 0:MODE 0
30 DATA 0,13,6,15,18,24,26,1,3,9,12,16,25, [1531]
11,19
40 RANDOMIZE TIME:DIM m(49),b(9,4),bw(9,4) [2517]
50 PEN 2:PRINT TAB(5);"M ";:PEN 13:PRINT"E [6041]
 ";:PEN 5:PRINT"M ";:PEN 3:PRINT"O ";:PEN
4:PRINT"R ";:PEN 6:PRINT"Y"
60 PEN 7:LOCATE 3,3:PRINT CHR$(164);" Wern [4051]
er Schmidt"
70 LOCATE 17,4:PRINT"9/86"                 [1071]
80 PEN 1:LOCATE 1,7:PRINT"Wer soll anfange [2277]
n ?"
90 LOCATE 1,9:PRINT"(1) Ich"               [961]
100 LOCATE 1,10:PRINT"(2) Du"              [1232]
110 a$=INKEY$:IF a$<>"1" AND a$<>"2" THEN  [1364]
110
120 LOCATE 1,13:PRINT"Schwierigkeitsgrad ? [3716]
"
130 LOCATE 1,15:PRINT"(0-9)"               [1123]
140 b$=INKEY$:IF b$<"0" OR b$>"9" THEN 140 [3219]
150 sch=(VAL(b$)/10)+0.1                   [1429]
160 FOR n=1 TO 25:m(n)=2:NEXT              [1929]
170 MODE 0:FOR n=0 TO 9:FOR m=0 TO 4       [2460]
180 a=INT(RND*25)+1:IF m(a)=0 THEN 180     [3130]
190 b(n,m)=a:m(a)=m(a)-1:bw(n,m)=2:NEXT m, [2217]
n
200 FOR n=0 TO 4:FOR m=0 TO 9              [1160]
210 CALL 41000,49152+8*m+320*n,34000:m(10* [3093]
n+m)=0:NEXT m,n
220 PEN 11:LOCATE 2,25:PRINT"Ich: 0";TAB ( [3423]
14);"Du: 0";
230 zahl=0:ps=0:pc=0                       [1169]
240 IF a$="1" THEN 470                     [917]
250 zs=0                                   [651]
260 s=0:z=0:sv=0:zv=0:LOCATE 1,3:PEN 15:PR [4448]
INT CHR$(22);CHR$(1);CHR$(230);CHR$(22);CH
R$(0);
270 IF JOY(0)=1 AND z>0 THEN z=z-1         [1685]
280 IF JOY(0)=2 AND z<4 THEN z=z+1         [933]
290 IF JOY(0)=4 AND s>0 THEN s=s-1         [2230]
300 IF JOY(0)=8 AND s<9 THEN s=s+1         [1982]
310 IF JOY(0)=16 THEN 370                  [1750]
320 IF INKEY$="h" THEN 680                 [339]
330 IF bw(sv,zv)=2 THEN a=34000 ELSE IF bw [6476]
(sv,zv)=1 THEN a=256*b(sv,zv)+34000 ELSE I
F bw(sv,zv)=0 THEN a=40656
340 IF s<>sv OR z<>zv THEN CALL 41000,4915 [7393]
2+8*sv+320*zv,a:LOCATE s*2+1,(z+1)*4-1:PEN
 15:PRINT CHR$(22);CHR$(1);CHR$(230);CHR$(
22);CHR$(0);
350 sv=s:zv=z                              [559]
360 FOR n=1 TO 30:NEXT:GOTO 270            [1771]
370 IF bw(s,z)<>2 THEN 320                 [903]
380 CALL 41000,49152+8*s+320*z,34000+256*b [3465]
(s,z):m(10*z+s)=b(s,z)
390 zs=zs+1:IF zs<>2 THEN bw(s,z)=1:sm=s:z [5967]
m=z:LOCATE s*2+1,(z+1)*4-1:PEN 15:PRINT CH
R$(22);CHR$(1);CHR$(230);CHR$(22);CHR$(0);
:GOTO 320
400 FOR n=1 TO 1000:NEXT                   [1110]
410 IF b(sm,zm)<>b(s,z) THEN bw(sm,zm)=2:C [7563]
ALL 41000,49152+8*sm+320*zm,34000:CALL 410
00,49152+8*s+320*z,34000:FOR so=50 TO 500
STEP 20:SOUND 1,so,1:NEXT:GOTO 470
420 FOR so=500 TO 50 STEP -20:SOUND 4,so,1 [2785]
:NEXT
430 CALL 41000,49152+8*sm+320*zm,40656:CAL [4142]
L 41000,49152+8*s+320*z,40656
440 bw(sm,zm)=0:bw(s,z)=0:m(10*zm+sm)=0:m( [4157]
10*z+s)=0
450 ps=ps+1:PEN 11:LOCATE 17,25:PRINT ps;  [868]
460 zahl=zahl+1:IF zahl<>25 THEN 250 ELSE  [2098]
720
470 IF RND>sch THEN 520                    [1017]
480 FOR n=0 TO 49:IF m(n)=0 THEN 510       [1417]
490 FOR m=n+1 TO 49:IF m(n)=m(m) THEN zm=I [4613]
NT(n/10):sm=n-10*zm:n=INT(m/10):m=m-10*n:G
OTO 600
500 NEXT m                                 [371]
510 NEXT n                                 [366]
520 n=INT(RND*50):zm=INT(n/10):sm=n-10*zm: [4249]
IF bw(sm,zm)<>2 THEN 520
530 m(n)=b(sm,zm)                          [267]
540 IF RND>sch THEN 580                    [1005]
550 FOR m=0 TO 49                          [414]
560 IF m(n)=m(m) AND n<>m THEN 590         [1332]
570 NEXT m                                 [371]
580 m=INT(RND*50):IF m=n OR bw(m-10*INT(m/ [3883]
10),INT(m/10))<>2 THEN 580
590 n=INT(m/10):m=m-10*n                   [1141]
600 CALL 41000,49152+8*sm+320*zm,34000+256 [6283]
*b(sm,zm):CALL 41000,49152+8*m+320*n,34000
+256*b(m,n)
610 FOR zs=1 TO 1000:NEXT                  [1685]
620 IF b(sm,zm)<>b(m,n) THEN m(10*n+m)=b(m [9122]
,n):CALL 41000,49152+8*sm+320*zm,34000:CAL
L 41000,49152+8*m+320*n,34000:FOR so=50 TO
 500 STEP 20:SOUND 1,so,1:NEXT:GOTO 250
630 FOR so=500 TO 50 STEP -20:SOUND 4,so,1 [2785]
:NEXT
640 bw(sm,zm)=0:bw(m,n)=0:m(10*zm+sm)=0    [3108]
650 CALL 41000,49152+8*sm+320*zm,40656:CAL [4497]
L 41000,49152+8*m+320*n,40656
660 pc=pc+1:PEN 11:LOCATE 6,25:PRINT pc    [1808]
670 zahl=zahl+1:IF zahl<>25 THEN 470 ELSE  [2418]
720
680 FOR n=0 TO 4:FOR m=0 TO 9              [1160]
690 IF bw(m,n)=2 THEN CALL 41000,49152+8*m [3500]
+320*n,34000+256*b(m,n)
```

```
700 NEXT m,n                                        [407]
710 IF JOY(0)<>16 THEN 710 ELSE MODE 0:GOT [2966]
O 50
720 PEN 13:LOCATE 2,4:IF pc>ps THEN PRINT" [4751]
Ich habe gewonnen" ELSE PRINT"Du hast gewo
nnen"
730 IF JOY(0)<>16 THEN 730 ELSE MODE 0:GOT [2224]
O 50
```

```
10 MEMORY &84CF:FOR n=&84D0 TO &9ECF:READ  [3358]
a:POKE n,a:NEXT
20 SAVE"mg",b,&84D0,&1A00                           [799]
30 DATA 192,192,192,192,192,192,192,128,19 [2187]
2,192
40 DATA 192,192,192,192,192,128,192,192,19 [1858]
2,192
50 DATA 192,192,192,128,192,192,192,192,19 [1267]
2,192
60 DATA 192,128,192,252,252,252,252,252,23 [1767]
2,128
70 DATA 192,252,252,252,252,252,232,128,19 [1875]
2,169
80 DATA 192,192,192,192,232,128,192,169,19 [2146]
2,192
90 DATA 192,129,232,128,192,232,66,192,192 [1408]
,66
100 DATA 232,128,192,232,66,192,192,66,232 [2054]
,128
110 DATA 192,232,129,192,129,192,232,128,1 [1858]
92,232
120 DATA 129,192,129,192,232,128,192,232,1 [2539]
92,195
130 DATA 194,192,232,128,192,232,192,195,1 [2062]
94,192
140 DATA 232,128,192,232,192,195,194,192,2 [1985]
32,128
150 DATA 192,232,192,195,194,192,232,128,1 [2248]
92,232
160 DATA 192,195,194,192,232,128,192,232,1 [2199]
92,195
170 DATA 194,192,232,128,192,232,129,192,1 [1848]
29,192
180 DATA 232,128,192,232,129,192,129,192,2 [1935]
32,128
190 DATA 192,232,66,192,192,66,232,128,192 [1807]
,232
200 DATA 66,192,192,66,232,128,192,169,192 [1443]
,192
210 DATA 192,129,232,128,192,169,192,192,1 [1488]
92,129
220 DATA 232,128,192,252,252,252,252,252,2 [1533]
32,128
230 DATA 192,252,252,252,252,252,232,128,1 [1928]
92,192
240 DATA 192,192,192,192,192,128,192,192,1 [1858]
92,192
250 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
260 DATA 192,128,192,192,192,192,192,192,1 [2288]
92,128
270 DATA 0,0,0,0,0,0,0,0,0,0                       [1269]
280 DATA 0,0,0,0,0,0,192,192,192,192       [1548]
290 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
300 DATA 192,128,192,192,192,192,192,192,1 [2087]
48,128
310 DATA 192,192,192,192,192,192,104,128,1 [2051]
92,192
320 DATA 192,192,192,148,192,128,192,192,1 [2053]
92,129
330 DATA 192,129,192,128,192,192,192,3,66, [2132]
129
340 DATA 192,128,192,192,129,3,3,129,192,1 [1757]
28
350 DATA 192,192,3,3,3,3,192,128,192,129   [1723]
360 DATA 3,3,3,3,192,128,192,3,3,3          [1247]
370 DATA 3,3,66,128,129,3,3,3,3,3           [1275]
380 DATA 3,128,132,12,12,12,12,12,12,128    [1831]
390 DATA 132,12,12,12,12,12,12,128,132,12   [1916]
400 DATA 12,12,12,12,12,128,132,12,12,12    [1660]
410 DATA 12,12,12,128,132,243,179,243,166,  [1822]
12
420 DATA 12,128,132,243,179,243,166,12,12,  [1763]
128
430 DATA 132,243,179,243,166,12,12,128,132 [2213]
,243
440 DATA 179,243,166,195,134,128,132,243,1 [2061]
79,243
450 DATA 166,195,134,128,132,243,179,243,1 [2015]
66,195
460 DATA 134,128,132,243,179,243,166,195,1 [2011]
34,128
470 DATA 132,12,12,12,12,65,134,128,132,12 [2008]
480 DATA 12,12,12,195,134,128,132,12,12,12 [1854]
490 DATA 12,195,134,128,132,12,12,12,12,19 [2316]
5
500 DATA 134,128,192,192,192,192,192,192,1 [2237]
92,128
510 DATA 192,192,192,192,192,192,192,128,1 [2187]
92,192
520 DATA 192,192,192,192,192,128,0,0,0,0   [1544]
530 DATA 0,0,0,0,0,0,0,0,0,0                [1269]
540 DATA 0,0,192,192,192,192,192,192,192,1 [2242]
28
550 DATA 192,192,192,192,192,192,192,128,1 [2187]
92,192
560 DATA 192,192,192,192,192,128,192,192,1 [1858]
92,192
570 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
580 DATA 192,128,192,192,192,192,192,192,1 [2288]
92,128
590 DATA 192,192,192,252,252,252,252,128,1 [2053]
92,192
600 DATA 212,252,252,252,233,128,192,192,2 [2208]
52,252
610 DATA 252,252,233,128,192,212,252,252,2 [2455]
52,252
620 DATA 195,128,192,252,252,252,252,233,1 [2065]
95,128
630 DATA 212,252,252,252,252,195,195,128,1 [2042]
96,204
640 DATA 204,204,201,195,195,128,196,204,2 [2096]
04,204
650 DATA 201,195,195,128,196,204,204,204,2 [2340]
01,195
660 DATA 195,128,196,204,204,204,201,195,1 [2022]
95,128
670 DATA 196,204,204,204,201,195,195,128,1 [2316]
96,204
680 DATA 204,204,201,195,195,128,196,204,2 [2096]
04,204
690 DATA 201,195,195,128,196,204,204,204,2 [2340]
01,195
700 DATA 195,128,196,204,204,204,201,195,1 [2022]
95,128
710 DATA 196,204,204,204,201,195,195,128,1 [2316]
96,204
720 DATA 204,204,201,195,195,128,196,204,2 [2096]
04,204
730 DATA 201,195,194,128,196,204,204,204,2 [2541]
01,195
740 DATA 192,128,196,204,204,204,201,194,1 [2425]
92,128
750 DATA 196,204,204,204,201,192,192,128,1 [1809]
96,204
760 DATA 204,204,200,192,192,128,192,192,1 [2023]
```

```
92,192
770 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
780 DATA 192,128,0,0,0,0,0,0,0,0           [1248]
790 DATA 0,0,0,0,0,0,0,0,192,192           [1582]
800 DATA 192,192,192,192,192,128,192,192,1 [1858]
92,192
810 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
820 DATA 192,128,192,192,192,192,192,192,1 [2288]
92,128
830 DATA 192,192,192,192,192,192,192,128,1 [2187]
92,192
840 DATA 192,192,192,192,192,128,192,192,1 [1858]
92,192
850 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
860 DATA 192,128,145,51,51,51,51,51,51,128 [1738]
870 DATA 145,60,60,60,60,60,57,128,145,60  [1895]
880 DATA 60,60,60,60,57,128,145,51,51,51    [1595]
890 DATA 51,51,51,128,145,51,51,51,51,51    [608]
900 DATA 51,128,145,34,51,252,179,34,51,12 [2048]
8
910 DATA 145,34,51,252,179,34,51,128,145,5 [1982]
1
920 DATA 51,51,51,51,51,128,145,51,51,51    [1317]
930 DATA 51,51,51,128,145,51,51,51,51,51    [608]
940 DATA 51,128,145,51,15,15,15,27,51,128   [1518]
950 DATA 145,39,51,51,51,39,51,128,145,39   [1925]
960 DATA 51,51,51,39,51,128,192,192,192,19 [1515]
2
970 DATA 192,192,192,128,192,192,192,192,1 [1267]
92,192
980 DATA 192,128,192,192,192,192,192,192,1 [2288]
92,128
990 DATA 192,192,192,192,192,192,192,128,1 [2187]
92,192
1000 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
1010 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
1020 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1030 DATA 192,192,192,192,192,192,192,128, [1725]
0,0
1040 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
1050 DATA 0,0,0,0,192,192,192,192,192,192   [2262]
1060 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1070 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
1080 DATA 192,208,192,192,192,128,192,192, [1356]
192,240
1090 DATA 192,192,192,128,192,192,192,240, [1307]
192,192
1100 DATA 192,128,192,192,192,240,192,192, [2820]
192,128
1110 DATA 192,192,192,224,192,192,192,128, [2043]
192,192
1120 DATA 192,224,192,192,192,128,192,192, [2193]
192,64
1130 DATA 192,192,192,128,192,192,192,64,1 [1513]
92,192
1140 DATA 192,128,192,192,72,64,72,192,192 [1916]
,128
1150 DATA 192,192,12,4,72,192,192,128,192, [1827]
192
1160 DATA 12,12,72,192,192,128,192,192,12, [1793]
12
1170 DATA 72,192,192,128,192,192,12,12,72, [1597]
192
1180 DATA 192,128,192,192,12,12,72,192,192 [1787]
,128
1190 DATA 192,192,12,12,72,192,192,128,192 [1695]
,192
1200 DATA 12,12,72,192,192,128,192,192,12, [1793]
12
1210 DATA 72,192,192,128,192,192,12,12,72, [1597]
192
1220 DATA 192,128,192,192,12,12,72,192,192 [1787]
,128
1230 DATA 192,192,12,12,72,192,192,128,192 [1695]
,192
1240 DATA 12,12,72,192,192,128,192,192,12, [1793]
12
1250 DATA 72,192,192,128,192,192,12,12,72, [1597]
192
1260 DATA 192,128,192,145,12,12,25,192,192 [1706]
,128
1270 DATA 192,145,12,12,25,192,192,128,192 [1494]
,145
1280 DATA 51,51,51,192,192,128,192,192,51, [1501]
51
1290 DATA 98,192,192,128,0,0,0,0,0,0        [1200]
1300 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
1310 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
1320 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
1330 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
1340 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1350 DATA 192,192,192,192,192,192,192,128, [2070]
129,3
1360 DATA 3,3,3,3,3,128,129,3,3,3           [1235]
1370 DATA 3,3,3,128,129,3,3,3,3,3           [679]
1380 DATA 3,128,129,176,172,111,230,3,3,12 [2656]
8
1390 DATA 129,176,172,111,230,3,3,128,129, [1477]
176
1400 DATA 172,111,230,22,3,128,129,0,0,0    [1825]
1410 DATA 0,3,3,128,129,176,172,111,230,22  [1573]
1420 DATA 3,128,129,176,172,111,230,3,3,12 [2656]
8
1430 DATA 129,176,172,111,230,22,3,128,129 [1847]
,176
1440 DATA 172,111,230,3,3,128,129,0,0,0     [1550]
1450 DATA 0,22,3,128,129,176,172,111,230,3 [1352]
1460 DATA 3,128,129,176,172,111,230,3,3,12 [2656]
8
1470 DATA 129,3,3,3,3,3,41,128,129,3        [1630]
1480 DATA 3,3,3,3,3,128,129,3,3,3           [1235]
1490 DATA 3,3,3,128,192,192,192,192,192,19 [882]
2
1500 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1510 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
1520 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
1530 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
1540 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1550 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
1560 DATA 0,0,0,0,0,0,192,192,192,192       [1548]
1570 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
1580 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
1590 DATA 192,192,192,73,194,192,192,128,1 [2230]
92,192
1600 DATA 192,195,194,192,192,128,192,192, [1782]
192,195
1610 DATA 73,192,192,128,192,192,192,107,1 [1787]
95,192
1620 DATA 192,128,192,192,193,195,195,192, [1771]
192,128
1630 DATA 192,192,97,146,195,72,192,128,19 [1811]
2,192
1640 DATA 193,195,195,195,192,128,192,192, [2173]
```

```
73,97
1650 DATA 195,146,192,128,192,192,195,195, [1919]
195,97
1660 DATA 192,128,192,192,195,195,134,195, [1856]
192,128
1670 DATA 192,192,195,134,195,195,192,128, [2196]
192,192
1680 DATA 134,195,195,151,192,128,192,192, [2086]
97,195
1690 DATA 195,97,192,128,192,192,193,131,1 [1749]
95,194
1700 DATA 192,128,192,192,149,131,6,192,19 [1775]
2,128
1710 DATA 192,192,192,131,66,192,192,128,1 [1799]
92,192
1720 DATA 192,129,66,192,192,128,192,192,1 [2036]
92,128
1730 DATA 66,192,192,128,192,192,192,129,6 [1872]
6,192
1740 DATA 192,128,192,192,192,129,66,192,1 [1992]
92,128
1750 DATA 192,192,192,129,66,192,192,128,1 [1732]
92,192
1760 DATA 192,129,66,192,192,128,192,192,1 [2039]
92,129
1770 DATA 64,192,192,128,192,192,192,129,6 [2353]
6,192
1780 DATA 192,128,192,192,192,129,64,192,1 [2044]
92,128
1790 DATA 192,192,192,129,3,192,192,128,19 [2465]
2,192
1800 DATA 192,3,3,192,192,128,0,0,0,0       [1135]
1810 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
1820 DATA 0,0,192,192,192,192,192,192,192, [2242]
128
1830 DATA 192,212,252,252,252,252,192,128, [2032]
192,212
1840 DATA 252,252,252,252,192,128,192,212, [1873]
252,252
1850 DATA 252,252,192,128,192,212,252,252, [2070]
252,252
1860 DATA 192,128,192,212,252,252,252,252, [2174]
192,128
1870 DATA 192,212,252,12,92,252,192,128,19 [1612]
2,212
1880 DATA 252,12,92,252,192,128,192,212,25 [2044]
2,188
1890 DATA 252,252,192,128,192,212,252,188, [2200]
252,252
1900 DATA 192,128,192,212,252,188,252,252, [2456]
192,128
1910 DATA 192,212,252,188,252,252,192,128, [1515]
192,212
1920 DATA 252,188,252,252,192,128,192,212, [1357]
252,188
1930 DATA 252,252,192,128,192,212,252,188, [2200]
252,252
1940 DATA 192,128,192,212,252,188,252,252, [2456]
192,128
1950 DATA 192,212,252,188,252,252,192,128, [1515]
192,212
1960 DATA 252,188,252,252,192,128,192,212, [1357]
252,188
1970 DATA 252,252,192,128,192,212,252,188, [2200]
252,252
1980 DATA 192,128,192,212,252,188,252,252, [2456]
192,128
1990 DATA 192,212,252,188,252,252,192,128, [1515]
192,212
2000 DATA 252,188,252,252,192,128,192,212, [1357]
252,188
2010 DATA 252,252,192,128,192,212,252,188, [2200]
252,252
2020 DATA 192,128,192,192,192,128,192,192, [1877]
192,128
2030 DATA 192,192,192,128,192,192,192,128, [1556]
192,192
2040 DATA 192,128,192,192,192,128,192,192, [1877]
192,128
2050 DATA 192,192,192,128,192,192,192,128, [1556]
192,192
2060 DATA 192,128,0,0,0,0,0,0,0,0           [1248]
2070 DATA 0,0,0,0,0,0,0,0,192,192           [1582]
2080 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2090 DATA 192,192,192,128,192,128,0,0,0,0   [1541]
2100 DATA 192,128,192,128,0,4,0,0,192,128   [1929]
2110 DATA 192,128,0,4,0,0,192,128,192,128   [1039]
2120 DATA 0,12,8,0,192,128,192,128,0,4      [1676]
2130 DATA 0,0,192,128,192,128,0,4,0,0       [1972]
2140 DATA 192,128,192,128,0,0,0,0,192,128   [1906]
2150 DATA 192,128,0,0,0,0,192,128,192,128   [1700]
2160 DATA 0,0,0,0,192,128,192,128,0,80      [1828]
2170 DATA 0,0,192,128,192,128,0,80,0,0      [2079]
2180 DATA 192,128,192,128,0,240,160,0,192, [1487]
128
2190 DATA 192,128,0,80,0,0,192,128,192,128 [1880]
2200 DATA 0,80,0,0,192,128,192,128,0,0      [1578]
2210 DATA 0,0,192,128,192,128,0,0,0,0       [1783]
2220 DATA 192,128,192,128,0,0,0,0,192,128   [1906]
2230 DATA 192,128,0,16,0,0,192,128,192,128 [1547]
2240 DATA 0,16,0,0,192,128,192,128,0,48     [1919]
2250 DATA 32,0,192,128,192,128,0,16,0,0     [1895]
2260 DATA 192,128,192,128,0,16,0,0,192,128 [1389]
2270 DATA 192,128,0,0,0,0,192,128,192,128   [1700]
2280 DATA 0,0,0,0,192,128,192,128,0,0       [1586]
2290 DATA 0,0,192,128,192,192,192,128,192, [2137]
192
2300 DATA 192,128,192,192,192,128,192,192, [1877]
192,128
2310 DATA 192,192,192,128,192,192,192,128, [1440]
0,0
2320 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
2330 DATA 0,0,0,0,192,192,192,192,192,192   [2262]
2340 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2350 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2360 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2370 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2380 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2390 DATA 192,192,192,192,192,192,192,128, [2148]
208,240
2400 DATA 224,192,192,192,192,128,208,240, [2233]
224,192
2410 DATA 192,192,192,128,208,192,224,192, [2070]
192,208
2420 DATA 192,128,208,192,224,192,192,208, [2378]
208,128
2430 DATA 208,192,224,192,208,208,208,128, [1510]
208,192
2440 DATA 224,192,208,208,208,128,208,192, [2083]
224,192
2450 DATA 208,208,208,128,208,192,240,240, [1968]
240,240
2460 DATA 240,128,208,192,240,240,240,240, [1997]
240,128
2470 DATA 208,192,224,192,192,192,192,128, [1102]
208,192
2480 DATA 224,192,192,192,192,128,208,192, [2325]
224,192
2490 DATA 192,192,192,128,208,192,224,192, [2062]
192,192
2500 DATA 192,128,208,192,224,192,192,192, [2172]
192,128
2510 DATA 208,240,224,192,192,192,192,128, [2213]
208,240
2520 DATA 224,192,192,192,192,128,192,192, [2403]
```

```
192,192
2530 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2540 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2550 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2560 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2570 DATA 192,192,192,128,0,0,0,0,0,0      [1125]
2580 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
2590 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2600 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2610 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2620 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2630 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2640 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2650 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2660 DATA 192,128,192,193,195,199,195,192, [2430]
192,128
2670 DATA 192,201,195,199,195,194,192,128, [1936]
208,73
2680 DATA 195,199,195,214,192,128,12,201,1 [1619]
95,199
2690 DATA 195,214,232,128,208,225,195,199, [1663]
195,214
2700 DATA 252,128,132,225,195,199,195,214, [2098]
252,168
2710 DATA 208,73,195,199,195,214,252,128,1 [2337]
96,225
2720 DATA 195,199,195,214,232,128,140,225, [2469]
195,199
2730 DATA 195,214,192,128,192,201,195,199, [1979]
195,194
2740 DATA 192,128,192,193,195,199,195,192, [2430]
192,128
2750 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2760 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2770 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2780 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2790 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2800 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2810 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2820 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2830 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
2840 DATA 0,0,0,0,0,0,192,192,192,192      [1548]
2850 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2860 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2870 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2880 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
2890 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2900 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2910 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2920 DATA 192,192,192,192,192,128,192,192, [1858]
```

```
192,192
2930 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
2940 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
2950 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
2960 DATA 192,192,192,192,192,128,192,192, [1885]
212,252
2970 DATA 252,192,192,128,192,192,212,243, [2448]
246,192
2980 DATA 192,128,192,192,212,243,246,192, [2400]
192,128
2990 DATA 192,192,212,243,246,192,192,128, [2392]
208,252
3000 DATA 252,252,252,252,232,128,212,252, [1736]
252,252
3010 DATA 252,252,72,128,212,252,252,252,2 [1829]
52,252
3020 DATA 232,128,212,168,252,252,252,84,2 [1705]
32,128
3030 DATA 192,128,192,192,192,64,192,128,1 [2099]
92,192
3040 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
3050 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
3060 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
3070 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
3080 DATA 192,192,192,192,192,128,0,0,0,0  [1544]
3090 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
3100 DATA 0,0,192,192,192,192,192,192,192, [2242]
128
3110 DATA 192,192,192,192,192,192,192,128, [2143]
192,132
3120 DATA 12,12,12,12,192,128,192,132,12,1 [1859]
2
3130 DATA 12,12,192,128,192,129,192,192,19 [1593]
2,129
3140 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
3150 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
3160 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
3170 DATA 192,132,192,128,192,132,192,192, [983]
192,132
3180 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
3190 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
3200 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
3210 DATA 192,132,192,128,192,129,3,3,3,3  [1087]
3220 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
3230 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
3240 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
3250 DATA 192,132,192,128,192,132,192,192, [983]
192,132
3260 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
3270 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
3280 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
3290 DATA 192,132,192,128,192,132,192,192, [983]
192,132
3300 DATA 192,128,192,129,192,192,192,129, [2266]
192,128
3310 DATA 192,132,12,12,12,12,192,128,192, [1909]
132
3320 DATA 12,12,12,12,192,128,192,192,192, [2066]
```

```
192
3330 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
3340 DATA 192,128,0,0,0,0,0,0,0,0          [1248]
3350 DATA 0,0,0,0,0,0,0,0,192,192          [1582]
3360 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
3370 DATA 192,192,192,128,192,129,3,3,3,3  [1390]
3380 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
3390 DATA 192,129,192,192,192,129,192,128, [2044]
192,129
3400 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3410 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3420 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3430 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3440 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3450 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3460 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
3470 DATA 192,129,3,3,3,3,192,128,192,129  [1708]
3480 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3490 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3500 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3510 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3520 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3530 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3540 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3550 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3560 DATA 192,192,192,129,192,128,192,129, [1769]
3,3
3570 DATA 3,3,192,128,192,129,3,3,3,3      [1074]
3580 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
3590 DATA 192,192,192,192,192,192,192,128, [1725]
0,0
3600 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
3610 DATA 0,0,0,0,192,192,192,192,192,192  [2262]
3620 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
3630 DATA 192,132,12,12,12,12,192,128,192, [1909]
132
3640 DATA 12,12,12,12,192,128,192,129,192, [2038]
192
3650 DATA 192,129,192,128,192,129,192,192, [1773]
192,132
3660 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3670 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3680 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3690 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3700 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3710 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3720 DATA 12,12,12,9,192,128,192,129,12,12 [1906]
3730 DATA 12,9,192,128,192,132,192,192,192 [1977]
,129
3740 DATA 192,128,192,132,192,192,192,129, [1511]
192,128
3750 DATA 192,132,192,192,192,129,192,128, [1613]
192,132
3760 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
3770 DATA 192,129,192,128,192,132,192,192, [1887]
192,129
3780 DATA 192,128,192,132,192,192,192,129, [1511]
192,128
3790 DATA 192,132,192,192,192,129,192,128, [1613]
192,132
3800 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
3810 DATA 192,129,192,128,192,129,192,192, [1740]
192,129
3820 DATA 192,128,192,132,12,12,12,12,192, [1590]
128
3830 DATA 192,132,12,12,12,12,192,128,192, [1905]
192
3840 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
3850 DATA 192,192,192,128,0,0,0,0,0,0      [1125]
3860 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
3870 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
3880 DATA 192,192,192,192,192,128,192,132, [1575]
12,12
3890 DATA 12,12,192,128,192,132,12,12,12,1 [1210]
2
3900 DATA 192,128,192,129,192,192,192,129, [2266]
192,128
3910 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3920 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3930 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
3940 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
3950 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
3960 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
3970 DATA 192,132,192,128,192,129,12,12,12 [1713]
,9
3980 DATA 192,128,192,129,12,12,12,9,192,1 [2209]
28
3990 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4000 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
4010 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4020 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4030 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4040 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
4050 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4060 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4070 DATA 192,129,192,192,192,129,192,128, [2045]
192,132
4080 DATA 12,12,12,12,192,128,192,132,12,1 [1859]
2
4090 DATA 12,12,192,128,192,192,192,192,19 [1832]
2,192
4100 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
4110 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
4120 DATA 0,0,0,0,0,0,0,192,192,192,192    [1548]
4130 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
4140 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
4150 DATA 192,129,3,3,3,3,192,128,192,129  [1708]
4160 DATA 192,192,192,129,192,128,192,132, [2192]
```

```
192,192
4170 DATA 192,132,192,128,192,132,192,192, [983]
192,132
4180 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
4190 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
4200 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
4210 DATA 192,132,192,128,192,132,192,192, [983]
192,132
4220 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
4230 DATA 192,129,12,12,12,9,192,128,192,1 [2079]
29
4240 DATA 12,12,12,9,192,128,192,129,192,1 [2138]
92
4250 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4260 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4270 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4280 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
4290 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4300 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4310 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4320 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
4330 DATA 192,129,192,128,192,129,3,3,3,3  [1556]
4340 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
4350 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
4360 DATA 192,192,192,192,192,128,0,0,0,0  [1544]
4370 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
4380 DATA 0,0,192,192,192,192,192,192,192, [2242]
128
4390 DATA 192,192,192,192,192,192,192,128, [2143]
192,132
4400 DATA 12,12,12,12,192,128,192,132,12,1 [1859]
2
4410 DATA 12,12,192,128,192,129,192,192,19 [1593]
2,129
4420 DATA 192,128,192,132,192,192,192,129, [1511]
192,128
4430 DATA 192,132,192,192,192,129,192,128, [1613]
192,132
4440 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
4450 DATA 192,129,192,128,192,132,192,192, [1887]
192,129
4460 DATA 192,128,192,132,192,192,192,129, [1511]
192,128
4470 DATA 192,132,192,192,192,129,192,128, [1613]
192,132
4480 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
4490 DATA 192,129,192,128,192,129,12,12,12 [1646]
,9
4500 DATA 192,128,192,129,12,12,12,9,192,1 [2209]
28
4510 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4520 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
4530 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4540 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4550 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4560 DATA 192,192,192,132,192,128,192,129, [1831]
```

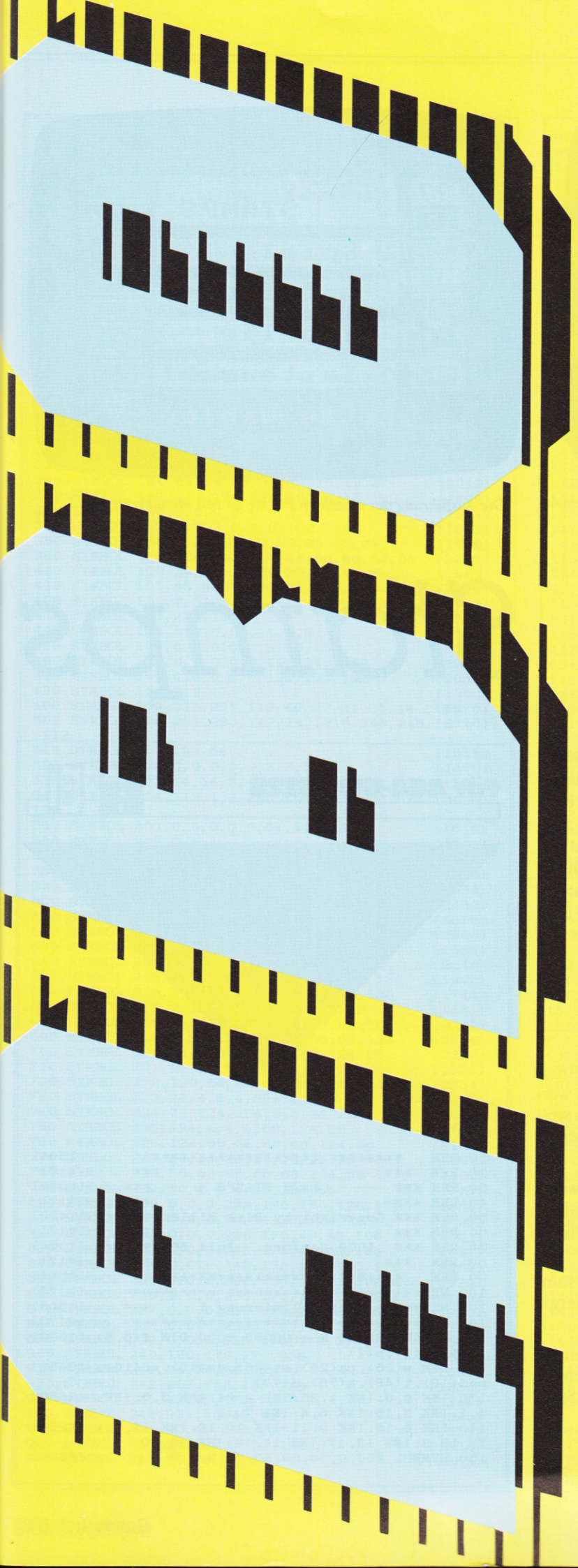
```
192,192
4570 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4580 DATA 192,128,192,129,192,192,192,129, [2266]
192,128
4590 DATA 192,132,12,12,12,12,192,128,192, [1909]
132
4600 DATA 12,12,12,12,192,128,192,192,192, [2066]
192
4610 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
4620 DATA 192,128,0,0,0,0,0,0,0,0          [1248]
4630 DATA 0,0,0,0,0,0,0,0,192,192          [1582]
4640 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
4650 DATA 192,192,192,128,192,132,12,12,12 [1418]
,12
4660 DATA 192,128,192,132,12,12,12,12,192, [1590]
128
4670 DATA 192,129,192,192,192,129,192,128, [2045]
192,132
4680 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
4690 DATA 192,129,192,128,192,132,192,192, [1887]
192,129
4700 DATA 192,128,192,132,192,192,192,129, [1511]
192,128
4710 DATA 192,132,192,192,192,129,192,128, [1613]
192,132
4720 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
4730 DATA 192,129,192,128,192,132,192,192, [1887]
192,129
4740 DATA 192,128,192,129,12,12,12,9,192,1 [2209]
28
4750 DATA 192,129,12,12,12,9,192,128,192,1 [2096]
32
4760 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
4770 DATA 192,132,192,128,192,132,192,192, [983]
192,132
4780 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
4790 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
4800 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
4810 DATA 192,132,192,128,192,132,192,192, [983]
192,132
4820 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
4830 DATA 192,132,192,192,192,132,192,128, [2014]
192,129
4840 DATA 192,192,192,129,192,128,192,132, [1781]
12,12
4850 DATA 12,12,192,128,192,132,12,12,12,1 [1210]
2
4860 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
4870 DATA 192,192,192,192,192,192,192,128, [1725]
0,0
4880 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
4890 DATA 0,0,0,0,192,192,192,192,192,192  [2262]
4900 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
4910 DATA 192,132,12,12,12,12,192,128,192, [1909]
132
4920 DATA 12,12,12,12,192,128,192,129,192, [2038]
192
4930 DATA 192,129,192,128,192,129,192,192, [1773]
192,132
4940 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4950 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
4960 DATA 192,192,192,132,192,128,192,129, [1831]
```

```
192,192
4970 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
4980 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
4990 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
5000 DATA 3,3,3,3,192,128,192,129,3,3      [1282]
5010 DATA 3,3,192,128,192,129,192,192,192, [1223]
132
5020 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
5030 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
5040 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
5050 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
5060 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
5070 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
5080 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
5090 DATA 192,132,192,128,192,129,192,192, [1686]
192,129
5100 DATA 192,128,192,129,3,3,3,3,192,128  [1805]
5110 DATA 192,129,3,3,3,3,192,128,192,192  [1657]
5120 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
5130 DATA 192,192,192,128,0,0,0,0,0,0      [1125]
5140 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
5150 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
5160 DATA 192,192,192,192,192,128,192,132, [1575]
12,12
5170 DATA 12,12,192,128,192,132,12,12,12,1 [1210]
2
5180 DATA 192,128,192,129,192,192,192,129, [2266]
192,128
5190 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
5200 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
5210 DATA 192,132,192,128,192,132,192,192, [983]
192,132
5220 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
5230 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
5240 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
5250 DATA 192,132,192,128,192,129,12,12,12 [1713]
,9
5260 DATA 192,128,192,129,12,12,12,9,192,1 [2209]
28
5270 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
5280 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
5290 DATA 192,132,192,128,192,132,192,192, [983]
192,132
5300 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
5310 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
5320 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
5330 DATA 192,132,192,128,192,132,192,192, [983]
192,132
5340 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
5350 DATA 192,129,192,192,192,129,192,128, [2045]
192,132
5360 DATA 12,12,12,12,192,128,192,132,12,1 [1859]
2
5370 DATA 12,12,192,128,192,192,192,192,19 [1832]
2,192
5380 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
5390 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
5400 DATA 0,0,0,0,0,0,192,192,192,192      [1548]
5410 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
5420 DATA 192,128,192,132,12,12,12,12,192, [1590]
128
5430 DATA 192,132,12,12,12,12,192,128,192, [1940]
129
5440 DATA 192,192,192,129,192,128,192,132, [2192]
192,192
5450 DATA 192,132,192,128,192,132,192,192, [983]
192,132
5460 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
5470 DATA 192,132,192,192,192,132,192,128, [2027]
192,132
5480 DATA 192,192,192,132,192,128,192,132, [1964]
192,192
5490 DATA 192,132,192,128,192,132,192,192, [983]
192,132
5500 DATA 192,128,192,132,192,192,192,132, [1781]
192,128
5510 DATA 192,129,12,12,12,9,192,128,192,1 [2079]
29
5520 DATA 12,12,12,9,192,128,192,129,192,1 [2138]
92
5530 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
5540 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
5550 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
5560 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
5570 DATA 192,132,192,128,192,129,192,192, [1667]
192,132
5580 DATA 192,128,192,129,192,192,192,132, [2436]
192,128
5590 DATA 192,129,192,192,192,132,192,128, [2171]
192,129
5600 DATA 192,192,192,132,192,128,192,129, [1831]
192,192
5610 DATA 192,129,192,128,192,132,12,12,12 [1553]
,12
5620 DATA 192,128,192,132,12,12,12,12,192, [1590]
128
5630 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
5640 DATA 192,192,192,192,192,128,0,0,0,0  [1544]
5650 DATA 0,0,0,0,0,0,0,0,0,0              [1269]
5660 DATA 0,0,243,243,243,243,243,243,243, [1900]
162
5670 DATA 243,243,243,243,243,243,243,162, [1608]
243,243
5680 DATA 243,243,243,243,243,162,243,243, [1595]
243,243
5690 DATA 243,243,243,162,243,243,243,243, [2069]
243,243
5700 DATA 243,162,243,243,243,243,243,243, [2342]
243,162
5710 DATA 243,243,243,243,243,243,243,162, [1608]
243,243
5720 DATA 243,243,162,243,243,162,243,243, [1459]
243,243
5730 DATA 162,243,243,162,243,243,60,121,2 [2016]
30,243
5740 DATA 243,162,243,243,30,121,162,243,2 [1790]
43,162
5750 DATA 243,243,60,121,162,243,243,162,2 [1790]
43,182
5760 DATA 60,121,162,243,243,162,195,195,1 [2095]
95,195
5770 DATA 195,195,195,162,195,195,195,195, [1353]
195,195
```

```
5780 DATA 195,162,195,147,195,195,195,195, [2131]
195,162
5790 DATA 227,195,195,195,195,195,195,162, [2052]
252,195
5800 DATA 195,195,195,195,195,168,252,233, [1747]
195,195
5810 DATA 195,195,195,168,252,252,195,195, [1559]
195,195
5820 DATA 195,168,252,252,252,252,252,252, [1874]
252,168
5830 DATA 252,252,252,252,252,252,252,168, [1982]
252,252
5840 DATA 252,252,252,252,252,168,252,252, [2085]
252,252
5850 DATA 252,252,252,168,252,252,252,252, [1785]
252,252
5860 DATA 252,168,252,252,252,252,252,252, [1890]
252,168
5870 DATA 252,252,252,252,252,252,252,168, [1982]
252,252
5880 DATA 252,252,252,252,252,168,252,252, [2085]
252,252
5890 DATA 252,252,252,168,252,252,252,252, [1785]
252,252
5900 DATA 252,168,0,0,0,0,0,0,0,0           [623]
5910 DATA 0,0,0,0,0,0,0,0,192,192           [1582]
5920 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
5930 DATA 192,192,192,128,149,46,63,46,63, [1673]
46
5940 DATA 63,128,149,46,63,46,63,46,63,128 [1005]
5950 DATA 149,63,63,46,63,63,63,128,149,63 [1166]
5960 DATA 63,126,63,63,63,128,132,63,63,12 [2114]
6
5970 DATA 63,63,46,128,132,63,63,126,63,63 [1484]
5980 DATA 46,128,149,63,63,126,63,63,63,12 [2087]
8
5990 DATA 149,63,63,126,63,63,63,128,149,6 [1255]
3
6000 DATA 63,126,63,63,63,128,149,63,63,12 [2262]
6
6010 DATA 63,63,63,128,149,63,63,126,63,63 [1941]
6020 DATA 63,128,149,63,63,126,63,63,63,12 [1432]
8
6030 DATA 132,29,63,126,252,189,12,128,132 [1853]
,29
6040 DATA 63,126,252,189,12,128,149,63,63, [1311]
63
6050 DATA 63,63,63,128,149,63,63,63,63,63  [1802]
6060 DATA 63,128,149,63,63,63,63,63,63,128 [1425]
6070 DATA 149,63,63,63,63,63,63,128,149,63 [1267]
6080 DATA 63,63,63,63,63,128,149,63,63,63  [959]
6090 DATA 63,63,63,128,132,63,63,63,63,63  [1680]
6100 DATA 46,128,132,63,63,63,63,63,46,128 [1888]
6110 DATA 149,63,63,46,63,63,63,128,149,63 [1166]
6120 DATA 63,46,63,63,63,128,149,46,63,46  [1713]
6130 DATA 63,46,63,128,149,46,63,46,63,46  [1363]
6140 DATA 63,128,192,192,192,192,192,192,1 [1287]
92,128
6150 DATA 192,192,192,192,192,192,192,128, [1725]
0,0
6160 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
6170 DATA 0,0,0,0,192,192,192,192,192,192   [2262]
6180 DATA 192,128,192,192,192,192,192,192, [2288]
192,128
6190 DATA 192,192,192,192,192,192,192,128, [2187]
192,192
6200 DATA 192,192,192,192,192,128,192,192, [1858]
192,192
6210 DATA 192,192,192,128,192,192,192,192, [1267]
192,192
6220 DATA 192,128,192,132,72,192,192,12,19 [1873]
2,128
6230 DATA 192,12,12,192,132,12,72,128,192, [1791]
12
6240 DATA 12,72,12,12,72,128,132,12,12,12   [1519]
```

```
6250 DATA 12,12,12,128,132,12,12,12,12,12   [1210]
6260 DATA 12,128,12,12,12,12,12,12,12,8     [1552]
6270 DATA 12,12,12,12,12,12,12,8,12,12      [1589]
6280 DATA 12,12,12,12,12,8,12,12,12,12      [1886]
6290 DATA 12,12,12,8,12,12,12,12,12,12      [1041]
6300 DATA 12,8,12,12,12,12,12,12,12,8       [1262]
6310 DATA 12,12,12,12,12,12,12,8,132,12     [1743]
6320 DATA 12,12,12,12,12,128,132,12,12,12   [1660]
6330 DATA 12,12,12,128,192,12,12,12,12,12   [1670]
6340 DATA 72,128,192,12,12,12,12,12,72,128  [1704]
6350 DATA 192,132,12,12,12,12,192,128,192,  [1909]
132
6360 DATA 12,12,12,12,192,128,192,192,12,1  [1864]
2
6370 DATA 12,72,192,128,192,192,132,12,12,  [1671]
192
6380 DATA 192,128,192,192,132,12,12,192,19  [2190]
2,128
6390 DATA 192,192,192,12,72,192,192,128,19  [1835]
2,192
6400 DATA 192,132,192,192,192,128,192,192,  [2242]
192,192
6410 DATA 192,192,192,128,0,0,0,0,0,0       [1125]
6420 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
6430 DATA 252,252,252,252,252,252,252,168,  [1982]
252,252
6440 DATA 252,252,252,252,252,168,243,243,  [1885]
243,243
6450 DATA 243,243,243,162,243,243,243,243,  [2069]
243,243
6460 DATA 243,162,195,195,195,195,195,195,  [1699]
195,130
6470 DATA 195,195,195,195,195,195,195,130,  [1702]
48,48
6480 DATA 48,48,48,48,48,32,48,48,48,48     [1579]
6490 DATA 48,48,48,32,63,63,63,63,63,63     [1647]
6500 DATA 63,42,63,63,63,63,63,63,63,42     [1387]
6510 DATA 15,15,15,15,15,15,15,10,15,15     [1103]
6520 DATA 15,15,15,15,15,10,240,240,240,24  [1011]
0
6530 DATA 240,240,240,160,240,240,240,240,  [1786]
240,240
6540 DATA 240,160,51,51,51,51,51,51,51,34   [1843]
6550 DATA 51,51,51,51,51,51,51,34,3,3       [752]
6560 DATA 3,3,3,3,3,2,3,3,3,3               [1015]
6570 DATA 3,3,3,2,12,12,12,12,12,12         [1017]
6580 DATA 12,8,12,12,12,12,12,12,12,8       [1262]
6590 DATA 204,204,204,204,204,204,204,136,  [1915]
204,204
6600 DATA 204,204,204,204,204,136,207,207,  [1970]
207,207
6610 DATA 207,207,207,138,207,207,207,207,  [1960]
207,207
6620 DATA 207,138,192,192,192,192,192,192,  [2326]
192,128
6630 DATA 192,192,192,192,192,192,192,128,  [2132]
60,60
6640 DATA 60,60,60,60,60,40,60,60,60,60     [1478]
6650 DATA 60,60,60,40,0,0,0,0,0,0           [1050]
6660 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
6670 DATA 0,0,0,0,0,0,0,0,0,0               [1269]
6680 DATA 0,0,0,0,0,0                       [826]
```

```
10 MEMORY &A027:FOR n=&A028 TO &A052:READ   [2956]
a:POKE n,a:NEXT
20 SAVE"mm",b,&A028,&2B                     [944]
30 DATA 221,94,0,221,86,1,221,110,2,221     [1877]
40 DATA 102,3,6,4,197,229,6,8,197,6         [1589]
50 DATA 8,26,119,19,35,16,250,1,248,7       [1430]
60 DATA 9,193,16,240,225,1,80,0,9,193       [1083]
70 DATA 16,228,201                          [486]
```



Sichern Sie sich Ihre »PC Schneider International« Ausgabe für Ausgabe mit einem zusätzlichen Preisvorteil.

**Abonnieren Sie jetzt! Top-Information für Ihren Schneider Computer.**

**Natürlich vom führenden Schneider Magazin.**

**PC Schneider International kostet im Abonnement:**

| | | |
|---|---|---|
| Im Inland und Westberlin: | für 1/2 Jahr | 30,– DM |
| | für 1 Jahr | 60,– DM |
| Im europäischen Ausland: | für 1/2 Jahr | 45,– DM |
| | für 1 Jahr | 90,– DM |
| Im außereurop. Ausland: | für 1/2 Jahr | 60,– DM |
| | für 1 Jahr | 120,– DM |

**BITTE BENUTZEN SIE DIE BESTELLKARTE**

DMV GmbH · PC Schneider International
Postfach 250 · 3440 Eschwege · Tel. (0 56 51) 87 02

DMV Daten- und Medienverlag




*Hier sollte strategisch vorgegangen werden, da man sonst in arge Schwierigkeiten kommen kann.*




*Die Ausführung der einzelnen Punkte ist voll Menuegesteuert.*

# Crazy Stamps

**Hier kommt eines der verrücktesten Programme mit einer noch wilderen Geschichte die man je gesehen und gelesen hat. Aber das soll nicht heißen, daß das Programm verrückt ist, im Gegenteil, bei diesem Spiel kommt soviel Spaß auf, daß es schon richtig verrückt ist. Die Geschichte beginnt damit, was eigentlich normal ist, daß man kein Geld mehr in der Tasche hat. Was tun? In diesem Falle ganz einfach: Sie gehen jobben, in einem Postamt, wo Sie die eingehende Post AUSREICHEND frankieren müssen, um es sich dann von der Post abstempeln zu lassen.**

## Aufgabe:

Sie müssen sich einen geschickten Weg durch die Welt der Briefmarken bahnen. Sollte man denken, das wäre einfach dann werden Sie sich ganz schön wundern. Ihnen wird vorgegeben welchen Wert das gute Poststück haben muß, um es durchgehen zu lassen.

Zum Aufkleben der Briefmarken wird ein »Rahmen« bewegt, der durch den Druck des Feuerknopfes die Frankierung auslöst. Es kann aber auch, eine der zu definierenden Tasten benutzt werden.

Ein Weg der eingeschlagen wurde, kann erst nach einer Runde verlassen werden. Gearbeitet wird, solange die Energie reicht, aber das wird Ihnen verdammt schwer gemacht.

Die Schwierigkeiten fangen an, von sich ändernden Werten auf Briefmarken über schwarze Löcher und Totenköpfe. Lassen Sie sich begeistern von und mit Crazy Stamps. Eine genaue Spielanleitung ist im Programm enthalten.

Viel Spaß mit Crazy Stamps.

(D. Müller/CD)

**für 464-664-6128**

```
10 REM   ****************************      [1504]
20 REM  ***                          ***     [417]
30 REM *** ----- CRAZY STAMPS ----- ***     [2040]
40 REM ***                           ***     [417]
50 REM *** Copyright by Dirk Mller ***     [2429]
60 REM ***                           ***     [417]
70 REM ***  4900 Herford,  Juli 86   ***     [1364]
80 REM  ***                          ***     [417]
90 REM   ****************************      [1504]
100 REM================================     [2173]
101 REM===     Initialisierung 1      ===    [1308]
102 REM================================     [2173]
110 CLEAR:DEFINT a-z:GOSUB 6000:DIM f(6,5) [1359]
,r(5),k(6),s(7)
120 DIM w(26),pa(26),st(26),dm(10),gr(10), [3458]
na$(10),t(48),h(78),d(78)
130 INK 0,0:INK 1,26:INK 2,24:INK 3,6:INK  [2685]
4,1:INK 5,15:INK 6,4:INK 7,12
140 INK 8,16:INK 9,11:INK 10,10:INK 11,9:I [4268]
NK 12,3:INK 13,17:INK 14,19:INK 15,20
150 SYMBOL 250,0,30,30,51,51,3,3,6         [1637]
```

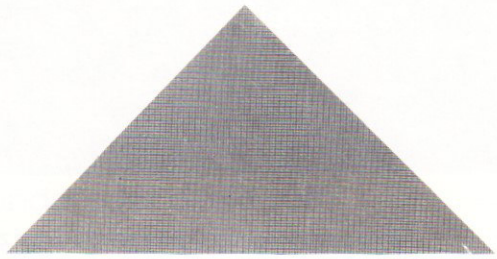
```
160 SYMBOL 251,6,12,12,0,0,12,12                [2064]
170 SYMBOL 252,0,60,60,102,102,6,6,12           [2033]
180 SYMBOL 253,12,24,24,0,0,24,24               [1281]
190 SYMBOL 254,1,1,1,1,1,1,1,1:SYMBOL 255,      [5361]
128,128,128,128,128,128,128,128:SYMBOL 169
,170,85,170,85,170,85,170,85
200 SYMBOL 170,0,0,0,0,0,5,7,3                  [1554]
210 SYMBOL 171,0,0,0,0,0,85,255,255             [1831]
220 SYMBOL 172,0,0,0,0,0,64,192,128             [1663]
230 SYMBOL 173,6,2,6,2,6,2,6,2                  [1305]
240 SYMBOL 174,192,128,192,128,192,128,192      [2675]
,128
250 SYMBOL 175,7,7,5                            [1244]
260 SYMBOL 176,255,255,85                       [1457]
270 SYMBOL 177,192,192,64                       [1187]
280 SYMBOL 178,3,12,16,19,36,40,67,68           [2045]
290 SYMBOL 179,192,48,8,200,36,20,194,34        [2375]
300 SYMBOL 180,136,143,152,144,144,144,151      [2272]
,151
310 SYMBOL 181,17,241,25,9,9,9,233,233          [2181]
320 SYMBOL 182,151,144,144,144,144,152,143      [2537]
,128
330 SYMBOL 183,233,9,9,9,9,25,241,1             [1510]
340 SYMBOL 184,64,64,35,39,19,16,12,3           [2012]
350 SYMBOL 185,2,2,196,228,200,8,48,192         [1700]
360 SYMBOL 186,0,0,0,0,0,0,3,7                  [1544]
370 SYMBOL 187,0,0,0,0,0,0,192,224              [1883]
380 SYMBOL 188,7,15,15,29,29,25,25,29           [1590]
390 SYMBOL 189,224,240,240,24,88,88,88,88       [2306]
400 SYMBOL 190,29,29,29,29,29,15,15,7           [1925]
410 SYMBOL 191,88,88,88,88,24,240,240,224       [2304]
420 SYMBOL 192,7,3                              [1413]
430 SYMBOL 193,224,192                          [1269]
440 SYMBOL 194,0,0,0,0,0,1,3,3                  [1532]
450 SYMBOL 195,0,0,0,0,0,3,15,159               [2309]
460 SYMBOL 196,0,0,0,0,0,192,240,249            [1928]
470 SYMBOL 197,0,0,0,0,0,128,192,192            [2057]
480 SYMBOL 198,3,2                              [1300]
490 SYMBOL 199,223,251,113,49,27,31,31,14       [1970]
500 SYMBOL 200,251,223,142,140,216,248,248      [2770]
,112
510 SYMBOL 201,192,64                           [1081]
520 SYMBOL 202,0,0,0,0,0,2,3,3                  [1476]
530 SYMBOL 203,14,14,31,63,119,230,194,131      [1971]
540 SYMBOL 204,112,112,248,252,238,103,67,      [3056]
193
550 SYMBOL 205,0,0,0,0,0,64,192,192             [1616]
560 SYMBOL 206,3,3,1                            [1101]
570 SYMBOL 207,3,3,1                            [1165]
580 SYMBOL 208,192,192,128                      [1366]
590 SYMBOL 209,192,192,128                      [1554]
600 SYMBOL 210,0,56,124,68,68,76,76,84          [2430]
610 SYMBOL 211,0,16,16,48,48,16,16,16           [1678]
620 SYMBOL 212,0,56,124,68,4,4,4,60             [2340]
630 SYMBOL 213,0,56,124,68,4,4,4,24             [1886]
640 SYMBOL 214,0,8,8,24,24,40,40,72             [2240]
650 SYMBOL 215,0,124,124,68,64,64,64,120        [1902]
660 SYMBOL 216,0,56,124,68,64,64,64,120         [1744]
670 SYMBOL 217,0,124,124,68,4,4,4,12            [1881]
680 SYMBOL 218,0,56,124,68,68,68,68,56          [2312]
690 SYMBOL 219,0,56,124,68,68,68,68,124         [2964]
700 SYMBOL 220,84,100,100,68,68,124,56          [2340]
710 SYMBOL 221,16,16,16,16,16,56,56             [1621]
720 SYMBOL 222,120,64,64,64,68,124,124          [2534]
730 SYMBOL 223,24,4,4,4,68,124,56               [2000]
740 SYMBOL 224,72,124,124,8,8,28,28             [1745]
750 SYMBOL 225,124,4,4,4,68,124,56              [1740]
760 SYMBOL 226,124,68,68,68,68,124,56           [2130]
770 SYMBOL 227,24,16,16,16,16,16,16             [1362]
780 SYMBOL 228,56,68,68,68,68,124,56            [1466]
790 SYMBOL 229,60,4,4,4,68,124,56               [1779]
800 SYMBOL 230,0,0,0,238,170,170,170,170        [1889]
810 SYMBOL 231,0,0,0,78,74,202,74,74            [1895]
820 SYMBOL 232,0,0,0,238,170,42,42,106          [2061]
830 SYMBOL 233,0,0,0,238,170,42,42,106          [2037]
840 SYMBOL 234,0,0,0,142,138,138,170,170        [2359]
850 SYMBOL 235,0,0,0,238,138,138,138,234        [2124]
860 SYMBOL 236,0,0,0,238,170,138,138,234        [1848]
870 SYMBOL 237,0,0,0,238,170,42,42,106          [1990]
880 SYMBOL 238,0,0,0,238,170,170,170,234        [2703]
890 SYMBOL 239,0,0,0,238,170,170,170,234        [2664]
900 SYMBOL 240,170,170,170,238                  [1831]
910 SYMBOL 241,74,74,74,238                     [1684]
920 SYMBOL 242,202,138,138,238                  [1839]
930 SYMBOL 243,42,42,170,238                    [1344]
940 SYMBOL 244,234,42,42,46                     [1240]
950 SYMBOL 245,42,42,170,238                    [1678]
960 SYMBOL 246,170,170,170,238                  [1834]
970 SYMBOL 247,74,74,74,78                      [1652]
980 SYMBOL 248,170,170,170,238                  [1649]
990 SYMBOL 249,42,42,170,238                    [1679]
1000 REM==============================          [2077]
1001 REM===     Initialisierung 2     ===       [1390]
1002 REM==============================          [2077]
1010 RESTORE 1000:MODE 1:BORDER 0:PAPER 0:      [1687]
PEN 1
1020 FOR q=1 TO 48:READ t(q):NEXT q             [2119]
1030 FOR q=1 TO 26:READ w(q):w(q)=w(q)*10:      [2982]
NEXT q
1040 FOR q=1 TO 26:READ pa(q),st(q):st(q)=      [3252]
3*st(q)+12:NEXT q
1050 DATA 478,451,426,402,379,358,338,319,      [2887]
301,284,268,253
1060 DATA 239,225,213,201,190,179,169,159,      [2203]
150,142,134,127
1070 DATA 119,113,106,100,95,89,84,80,75,7      [2120]
1,67,63
1080 DATA 60,56,53,50,47,45,42,40,38,36,34      [1326]
,32
1090 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14      [3170]
,15,16,17,18,19,20,25,30,40,50,100,0
1100 DATA 4,1,5,1,6,1,7,1,8,,9,1,10,1,11,1      [649]
,12,1,13,
1110 DATA 4,1,5,1,6,1,7,1,8,,9,1,10,1,11,1      [1377]
,12,1,13,,15,,14,,14,,15,,,2,,1
1120 FOR q=1 TO 10:na$(q)="PHIL PHILATELY"      [4343]
:dm(q)=220-20*q:gr(q)=0:mz(q)=0
1130 NEXT q                                     [367]
1140 POKE &BDEE,201:POKE &BDEF,0:POKE &BDF      [1818]
0,0
1150 an$=CHR$(22)+CHR$(1):aus$=CHR$(22)+CH      [2540]
R$(0)
1160 r(1)=0:r(2)=1:r(3)=2:r(4)=8:r(5)=9         [1725]
1170 IF PEEK(&BB01)=&E0 THEN ad!=PEEK(&B29      [3427]
6)+PEEK(&B297)*256
1180 IF PEEK(&BB01)=&5C THEN ad!=PEEK(&B73      [3012]
6)+PEEK(&B737)*256
1190 IF PEEK(&BB01)=&5C THEN PRINT"CPC 664      [4483]
/6128"ELSE PRINT"CPC 464"
1200 REM==============================          [2077]
1201 REM===       Melodie-Datas       ===       [1715]
1202 REM==============================          [2077]
1210 FOR q=1 TO 78:READ h(q),d(q):NEXT q        [2090]
1220 DATA 1,2,6,2,10,2,13,4,10,2,6,2,1,4,1      [2266]
3,2,10,2,11,2,8,6
1230 DATA 0,2,1,2,5,2,8,2,11,4,8,2,5,2,1,4      [3311]
,11,2,8,2, 10,2,6,6
1240 DATA 1,2,6,2,10,2,13,4,10,2,6,2,1,4,1      [2266]
3,2,10,2,11,2,8,6
1250 DATA 0,2,1,2,5,2,8,2,11,4,8,2,5,2,1,4      [3448]
,11,2,8,2, 10,2,6,4,1,2
1260 DATA 6,1,6,1,6,2,5,2,3,2,5,2,5,4,1,2       [1291]
1270 DATA 5,1,5,1,5,2,3,2,5,2, 6,4,0,2,1,2      [1407]
1280 DATA 6,1,6,1,6,2,5,2,3,2,5,2,5,4,1,2       [1291]
1290 DATA 5,1,5,1,5,2,3,2,5,2, 6,6,0,4          [845]
1300 REM==============================          [2077]
1301 REM===          Titel 1          ===       [974]
1302 REM==============================          [2077]
1310 MODE 0:BORDER 9:PAPER 11:PEN 1:CLS:IN      [3904]
K 0,0:INK 1,9:INK 2,24:INK 3,6
1320 SYMBOL 167,0,0,0,0,85,85,255,255           [1726]
1330 SYMBOL 168,255,255,85,85                   [1768]
1340 FOR y=1 TO 22:c=143                        [1830]
1350 IF y=1 OR y=7 OR y=11 OR y=15 OR y=19      [2474]
 THEN c=167
1360 IF y=6 OR y=10 OR y=14 OR y=18 OR y=2      [2018]
2 THEN c=168
1370 LOCATE 5,y:PEN 1:PRINT STRING$(12,c);      [3857]
:NEXT y:INK 1,26
1380 LOCATE 6,2:PAPER 1:PEN 3:dat$="CRAZY"      [1607]
:GOSUB 4100
1390 LOCATE 10,4:PEN 3:dat$="STAMPS":GOSUB      [3167]
 4100
1400 REM==============================          [2077]
1401 REM===          Titel 2          ===       [998]
1402 REM==============================          [2077]
1410 LOCATE 5,8:PEN 5:dat$="1-":GOSUB 4100      [1435]
1420 LOCATE 7,8:PEN 9:dat$="Spiel":GOSUB 4      [2042]
100
1430 LOCATE 5,12:PEN 5:dat$="2-":GOSUB 410      [1579]
0
1440 LOCATE 7,12:PEN 9:dat$="Anleitung":GO      [3010]
SUB 4100
1450 LOCATE 5,16:PEN 5:dat$="3-":GOSUB 410      [1735]
0
1460 LOCATE 7,16:PEN 9:dat$="Tabelle":GOSU      [1843]
B 4100
1470 LOCATE 5,20:PEN 5:dat$="4-":GOSUB 410      [2539]
```

```
0
1480 LOCATE 7,20:PEN 9:dat$="Tastenwahl":G [2066]
OSUB 4100
1490 LOCATE 5,24:PAPER 11:PEN 2:dat$="Bitt [2735]
e w@hlen":GOSUB 4100
1500 REM================================    [2077]
1501 REM===          Menue 1           ===    [935]
1502 REM================================    [2077]
1510 WINDOW#1,1,4,1,25:WINDOW#2,17,20,1,25 [5977]
:PAPER#1,11:PAPER#2,11:CLS#1:CLS#2
1520 wa=VAL(INKEY$):IF wa<>0 THEN 1520 ELS [4102]
E u=23:GOSUB 4000
1530 wa=VAL(INKEY$)                         [1541]
1540 IF wa=1 THEN 1700                      [1458]
1550 IF wa=2 THEN 5000                      [880]
1560 IF wa=3 THEN 4700                      [1067]
1570 IF wa=4 THEN 4900                      [1009]
1580 IF INKEY(18)=0 THEN u=78:GOSUB 4000:C [2351]
LS#1:CLS#2
1590 IF CINT(RND(TIME)*1000)>995 THEN u=78 [3448]
:GOSUB 4000:CLS#1:CLS#2
1600 REM================================    [2077]
1601 REM===          Menue 2           ===    [736]
1602 REM================================    [2077]
1610 h=CINT(RND(TIME)*30):q=INT(RND(TIME)* [2852]
2)+1:pp=0
1620 IF h>0 AND h<25 THEN pp=pa(h):pe=st(h [2483]
):wert=w(h)
1630 IF q=1 THEN LOCATE#1,1,1:PRINT#1,STRI [2558]
NG$(4,11);:y=1
1640 IF q=2 THEN LOCATE#2,1,25:PRINT#2,STR [3563]
ING$(4,10);:y=22
1650 x=16*(q-1)+1                           [861]
1660 IF h=0 OR h>28 THEN 1530               [1945]
1670 IF h=25 THEN GOSUB 3200 ELSE IF h=26   [3507]
THEN GOSUB 3300
1680 IF h=27 THEN GOSUB 3000:GOSUB 3500 EL [3546]
SE IF h>0 AND h<25 THEN GOSUB 3000
1690 GOTO 1530                              [393]
1700 REM================================    [2077]
1701 REM===    Spielfeld-Fenster      ===    [2046]
1702 REM================================    [2077]
1710 IF INKEY$<>""THEN 1710 ELSE GOSUB 450 [2448]
0:MODE 0:BORDER 3:PAPER 0:PEN 1
1720 WINDOW#1,1,20,1,3:PAPER#1,12:CLS#1:WI [3818]
NDOW#2,1,20,4,9:PAPER#2,11:CLS#2
1730 FOR q=348 TO 350 STEP 2                [1147]
1740 PLOT 0,q,14:DRAWR 588,0:DRAWR 0,-92:D [3890]
RAWR-588,0:DRAWR 0,92
1750 NEXT q                                 [367]
1760 FOR q=4 TO 25                          [976]
1770 PAPER 0:PEN 4:LOCATE 19,q:PRINT an$;C [3897]
HR$(138)CHR$(143);aus$;:NEXT q
1780 FOR y=6 TO 342 STEP 16:PLOT 596,y,15: [2801]
DRAW 632,y,15:NEXT y
1790 FOR y=14 TO 334 STEP 16:PLOT 600,y,15 [2868]
:DRAW 628,y,15:NEXT y
1800 REM================================    [2077]
1801 REM===  Markenfeld erstellen    ===    [2171]
1802 REM================================    [2077]
1810 LOCATE 1,1:PAPER 12:PEN 1:dat$="CRAZY [4034]
":GOSUB 4100
1820 LOCATE 15,1:PAPER 12:PEN 1:dat$="STAM [2243]
PS":GOSUB 4100
1830 h=1:FOR y=1 TO 4:FOR x=1 TO 6:f(x,y)= [1472]
h:h=h+1:NEXT x,y
1840 h=20:FOR x=1 TO 5:f(x,5)=h:h=h+1:NEXT [2976]
 x:f(6,5)=10
1850 FOR y=1 TO 5:FOR x=1 TO 6              [1510]
1860 hx=INT(RND(TIME)*6)+1                  [1740]
1870 hy=INT(RND(TIME)*5)+1                  [1775]
1880 h=f(x,y):f(x,y)=f(hx,hy):f(hx,hy)=h    [2818]
1890 NEXT x,y                               [548]
1900 REM================================    [2077]
1901 REM===  Markenfeld ausdrucken   ===    [2780]
1902 REM================================    [2077]
1910 FOR q=1 TO 6:k(q)=-1:NEXT q            [992]
1920 dmark=0:grosch=0:en=42                 [2053]
1930 GOSUB 3700                             [1013]
1940 FOR hy=1 TO 5:FOR hx=1 TO 6            [1292]
1950 wf=f(hx,hy):wert=w(wf):pp=pa(wf):pe=s [5592]
t(wf):x=3*hx-2:y=3*hy+7
1960 IF f(hx,hy)<25 THEN GOSUB 3000         [1986]
1970 IF f(hx,hy)=25 THEN GOSUB 3200         [2641]
1980 IF f(hx,hy)=26 THEN GOSUB 3300         [2958]
1990 NEXT hx,hy                             [592]
2000 REM================================    [2077]
2001 REM===       Sucher-Start        ===    [1389]
2002 REM================================    [2077]
2010 sx=INT(RND(TIME)*6)+1                  [923]
2020 sy=INT(RND(TIME)*5)+1                  [827]
2030 IF f(sx,sy)>24 THEN 2010               [1761]
2040 m=0:x=3*sx-2:y=3*sy+7:GOSUB 3900       [2274]
2050 REM================================    [2077]
2051 REM===       Wert vorgeben       ===    [1324]
2052 REM================================    [2077]
2060 m=m+1                                  [200]
2070 mwert=w(INT(RND(TIME)*24)+1)           [2499]
2080 mx=3*m-2                               [391]
2090 GOSUB 4200                             [883]
2100 REM================================    [2077]
2101 REM===     Tastaturabfrage      ===    [1150]
2102 REM================================    [2077]
2110 nx=sx:ny=sy                            [1143]
2120 wx=nx:wy=ny                            [955]
2130 IF INKEY(76)=0 OR INKEY(r(5))=0 THEN   [323]
2500
2140 IF INKEY(72)=0 OR INKEY(r(1))=0 THEN   [1780]
wy=wy-1
2150 IF INKEY(74)=0 OR INKEY(r(4))=0 THEN   [2602]
wx=wx-1
2160 IF INKEY(73)=0 OR INKEY(r(3))=0 THEN   [2495]
wy=wy+1
2170 IF INKEY(75)=0 OR INKEY(r(2))=0 THEN   [964]
wx=wx+1
2180 IF INKEY(66)>-1 THEN 2900 ELSE IF wx=  [3925]
nx AND wy=ny THEN 2130
2190 IF wx<>nx AND wy<>ny THEN 2120         [2333]
2200 REM================================    [2077]
2201 REM===    Sucher Kontrolle      ===    [2728]
2202 REM================================    [2077]
2210 IF wx<1 OR wx>6 THEN 2120              [1554]
2220 IF wy<1 OR wy>5 THEN 2120              [1208]
2230 IF wx<>sx AND wy<>sy THEN 2120         [1591]
2240 x=3*nx-2:y=3*ny+7:GOSUB 3950           [2278]
2250 nx=wx:ny=wy:x=3*nx-2:y=3*ny+7:GOSUB 3 [4758]
900
2260 IF f(nx,ny)>0 AND f(nx,ny)<26 THEN 21  [2245]
20
2270 IF f(nx,ny)<0 THEN 2300                [2209]
2280 IF f(nx,ny)=0 THEN sx=nx:sy=ny:GOTO 2  [2953]
450
2290 IF f(nx,ny)>25 THEN sx=nx:sy=ny:GOTO   [2819]
2400
2300 REM================================    [2077]
2301 REM===      Auf CRAZY STAMP      ===    [2054]
2302 REM================================    [2077]
2310 FOR q=500 TO 50 STEP-50:SOUND 1,q,3,7 [3909]
:NEXT q:wf=1+(22-f(nx,ny))MOD 24
2320 wert=w(wf):pp=pa(wf):pe=st(wf):x=3*nx [2925]
-2:y=3*ny+7:GOSUB 3000
2330 f(nx,ny)=ABS(wf):en=en-1:GOSUB 3800:I [3095]
F en=0 THEN 2900
2340 GOTO 2120                              [307]
2350 REM================================    [2077]
2351 REM===  Zu kleiner Markenwert  ===    [2080]
2352 REM================================    [2077]
2360 k(m)=1:x=3*sx-2:y=3*sy+7:GOSUB 3300    [1633]
2370 f(sx,sy)=30:en=en-5:GOSUB 3800         [2546]
2380 x=mx:GOSUB 3650                        [1713]
2390 IF en<1 THEN 2900 ELSE IF m=6 THEN 27  [2628]
50 ELSE GOSUB 4300:GOTO 2050
2400 REM================================    [2077]
2401 REM===       Auf Totenkopf       ===    [1835]
2402 REM================================    [2077]
2410 k(m)=1:x=3*sx-2:y=3*sy+7:GOSUB 3600:G [2664]
OSUB 3300
2420 f(sx,sy)=30:en=en-7:GOSUB 3800         [2917]
2430 x=mx:GOSUB 3650                        [1713]
2440 IF en<1 THEN 2900 ELSE IF m=6 THEN 27  [2628]
50 ELSE GOSUB 4300:GOTO 2050
2450 REM================================    [2077]
2451 REM===       Auf Leerfeld        ===    [1610]
2452 REM================================    [2077]
2460 k(m)=1:x=3*sx-2:y=3*sy+7:GOSUB 3300    [1633]
2470 f(sx,sy)=30:en=en-6:GOSUB 3800         [2797]
2480 x=mx:GOSUB 3650                        [1713]
2490 IF en<1 THEN 2900 ELSE IF m=6 THEN 27  [2628]
50 ELSE GOSUB 4300:GOTO 2050
2500 REM================================    [2077]
2501 REM===  Ausreichend frankiert  ===    [1611]
2502 REM================================    [2077]
2510 sx=nx:sy=ny:x=3*sx-2:y=3*sy+7:GOSUB 3  [2546]
600
2520 IF f(sx,sy)<0 THEN 2300                [2405]
2530 IF f(sx,sy)=0 THEN 2350                [2153]
2540 IF f(sx,sy)>25 THEN 2400               [2210]
2550 IF f(sx,sy)=25 THEN mun=3:GOSUB 4400:  [2815]
```

```
GOTO 2700
2560 IF w(f(sx,sy))<mwert THEN 2450          [1488]
2570 IF w(f(sx,sy))=mwert THEN mun=2:GOSUB [3553]
 4400:GOTO 2600
2580 IF w(f(sx,sy))>mwert THEN mun=1:GOSUB [1908]
 4400:GOTO 2660
2590 GOTO 2520                               [355]
2600 REM==============================      [2077]
2601 REM===    Passend / Zu viel     ===     [1191]
2602 REM==============================      [2077]
2610 wf=f(sx,sy):wert=w(wf):pp=pa(wf):pe=s [3755]
t(wf):x=mx:y=4:GOSUB 3000:GOSUB 3400
2620 k(m)=2:f(sx,sy)=25:x=3*sx-2:y=3*sy+7: [3640]
GOSUB 3200
2630 grosch=grosch+mwert:en=en+1:IF en>42  [3426]
THEN en=42
2640 IF grosch>90 THEN dmark=dmark+1:grosc [2593]
h=grosch-100:GOTO 2680
2650 GOSUB 3700:IF m=6 THEN 2750 ELSE GOSU [3625]
B 4300:GOTO 2050
2660 wf=f(sx,sy):wert=w(wf):pp=pa(wf):pe=s [3755]
t(wf):x=mx:y=4:GOSUB 3000:GOSUB 3400
2670 k(m)=2:f(sx,sy)=0:grosch=grosch+mwert [4364]
2680 IF grosch>90 THEN dmark=dmark+1:grosc [2593]
h=grosch-100:GOTO 2680
2690 GOSUB 3700:IF m=6 THEN 2750 ELSE GOSU [3625]
B 4300:GOTO 2050
2700 REM==============================      [2077]
2701 REM===         Auf Taler         ===     [1146]
2702 REM==============================      [2077]
2710 x=mx:y=5:GOSUB 3200                     [2581]
2720 k(m)=2:f(sx,sy)=0:dmark=dmark+10        [2018]
2730 GOSUB 3700                              [1013]
2740 IF m=6 THEN 2750 ELSE GOSUB 4300:GOTO [3206]
 2050
2750 REM==============================      [2077]
2751 REM===     Ende der Runde       ===     [1071]
2752 REM==============================      [2077]
2760 h=0:FOR q=1 TO 6:h=h+k(q):NEXT q:h=IN [4021]
T((h-6)/2)
2770 en=en+h:IF en>42 THEN en=42             [1719]
2780 IF h=3 THEN dmark=dmark+6:GOSUB 3700: [4725]
mun=4:GOSUB 4400 ELSE GOSUB 3700
2790 SOUND 1,0,40,4,0,0,6:FOR q=260 TO 346 [5602]
 STEP 2:MOVE 4,q:DRAW 584,q,11:NEXT q
2800 REM==============================      [2077]
2801 REM=== Zerfall des Totenkopfs  ===     [1887]
2802 REM==============================      [2077]
2810 FOR hy=1 TO 5:FOR hx=1 TO 6:x=3*hx-2: [2569]
y=3*hy+7
2820 IF f(hx,hy)=26 THEN f(hx,hy)=0:GOSUB  [2712]
3600
2830 IF f(hx,hy)>26 THEN f(hx,hy)=f(hx,hy) [3655]
-1
2840 NEXT hx,hy                              [592]
2850 REM==============================      [2077]
2851 REM===    Neue Briefmarken      ===     [1722]
2852 REM==============================      [2077]
2860 FOR hy=1 TO 5:FOR hx=1 TO 6             [1292]
2870 IF f(hx,hy)=0 THEN f(hx,hy)=INT(RND(T [3287]
IME)*24)+1 ELSE 2890
2880 wf=f(hx,hy):wert=w(wf):pp=pa(wf):pe=s [6526]
t(wf):x=3*hx-2:y=3*hy+7:GOSUB 3000
2890 NEXT hx,hy:m=0:GOTO 2050                [1609]
2900 REM==============================      [2077]
2901 REM===         GAME OVER         ===     [961]
2902 REM==============================      [2077]
2910 FOR h=1 TO 6:FOR q=1 TO 15              [1530]
2920 SOUND 1,0,1,0:SOUND 1,0,h,7,0,0,q:NEX [3264]
T q,h
2930 FOR q=0 TO 640 STEP 8:MOVE q,164:DRAW [3452]
R 0,60,1
2940 MOVE 644-q,164:DRAWR 0,60,1:NEXT q      [1742]
2950 LOCATE 2,13:PAPER 1                     [1189]
2960 PEN 9:dat$="G A M E ":GOSUB 4100        [1837]
2970 PEN 5:dat$="  O V E R":GOSUB 4100       [1621]
2980 h=JOY(0):FOR q=1 TO 5:h=h+1+INKEY(q): [4153]
NEXT q:IF h=0 THEN 2980
2990 BORDER 0:PAPER 0:PEN 1:CLS:GOTO 4600    [1639]
3000 REM==============================      [2077]
3001 REM===       Briefmarke 1       ===     [840]
3002 REM==============================      [2077]
3010 LOCATE x,y+1:PEN pp:PRINT an$;CHR$(25 [2428]
4);
3020 LOCATE x,y+2:PEN pp:PRINT an$;CHR$(25 [2978]
4);
3030 PRINT aus$;:PAPER pp:PEN pe             [2623]
3040 IF wert<100 THEN c=32:h=0 ELSE c=INT( [3427]
wert/100)+210:h=10
3050 LOCATE x+1,y+1:PRINT CHR$(c);           [1776]
3060 LOCATE x+1,y+2:PRINT CHR$(c+h);         [1531]
3070 c=(wert-100*INT(wert/100))/10+230:h=1 [2666]
0
3080 LOCATE x+2,y+1:PRINT CHR$(c);           [1097]
3090 LOCATE x+2,y+2:PRINT CHR$(c+h);         [2314]
3100 REM==============================      [2077]
3101 REM===       Briefmarke 2       ===     [832]
3102 REM==============================      [2077]
3110 PAPER 0:PEN 1                           [965]
3120 LOCATE x,y:PRINT an$;CHR$(170)CHR$(17 [2349]
1)CHR$(171)CHR$(172);
3130 LOCATE x,y+1:PRINT CHR$(173);           [1153]
3140 LOCATE x,y+2:PRINT CHR$(173);           [1838]
3150 LOCATE x+3,y+1:PRINT CHR$(174);         [2118]
3160 LOCATE x+3,y+2:PRINT CHR$(174);         [2456]
3170 LOCATE x,y+3:PRINT CHR$(175)CHR$(176) [2899]
CHR$(176)CHR$(177);
3180 PRINT aus$;                             [322]
3190 SOUND 1,0,5,5,0,0,1:RETURN             [2160]
3200 REM==============================      [2077]
3201 REM===          Taler          ===     [514]
3202 REM==============================      [2077]
3210 LOCATE x+1,y+1:PEN 7:PRINT an$;CHR$(1 [2322]
38)CHR$(133);
3220 LOCATE x+1,y+2:PEN 7:PRINT an$;CHR$(1 [3433]
38)CHR$(133);
3230 PEN 2                                   [548]
3240 LOCATE x+1,y:PRINT CHR$(186)CHR$(187) [2443]
;
3250 LOCATE x+1,y+1:PRINT CHR$(188)CHR$(18 [2726]
9);
3260 LOCATE x+1,y+2:PRINT CHR$(190)CHR$(19 [1384]
1);
3270 LOCATE x+1,y+3:PRINT CHR$(192)CHR$(19 [1497]
3);
3280 PRINT aus$;                             [322]
3290 FOR q=0 TO 2:SOUND 1,30,1,7:SOUND 2,7 [5954]
0,1,7:NEXT q:RETURN
3300 REM==============================      [2077]
3301 REM===        Totenkopf        ===     [1791]
3302 REM==============================      [2077]
3310 LOCATE x+1,y+1:PEN 5:PRINT an$;CHR$(1 [2629]
38)CHR$(133);
3320 LOCATE x+1,y+2:PEN 5:PRINT an$;CHR$(2 [3364]
54)CHR$(255);
3330 PEN 1                                   [549]
3340 LOCATE x,y:PRINT CHR$(194)CHR$(195)CH [3620]
R$(196)CHR$(197);
3350 LOCATE x,y+1:PRINT CHR$(198)CHR$(199) [2375]
CHR$(200)CHR$(201);
3360 LOCATE x,y+2:PRINT CHR$(202)CHR$(203) [3423]
CHR$(204)CHR$(205);
3370 LOCATE x,y+3:PRINT CHR$(206)CHR$(207) [3558]
CHR$(208)CHR$(209);
3380 PRINT aus$;                             [322]
3390 FOR q=300 TO 1800 STEP 100:SOUND 1,q, [3436]
3,7:NEXT q:RETURN
3400 REM==============================      [2077]
3401 REM===         Stempel          ===     [664]
3402 REM==============================      [2077]
3410 PEN 0                                   [550]
3420 PRINT an$;                              [725]
3430 LOCATE x+1,y+1:PRINT CHR$(178)CHR$(17 [2614]
9);
3440 LOCATE x+1,y+2:PRINT CHR$(180)CHR$(18 [1680]
1);
3450 LOCATE x+1,y+3:PRINT CHR$(182)CHR$(18 [1395]
3);
3460 LOCATE x+1,y+4:PRINT CHR$(184)CHR$(18 [1633]
5);
3470 PRINT aus$;                             [322]
3480 SOUND 1,0,7,7,0,0,15                    [1619]
3490 RETURN                                  [555]
3500 REM==============================      [2077]
3501 REM===       CRAZY STAMP        ===     [2487]
3502 REM==============================      [2077]
3510 PAPER 0:PEN 1                           [965]
3520 PRINT an$;                              [725]
3530 LOCATE x,y+1:PRINT CHR$(254);           [1225]
3540 LOCATE x,y+2:PRINT CHR$(254);           [1961]
3550 PRINT aus$;                             [322]
3560 PAPER 1:PEN 0                           [871]
3570 LOCATE x+1,y+1:PRINT CHR$(250)CHR$(25 [3116]
2);
3580 LOCATE x+1,y+2:PRINT CHR$(251)CHR$(25 [1298]
3);
3590 FOR q=50 TO 600 STEP 50:SOUND 1,q,3,7 [3882]
:NEXT q:RETURN
```

```
3600 REM================================    [2077]
3601 REM===         Leerfeld           ===    [612]
3602 REM================================    [2077]
3610 gx=32*x-12:gy=404-16*y                  [1622]
3620 FOR q=0 TO 21                           [1028]
3630 PLOT gx,gy-2*q,0:DRAWR 80,0:NEXT q      [1722]
3640 RETURN                                  [555]
3650 REM================================    [2077]
3651 REM===       Rot Ankreuzen        ===    [1209]
3652 REM================================    [2077]
3660 PAPER 0:PEN 3                           [963]
3670 LOCATE x+1,5:PRINT an$;CHR$(92)CHR$(4 [1808]
7);aus$;
3680 LOCATE x+1,6:PRINT an$;CHR$(47)CHR$(9 [2643]
2);aus$;
3690 RETURN                                  [555]
3700 REM================================    [2077]
3701 REM===      Anzeige * Geld        ===    [1441]
3702 REM================================    [2077]
3710 h$=STR$(dmark)                          [788]
3720 h$=STRING$(6-LEN(h$),"0")+MID$(h$,2,L [3049]
EN(h$)-1)
3730 FOR x=8 TO 12                           [924]
3740 c=152+ASC(MID$(h$,x-7,1))               [1889]
3750 FOR y=1 TO 2                            [1188]
3760 LOCATE x,y:PAPER 12:PEN 2:PRINT CHR$( [4411]
c+10*y);
3770 NEXT y,x                                [397]
3780 c=grosch/10+220                         [1235]
3790 FOR q=1 TO 2:LOCATE 13,q:PRINT CHR$(c [2788]
+10*q);:NEXT q
3800 REM================================    [2077]
3801 REM===     Anzeige * Energie    ===    [1829]
3802 REM================================    [2077]
3810 LOCATE 1,1:PRINT an$;                   [744]
3820 FOR y=1 TO 42:h=25-INT(y/2)             [1726]
3830 IF y>8 THEN pe=2 ELSE pe=3              [1774]
3840 IF y>en THEN pe=0                       [1206]
3850 IF INT(y/2)=y/2 THEN c=132 ELSE c=129 [1696]
3860 LOCATE 19,h:PEN pe:PRINT" ";CHR$(c);    [2564]
3870 NEXT y                                  [359]
3880 PRINT aus$;                             [322]
3890 RETURN                                  [555]
3900 REM================================    [2077]
3901 REM===         Sucher an          ===    [1426]
3902 REM================================    [2077]
3910 gx=32*x-20:gy=408-16*y:FOR q=0 TO 1:F [3395]
OR h=0 TO 1
3920 PLOT gx,gy-2*q-48*h,3+(q=h):DRAWR 96, [4172]
0:NEXT h,q:FOR q=0 TO 1:FOR h=0 TO 1
3930 PLOT gx+4*q+92*h,gy-2,3+(q=h):DRAWR 0 [3630]
,-46:NEXT h,q
3940 SOUND 1,70,7,7:RETURN                   [1553]
3950 REM================================    [2077]
3951 REM===         Sucher aus         ===    [1599]
3952 REM================================    [2077]
3960 gx=32*x-20:gy=408-16*y:FOR q=0 TO 1:F [3395]
OR h=0 TO 1
3970 PLOT gx,gy-2*q-48*h,0:DRAWR 96,0        [2276]
3980 PLOT gx+4*q+92*h,gy-2,0:DRAWR 0,-46:N [3235]
EXT h,q
3990 RETURN                                  [555]
4000 REM================================    [2077]
4001 REM===      Melodie spielen      ===    [1444]
4002 REM================================    [2077]
4010 FOR q=1 TO u                            [918]
4020 IF INKEY(47)=0 OR INKEY$<>""THEN q=78 [1457]
4030 IF h(q)=0 THEN SOUND 1,0,d(q)*8,7 ELS [4644]
E SOUND 1,t(h(q)),d(q)*8,7
4040 IF h(q)=0 THEN SOUND 2,0,d(q)*7,6 ELS [5104]
E SOUND 2,t(h(q)+12),d(q)*7,6
4050 IF h(q)=0 THEN SOUND 4,0,d(q)*6,5 ELS [3210]
E SOUND 4,t(h(q)+24),d(q)*6,5
4060 SOUND 7,0,2                             [951]
4070 IF INKEY(47)=0 OR INKEY$<>""THEN q=78 [1457]
4080 NEXT q                                  [367]
4090 RETURN                                  [555]
4100 REM================================    [2077]
4101 REM===       Riesenschrift        ===    [672]
4102 REM================================    [2077]
4110 FOR q=1 TO LEN(dat$)                    [1477]
4120 FOR h=0 TO 7                            [638]
4130 s(h)=PEEK(ad!+(ASC(MID$(dat$,q,1))-32 [1532]
)*8+h)
4140 NEXT h                                  [372]
4150 SYMBOL 167,s(0),s(0),s(1),s(1),s(2),s [2140]
(2),s(3),s(3)
4160 SYMBOL 168,s(4),s(4),s(5),s(5),s(6),s [2708]
(6),s(7),s(7)
4170 PRINT CHR$(167)CHR$(10)CHR$(8)CHR$(16 [2707]
8)CHR$(11);
4180 NEXT q                                  [367]
4190 RETURN                                  [555]
4200 REM================================    [2077]
4201 REM=== Wertvorgabe ausdrucken  ===    [1434]
4202 REM================================    [2077]
4210 PEN 15:PRINT an$;                       [1477]
4220 IF mwert<100 THEN c=32:h=0 ELSE c=INT [3443]
(mwert/100)+210:h=10
4230 LOCATE mx+1,5:PRINT CHR$(c);            [2683]
4240 LOCATE mx+1,6:PRINT CHR$(c+h);          [1721]
4250 c=(mwert-100*INT(mwert/100))/10+230:h [2805]
=10
4260 LOCATE mx+2,5:PRINT CHR$(c);            [1848]
4270 LOCATE mx+2,6:PRINT CHR$(c+h);          [2946]
4280 PRINT aus$;                             [322]
4290 RETURN                                  [555]
4300 REM================================    [2077]
4301 REM===     STAMP wird CRAZY     ===    [1163]
4302 REM================================    [2077]
4310 FOR h=1 TO 8-INT(en/6)                  [1127]
4320 hx=INT(RND(TIME)*6)+1                   [1740]
4340 hy=INT(RND(TIME)*5)+1                   [1775]
4350 IF hx=sx AND hy=sy THEN 4320            [1846]
4360 IF f(hx,hy)<1 OR f(hx,hy)>24 THEN 438 [2247]
0
4370 f(hx,hy)=-f(hx,hy):x=3*hx-2:y=3*hy+7: [2293]
GOSUB 3500
4380 NEXT h                                  [372]
4390 RETURN                                  [555]
4400 REM================================    [2077]
4401 REM===      4 Soundeffekte      ===    [1550]
4402 REM================================    [2077]
4410 ENV 1,1,15,1,8,0,1,15,-1,1              [1388]
4420 ON mun GOTO 4430,4450,4470,4490         [1006]
4430 FOR q=12 TO 24 STEP 2:SOUND 1,t(q),3, [3792]
0,1:SOUND 1,t(q+4),3,0,1:NEXT q
4440 SOUND 1,t(24),20,0,1:RETURN             [2109]
4450 FOR q=24 TO 36:SOUND 1,t(q),3,0,1:SOU [3289]
ND 1,t(q+2),3,0,1:NEXT q
4460 SOUND 1,t(36),20,0,1:RETURN             [1839]
4470 FOR q=24 TO 36:SOUND 1,t(q),2,7:SOUND [3270]
 1,t(q+8),2,7:NEXT q
4480 SOUND 1,t(36),20,0,1:RETURN             [1839]
4490 FOR q=1 TO 20:SOUND 1,70,2,7:SOUND 1, [4231]
70,1,0:NEXT q:RETURN
4500 REM================================    [2077]
4501 REM===       Namenseingabe        ===    [1883]
4502 REM================================    [2077]
4510 MODE 1:LOCATE 10,10:PAPER 0:PEN 2:PRI [5270]
NT"Bitte Namen eintippen:":name$=""
4520 SOUND 1,70,2,7:LOCATE 10,15:PEN 1:PRI [5660]
NT name$;:PEN 3:PRINT"_ ";
4530 w$=INKEY$                               [833]
4540 IF w$=CHR$(13)THEN 4580                 [1146]
4550 IF w$=""OR(w$=CHR$(127)AND LEN(name$) [2804]
=0)THEN 4530
4560 IF w$=CHR$(127)THEN name$=LEFT$(name$ [4302]
,LEN(name$)-1):GOTO 4520
4570 IF LEN(name$)<20 THEN name$=name$+w$: [2967]
GOTO 4520 ELSE 4530
4580 IF name$=""THEN name$="Mister SLUGGAR [2410]
D"
4590 LOCATE 10,15:PEN 1:PRINT name$;" ";:F [2389]
OR q=0 TO 1000:NEXT q:RETURN
4600 REM================================    [2077]
4601 REM=== Neue Tabelle erstellen  ===    [2480]
4602 REM================================    [2077]
4610 h=0                                     [416]
4620 FOR q=10 TO 1 STEP-1                    [1845]
4630 IF dm(q)<dmark OR(dm(q)=dmark AND gro [3327]
sch>gr(q))THEN h=q
4640 NEXT q                                  [367]
4650 IF h=0 THEN 4700                        [993]
4660 FOR q=10 TO h+1 STEP-1                  [1257]
4670 dm(q)=dm(q-1):gr(q)=gr(q-1):na$(q)=na [1941]
$(q-1)
4680 NEXT q                                  [367]
4690 dm(h)=dmark:gr(h)=grosch:na$(h)=name$ [1292]
4700 REM================================    [2077]
4701 REM===         Tabelle 1          ===    [602]
4702 REM================================    [2077]
4710 MODE 1:PAPER 1:CLS:BORDER 26:INK 2,2  [3307]
4720 PEN 3:LOCATE 1,1:dat$="* TOP TEN *  C [4647]
RAZY  STAMPS  * TOP TEN *":GOSUB 4100
4730 PEN 0:LOCATE 1,3:PRINT STRING$(39,154 [2405]
)
4740 h$=STR$(dmark):PEN 0:LOCATE 9,25        [1678]
4750 PRINT STRING$(6-LEN(h$),"0")+MID$(h$, [3119]
2,LEN(h$)-1);:PEN 0:PRINT"."
```

```
4760 h$=STR$(grosch+100):PEN 0:LOCATE 15,2 [2501]
5
4770 PRINT RIGHT$(h$,2)                    [888]
4780 PEN 0:LOCATE 19,25                    [1327]
4790 PRINT name$                           [588]
4800 REM=============================     [2077]
4801 REM===         Tabelle 2         ===  [552]
4802 REM=============================     [2077]
4810 FOR q=1 TO 10                         [837]
4820 LOCATE 4,2*q+3:PEN 0:PRINT USING"##"; [3811]
q;:PRINT"."
4830 h$=STR$(dm(q)):PEN 3:LOCATE 9,2*q+3   [1607]
4840 PRINT STRING$(6-LEN(h$),"0")+MID$(h$, [3119]
2,LEN(h$)-1);:PEN 0:PRINT"."
4850 h$=STR$(gr(q)+100):PEN 3:LOCATE 15,2* [1861]
q+3
4860 PRINT RIGHT$(h$,2)                    [888]
4870 PEN 2:LOCATE 19,2*q+3:PRINT na$(q)    [1999]
4880 NEXT q                                [367]
4890 WHILE INKEY$<>"":WEND:WHILE INKEY$="" [4349]
:WEND:GOTO 1300
4900 REM=============================     [2077]
4901 REM===    Tastaturbelegung     ===    [1366]
4902 REM=============================     [2077]
4910 MODE 1:LOCATE 10,2:PAPER 0:PEN 1:PRIN [5925]
T"TASTATURBELEGUNG:":FOR q=1 TO 5
4920 IF q=1 THEN LOCATE 10,4*q+2:PEN 2:PRI [3299]
NT"HOCH   : ";
4930 IF q=2 THEN LOCATE 10,4*q+2:PEN 2:PRI [3501]
NT"RECHTS : ";
4940 IF q=3 THEN LOCATE 10,4*q+2:PEN 2:PRI [3229]
NT"RUNTER : ";
4950 IF q=4 THEN LOCATE 10,4*q+2:PEN 2:PRI [2912]
NT"LINKS  : ";
4960 IF q=5 THEN LOCATE 10,4*q+2:PEN 2:PRI [2455]
NT"FEUER  : ";
4970 h=(h+1)MOD 80:IF INKEY(h)=-1 THEN 497 [4251]
0 ELSE SOUND 1,70,7,7
4980 r(q)=h:PEN 3:PRINT RIGHT$(STR$(r(q)+1 [2696]
00),2)
4990 NEXT q:FOR q=0 TO 3000:NEXT q:GOTO 13 [1399]
00
5000 REM=============================     [2077]
5001 REM===        Anleitung         ===   [788]
5002 REM=============================     [2077]
5010 MODE 1:BORDER 0:INK 0,0:INK 1,20:INK  [2413]
2,21:INK 3,15
5020 LOCATE 9,1:PEN 2:PRINT CHR$(164);" 19 [3312]
86 by Dirk Müller":LOCATE 1,3:PEN 3
5030 dat$="* Anleitung * CRAZY STAMPS * An [4102]
leitung *":GOSUB 4100
5040 FOR q=6 TO 24 STEP 18:LOCATE 1,q:PEN  [4403]
2:PRINT STRING$(40,154):NEXT q
5050 LOCATE 9,25:PEN 3:PRINT"Bitte Leertas [4735]
te drücken"
5060 FOR q=1 TO 4:LOCATE 1,8:PEN 1         [1830]
5070 ON q GOTO 5200,5300,5400,5500         [816]
5080 IF INKEY(47)<>0 THEN 5080             [690]
5090 SOUND 1,70,7,7                        [1167]
5100 REM=============================     [2077]
5101 REM===         Wischen           ===  [772]
5102 REM=============================     [2077]
5110 FOR h=30 TO 300 STEP 4:SOUND 1,0,2,4, [1693]
0,0,4
5120 MOVE 0,h:DRAW 640,h,1                 [987]
5130 MOVE 0,334-h:DRAW 640,334-h,1         [1827]
5140 NEXT h                                [372]
5150 FOR h=30 TO 300 STEP 4:SOUND 1,0,2,4, [1693]
0,0,4
5160 MOVE 0,h:DRAW 640,h,0                 [984]
5170 MOVE 0,334-h:DRAW 640,334-h,0         [1824]
5180 NEXT h                                [372]
5190 NEXT q:GOTO 1300                      [678]
5200 REM=============================     [2077]
5201 REM===         Text 1            ===  [1051]
5202 REM=============================     [2077]
5210 PRINT"CRAZY STAMPS - Ein total verrück [2874]
kter Job!"
5220 PRINT"Die Aufgabe besteht darin, Post [3429]
sendungen"
5230 PRINT"AUSREICHEND zu frankieren. In d [4338]
em grünen"
5240 PRINT"Fenster erscheint der Wert,  we [4919]
lchen die"
5250 PRINT"Briefmarke  mindestens  haben m [5393]
u#.  Wird"
5260 PRINT"ausreichend frankiert, stempelt [6674]
 die Post"
5270 PRINT"ab, und der Lohn steigt!  Nach  [2566]
6 richtig"
5280 PRINT"geklebten Marken erh@lt man ein [3515]
en Bonus."
5290 GOTO 5080                             [355]
5300 REM=============================     [2077]
5301 REM===         Text 2            ===  [863]
5302 REM=============================     [2077]
5310 PRINT"Mit dem Sucher  kann man  belie [3942]
big viele"
5320 PRINT"Felder in EINE Richtung ziehen  [4940]
 und eine"
5330 PRINT"durch FEUER ausgew@hlte Marke a [3902]
ufkleben."
5340 PRINT"Wird eine genau passende Marke  [4873]
gefunden,"
5350 PRINT"so erscheint als Belohnung an d [3635]
er Stelle"
5360 PRINT"ein Goldstück. Mit diesem Taler [4374]
 kann man"
5370 PRINT"jeweils einmal die Portokosten  [4689]
bezahlen."
5380 PRINT"Dabei  gibt es  einen sehr hohe [3941]
n Bonus !"
5390 GOTO 5080                             [355]
5400 REM=============================     [2077]
5401 REM===         Text 3            ===  [855]
5402 REM=============================     [2077]
5410 PRINT"Gelangt man auf ein Leerfeld  o [3202]
der sucht"
5420 PRINT"man eine Marke mit zu geringem  [5762]
Wert aus,"
5430 PRINT"erscheint ein Totenkopf, und di [4774]
e Energie"
5440 PRINT"sinkt deutlich,  wie auch,  fal [4727]
ls man zu"
5450 PRINT"einem Totenkopf steuert.  Um    [5306]
möglichst"
5460 PRINT"lange jobben zu können, darf di [5690]
e Energie"
5470 PRINT"nicht ganz abfallen.  Bei  jed  [6140]
em  Bonus"
5480 PRINT"wird  die  Energie  ein  wenig  [4988]
 erhöht."
5490 GOTO 5080                             [355]
5500 REM=============================     [2077]
5501 REM===         Text 4            ===  [847]
5502 REM=============================     [2077]
5510 PRINT"Seltsamerweise  spielen  die Br [4529]
iefmarken"
5520 PRINT"manchmal verrückt und verändern [4298]
 ihr Aus-"
5530 PRINT"sehen,  so da# man ihren Wert n [4627]
ur wieder"
5540 PRINT"erfährt,  wenn  man  auf die be [5686]
troffenen"
5550 PRINT"Marken gelangt.  Aber auch  die [5010]
s  kostet"
5560 PRINT"Energie, mit der man sparsam se [4877]
in soll !"
5570 PRINT"**** Den Briefmarken-Freaks wün [3805]
scht ****"
5580 PRINT"****    Dirk Müller alles Gute  [2809]
!    ****"
5590 GOTO 5080                             [355]
6000 REM"*****  Utility  No. 1   ***** *** [4291]
********** $ by Dirk Mueller *****
6010 SYMBOL AFTER 32                       [1296]
6020 SYMBOL  64,108,0,120,12,124,204,118   [2059]
6030 SYMBOL 124,102,24,60,102,102,126,102  [1662]
6040 SYMBOL  91,102,0,60,102,102,102,60    [2192]
6050 SYMBOL 123,198,0,124,198,198,198,124  [2394]
6060 SYMBOL  93,102,0,102,102,102,102,62   [2151]
6070 SYMBOL 125,198,0,198,198,198,198,124  [2573]
6080 SYMBOL 163,60,70,198,220,198,198,220, [2335]
192
6090 RETURN                                [555]
```



# Meine Meinung zu diesem Sonderheft:

Hier haben Sie Gelegenheit, uns Ihre Meinung zu diesem Sonderheft mitzuteilen. Wir hoffen auf eine rege Teilnahme, nur so können wir unsere Fehler erkennen, Ihre Anregungen verarbeiten und es das nächste Mal besser machen, denn – Nobody's perfect !

Füllen Sie die freigelassenen Felder für Ihre Bemerkungen aus und senden Sie das Blatt (oder eine Fotokopie, falls Sie Ihr Sonderheft nicht zerpflücken möchten) an o.g. Anschrift. Wir danken schon jetzt für Ihre aktive Unterstützung!

An den
DMV Verlag
Stichwort: Sonderheft Nr. 5
Postfach 250
3440 Eschwege

Das finde ich gut: _____

_____

_____

Das finde ich nicht gut: _____

_____

_____

Ich wünsche mir: _____

_____

_____

_____

Name und Anschrift: _____

_____

_____

Computerkonfiguration: _____

Ihre persönlichen Daten werden nur zu internen Verwaltungszwecken elektronisch gespeichert.

# STARTEST



für Schneider
CPC 464 · 664 · 6128

Der Zeitherr, ein Wesen aus den Tiefen des Weltraums, hat einen kosmischen Wettbewerb ausgeschrieben um etwas Zerstreuung in sein unendlich langes Leben zu bringen: Den STARTEST. Der Preis sind zusätzliche Lebensjahre. Jede Intelligenz, ob Spinnenwesen von Andalus, ob Menschen von der Erde oder Schlafende Steine von Murks können teilnehmen. Doch vor dem ersehnten Preis stehen 8 Prüfungen, die der Kandidat über sich ergehen lassen muß. 8 Prüfungen, die sowohl die Intelligenz als auch die Geschicklichkeit testen.

Nur wenige können Sie alle bestehen, aber wer es schafft, dem winkt eine Verlängerung seiner Lebensspanne.

**Kassettenversion 24,– DM***
**Diskette 3“ 29,– DM***

zu beziehen beim guten Fachhandel oder direkt vom:

* Unverbindliche Preisempfehlung

DMV-Verlag · Postfach 250 · 3440 Eschwege

# INFOCOM™

## Kommt!

DON'T PANIC!

Die Legenden – die Klassiker – die Neuheiten










INFOCOM c/o Activision Deutschland GmbH Postfach 760680 · 2000 Hamburg 76