14,- DM Ös 120,- sfr 14,-

CPC SONDERHEFT

INTERNATIONAL

Sonderheft

7-88/89

ca. 400 KB

ca. 400 KB

ca. 400 KB

ca. 400 KB

enthält ca. 400 KB CPC-Software

Grundlagen:

Alles über Floppy-Programmierung

Spiele:

3D-Tron

Auxilia - Im Reich der Magie

Anwendungen:

Musikprogramm de Luxe

Chemie – ganz einfach

CP/M:

Viele Tips und Ergänzungen

Tips & Tricks:

Stark: Basic-Programme verkürzen

Hardware:

Sprachsynthesizer de Luxe

Super: Viele Seiten Tips & Tricks

FAST BASIC COMPILER



Jetzt neu:
BASIC-Compiler für CPC 464/664/6128

Der Turbo-Antrieb für Ihre BASIC-Programme!

Haben auch Sie sich schon immer gewünscht, daß Ihre selbstgeschriebenen BASIC-Programme schneller laufen?
Mit dem BASIC-Compiler von DMV ist das nun kein Problem mehr, denn



- der Compiler hat den vollen Sprachumfang des BASIC 1.1 (CPC664/6128)
- das compilierte Programm ist auf jedem CPC lauffähig
- unterstützt Integer- und Fließkomma-Arithmetik
- kompatibel zu Vortex-Peripherie incl. Nutzen der RAM-Disk
- Programme, die spezielle BASIC 1.1.-Befehle beinhalten, sind auch auf dem CPC464 lauffähig (außer FILL und MASK)
- der Compiler arbeitet unter CP/M, das heißt, alle CP/M-Dienstprogramme können genutzt werden.
- bis 17 KB Quellcode können problemlos compiliert werden
- einzelne Programmteile können ebenfalls compiliert werden (z.B. wichtig bei Nachladeprogrammen)
- die ausführliche deutsche Bedienungsanleitung macht Sie auf einfache Weise mit dem Umgang des Compilers vertraut.
- viele Beispielprogramme veranschaulichen die Arbeitsweise des Compilers und zeigen die Geschwindigkeitsvorteile auf.
- das Programm ist in 100% Maschinencode geschrieben



Der BASIC-Compiler ist auf 3"-Diskette zum Preis von

Best.-Nr.: 209

Inland:		Ausland:	
Einzelpreis	69,– DM	Einzelpreis	69,– DM
zzgl. Versandkosten	3,– DM	zzgl. Versandkosten	5,– DM
Endpreis	72,– DM	Endpreis	74,– DM

Bitte benutzen Sie den Coupon auf Seite 129.

DMV Verlag · Postfach 250 · 3440 Eschwege

Impressum

Herausgeber
Christian Widuch

Chefredakteur
Stefan Ritter

Redaktion
Claus Daschner (cd), Jürgen Borngießer (jb)

Redaktions-Assistenz
Anke Kerstan (Ke)

Produktionsleitung
Gerd Köberich, Helmut Skoupy

Satz
Claudia Küllmer, Silvia Führer, Martina Siebert, Gabriela Joseph, Marcus Geppert

Gestaltung
Yvonne Hendricks, Mohamed Hawa, Petra Biehl

Reprografie
Margarete Schenk, Manuela Eska, Dieter Schnobl

Illustration
Heinrich Stiller

Fotografie
Christian Heckmann, Klaus Jatho

Lektorat
Susanne Mias, Christoph Dutt

Anzeigenleitung
Wolfgang Schnell
Es gilt die Anzeigenpreisliste Nr. 3 v. 1.12.'87

Anzeigenverkauf
Wolfgang Brill

Anschrift Verlag/Redaktion:
DMV-Daten & Medien Verlagsges. mbH
Postfach 250, Fuldaer Str. 6
3440 Eschwege
Telefon (0 56 51) 80 09–0 · Telex 993 210 dmv d
Telefax (0 56 51) 80 09–33

Vertrieb
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise:
CPC International Sonderheft
erscheint unregelmäßig

Einzelpreis DM 14,–/sfr. 14,–/OS 120,–

Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung.

Honorare nach Vereinbarung (die Zustimmung zum Abdruck wird vorausgesetzt).

Das Urheberrecht für veröffentlichte Manuskripte liegt ausschließlich beim Verlag. Nachdruck sowie Vervielfältigung oder sonstige Verwertung von Texten, nur mit schriftlicher Genehmigung des Verlages.

Namentlich gekennzeichnete Fremdbeiträge geben nicht in jedem Fall die Meinung der Redaktion wieder.



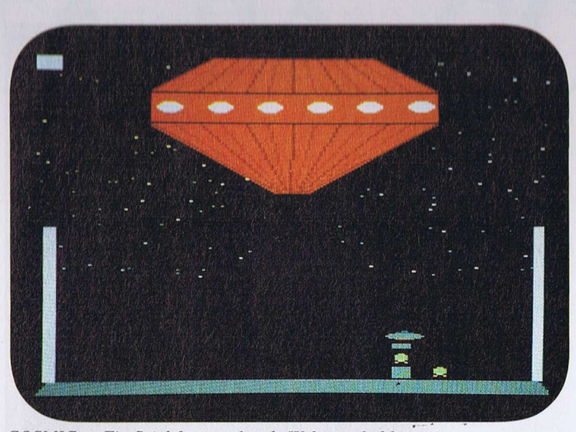
"Qualität als Maßstab"

Liebe Leser,
nach einer kurzen Pause liegt nun das insgesamt siebte CPC-Sonderheft vor Ihnen. Getreu dem Motto "was lange währt, wird endlich gut" haben wir uns diesmal bei der Vorbereitung dieser Sonderausgabe etwas Zeit gelassen. Dafür bieten wir Ihnen absloute Spitzenqualität in allen Anwendungsbereichen; die Software in dieser Ausgabe ist mit dem Heftpreis eigentlich nicht zu bezahlen. Daß dabei das eine oder andere Listing den normalen Umfang etwas überschreitet, kann diesmal wirklich als Qualitätsmerkmal gelten, das Abtippen lohnt sich in jedem Fall. Nach nunmehr fast vier Jahren CPC-Erfahrung müssen wir feststellen, daß die Leistungsmöglichkeiten dieser nach wie vor erfolgreichen Computer noch immer nicht ausgeschöpft sind. Täglich erreichen uns Zuschriften und Programme, die bis jetzt noch nie dagewesene Effekte und Geheimnisse der CPC-Programmierung offenbaren. Ein Ende ist bei weitem noch nicht abzusehen, so daß auch für die Zukunft eine weitere Qualitätssteigerung erwartet werden darf. Einen in der letzten Zeit immer häufiger auftretenden Wunsch unserer Leser können wir nun endlich erfüllen. Einige CPC-Besitzer nennen zusätzlich einen PC ihr Eigen und fragen nach der Möglichkeit, Daten und Programme vom CPC auf den PC zu konvertieren. Unser Beitrag "Von Rechner zu Rechner" macht Sie mit einer besonders gelungenen Lösung dieses Problems bekannt. Im Übrigen freuen wir uns immer wieder über Kritiken und Anregungen, die auch diesmal hoffentlich in großer Zahl an uns gerichtet werden.

Herzlichst Ihr

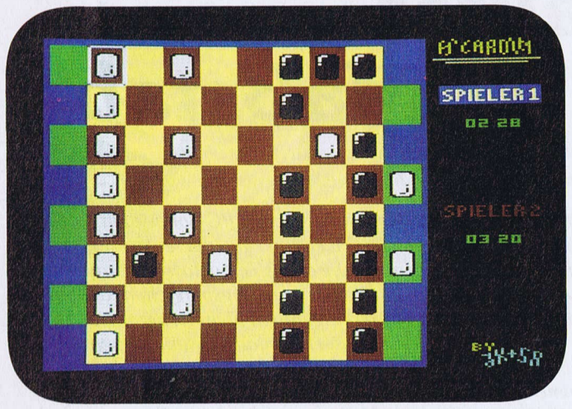
Stefan Ritter

Stefan Ritter
(Chefredakteur)



COSMIC – Ein Spiel für angehende Weltraumhelden.

S. 20



Ein Weltraum-Schach für zwei Spieler: H'CARDIN. Versuchen Sie, Ihren Gegner zu überlisten.

S. 24



Die Pyramide des Pharaos birgt viele Schätze. Suchen Sie sie im LABYRINTH

S. 47

Berichte:

Knack die Nuß 6
Für die Rätselfreunde unter Ihnen haben wir wieder knifflige Programmier- Probleme herausgesucht. Lerneffekt garantiert!!

Die Fließkomma-Arithmetik des CPC 8
Ob Spiele oder Anwendungen, ein Computer kommt nicht ohne Berechnungen aus. Wie Sie die Fließkomma-Arithmetik für sich selber nutzen können, zeigt Ihnen anschaulich unser Bericht.

Spiele:

Tron 3D 12
Ein altbekanntes Spiel in einer neuen Dimension. Versuchen Sie Ihrem Gegner den Weg abzuschneiden, aber passen Sie auf, daß dieser Ihnen nicht das Gleiche antut.

Müller 16
Kennen Sie 'Mühle'? Dann müssen Sie Müller kennen- und spielenlernen. Doch Vorsicht!! Ihr CPC als Spielgegner ist schlau!

Cosmic 20
– das Raumfahrt-Abenteuer. Steuern Sie Ihr Raumschiff durch ein Meteoritenfeld und landen Sie es auf dem verbotenen Planeten.

H'Cardin 24
Besetzen Sie zuerst die vier Hauptfelder Ihres Mitspielers, bevor er Ihre besetzt. Ein Spiel für Knobelfreunde.

Mission Wega 28
Eine Raumschiff zu starten, ist nicht schwer, aber die Landung… Versuchen Sie Ihr Glück und haben Sie Spaß dabei.

Acceleration 33
Erkunden Sie ein weiträumiges Höhlensystem – aber passen Sie auf, daß Sie nirgendwo anecken. Ein Actionspiel für starke Nerven.

Ein Stein mit Senf und Arni 36
Für alle, die Solitaire kennen, versuchen Sie doch mal diese Umsetzung. Sie werden Überraschungen erleben…

Auxilia 39
Besiegen Sie die gegnerische Armee der Unbesiegbaren. Aber Achtung! Nicht nur Zauberkräfte stellen sich Ihnen in den Weg…
Ein Strategiespiel mit viel Action.

Domingo 45
… schöner als Domino. Setzen Sie die farbigen Steine in die richtige Reihenfolge.

Wer sucht der findet! 47
Die Pyramide des Pharaos birgt viele Geheimnisse. Suchen und finden Sie die Schätze.

Anwendungen:

Manche mögen's digital! 54
Impuls – Ein Programm zur grafischen Darstellung digitaler Werte.

Die Welt der Elemente 58
Chemie ist ab sofort kein Thema mehr, wenn Sie unser Lehrprogramm 'Chemystical' abtippen. Die Welt der Elemente wird Ihnen anschaulich nähergebracht.

MUSICUS 62
Der CPC als Soundmachine – erstellen Sie sich Ihre eigenen Synthesizer- Klänge.

Taschenrechner im Großformat 70
Berechnungen im Großformat – Nutzen Sie Ihren Computer doch einmal als technisch-wissenschaftlichen Taschenrechner.

Mehr Farbe ins Bild 74
SCREEN PACK Plus – die optimale Ausnutzung des CPC im Grafikbereich.

Von Rechner zu Rechner 86
Datenübertragung vom PC zum CPC 6128 und zurück – mit diesem Programm kein Problem mehr.

Komplex 92
Die Problematik komplexer Zahlen auf den CPC's anschaulich gelöst.

Diskettenzauber 97
Sie möchten Ihre Disketten besser kennenlernen? Da haben wir zwei starke Programme für Sie: AMSMON+, der Diskmonitor, und BASIC765 – zum Programmieren Ihrer Floppy.

Control Program for Microcomputers – das unbekannte Wesen 110
Wichtige Informationen für diejenigen, die sich an die Programmierung von CP/M-Programmen heranwagen wollen.

Sportlich! Sportlich! – Vereinsverwaltung 115
Ob Sport,- Wander- oder Computerverein, mit unserem Programm bekommt Ihr Kassenwart Hilfe.

Tips und Tricks:

CPC-Listingsservice 114
Die Checksummer für Ihren CPC

Dreiecksverhältnisse 134
Das Pascalsche Dreieck – oder wie man mit wenig Aufwand fraktale Grafiken zaubern kann.

Rhythm and Soul 136
Schauen Sie sich doch mal Ihren Biorhytmus an, dank eingebauter Grafik- Routinen kein Problem.

Datensicherheit einmal anders 138
Erstellen Sie sich Backups von Ihren wichtigsten Disketten – auf Kassette.

Ein Tipp – ein Wort 141
Eine Funktionstastenbelegung mit Turbo Pascal-Schlüsselwörtern.

Je kürzer, desto besser 142
Der BASIC-Programmcode-Kompressor macht Ihre langen Listings bedeutend kürzer.

Mit Kompos KByte sparen! 145
Bildschirmgrafiken brauchen auf Diskette weniger Platz, wenn Sie Kompos benutzen.

Aus Eins mach Zwei 146
COPY-ALL-DISK – eine schnelle Kopierroutine für Diskettenbesitzer.

664/6128-Emulator 148
CPC 464-Besitzer können nun auch die meisten CPC 6128-Programme benutzen.

Page 150
CONTEXT-Texte im OCP-Art Studio einsetzen können Sie mit diesem Utility.

Programmschreiber 153
BASIC- Zeilen im Programm erzeugen – unser Utility zeigt Ihnen wie.

Hüllenlos 154
Erstellen Sie sich per Computer Ihre Kassettenhüllen – für Musik und Programme.

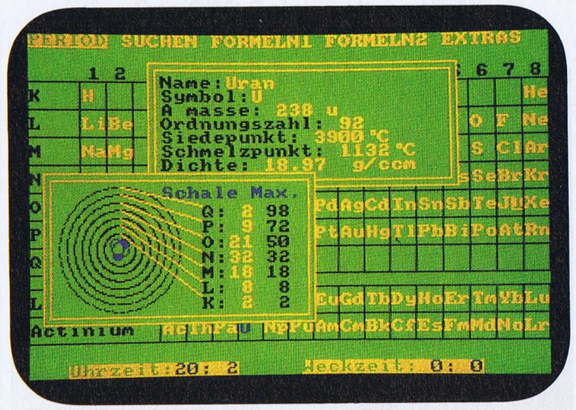
Auch Gutes kann noch besser werden! 158
Für alle, die gerne und oft unter CP/M arbeiten, ist dieses Utility gedacht. Lernen Sie unsere CP/M- Verbesserungen kennen…

Hardware:

Der Laberkasten 121
Lassen Sie Ihren CPC auch mal was sagen! Ein Sprachsynthesizer macht's möglich.

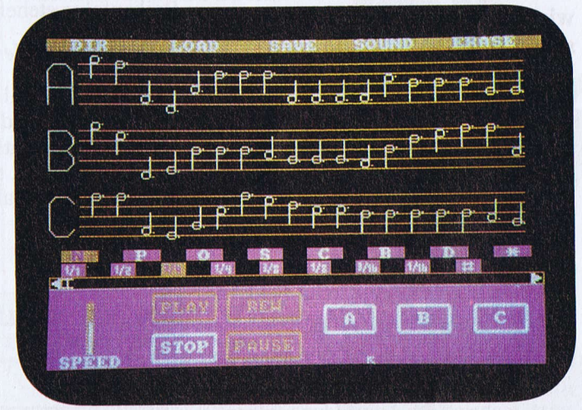
Rubriken:

Editorial 3
Impressum 3



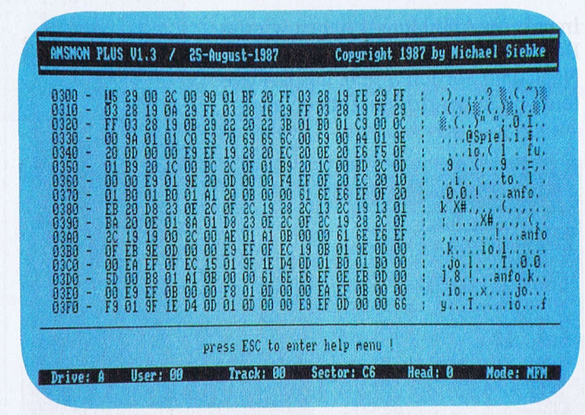
Chemie ist um uns herum. Lernen Sie die Elemente kennen mit CHEMYSTICAL.

S. 58



Komponieren Sie doch mal mit diesem Musik- Programm. MUSICUS macht aus Ihrem CPC einen Synthesizer.

S. 62



AMSMON+ (Bild) und BASIC765 – zwei Superprogramme für Ihre Disketten.

Knack die Nuß

COMPUTER chaos



Informatiklehrer haben sehr oft einen etwas bösartigen Zug und wollen hin und wieder Sachen wissen, bei denen man ganz schön ins Grübeln kommen kann. Ihre Neugierde kennt keinerlei Grenzen.

Neulich trat wieder einmal so ein Fall ein. Da wollte nämlich meiner folgendes wissen:

»Ein Zahlenfeld, genauer ein eindimensionales Array, mit zehn Feldern sei rein zufällig mit 10 Zahlen bestückt.

Ein Beispiel zeigt die erste Tabelle in der Abbildung 1. Die Zahlen sollen nur durch mehrmalige Anwendung folgender Operation sortiert werden:

Man wählt ein Feld aus. Die Zahlen links des ausgewählten Feldes bleiben unverändert, die restlichen Zahlen werden gespiegelt.

In unserer Abbildung 1 haben wir das Feld 3 ausgewählt (zweite Tabelle). Nach dem Sortieren sieht das Zahlenfeld wie in der dritten Tabelle aus.

Gesucht wird nun ein Lösungsweg, mit dem dann, bei beliebiger Belegung der Felder, die Zahlen in aufsteigender Reihenfolge stehen…«

Soweit das Problem.

Nun gut: Was soll man da machen? Problem lesen (und lesen und lesen…), »Was will er eigentlich?« fragen, Computer einschalten, probieren, Computer wieder ausschalten und ersteinmal überlegen:



Abb. 1

- **Schritt 1:**
  Untergrenze festlegen
- **Schritt 2:**
  Minimum festlegen
- **Schritt 3:**
  Ab Untergrenze kleinsten Wert in der Tabelle suchen, Minimum und Index merken
- **Schritt 4:**
  Ab dieser Stelle das gefundene Minimum nach oben spiegeln
- **Schritt 5:**
  Ab Untergrenze spiegeln, damit das oberste Element (aktuelles Minimum des Feldes) an den Rest angehängt wird
- **Schritt 6:**
  Untergrenze festlegen
- **Schritt 7:**
  Neues Minimum festlegen
- **Weiter mit Schritt 2**

Daß diese Methode funktioniert, zeigt unser kurzes Listing.

(Eckehart Röscheisen)

für 464-664-6128

```
1000 defint a-z
1010 for i=0 to 9:z[i]=rnd*9:print z[i];:next
1020 :
1030 m=0
1040 for u=0 to 8
1050   f=0
1060   for i=u to 9
1070     if z[i]<m then j=i:f=-1
1080   next
1090   if not f then m=m+1:goto 1060
1100   gosub 1160
1110   j=u:gosub 1160
1120 next
1130 print:for i=0 to 9:print z[i];:next
1140 end
1150 :
1160 o=9
1170 for i=j to (9+j)\2
1180   h=z[i]:z[i]=z[o]:z[o]=h:o=o-1
1190 next
1200 return
```


KNOW

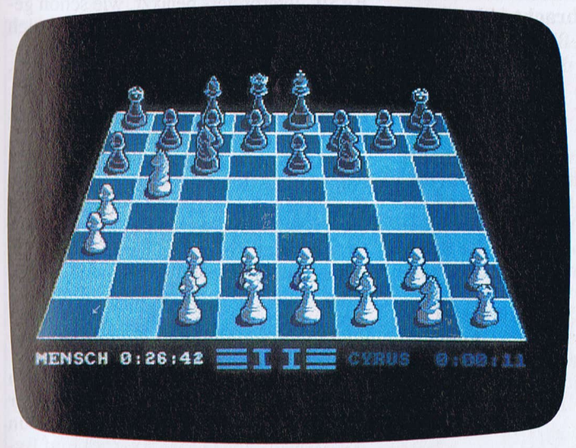


· kann eine **unbegrenzte** Anzahl von Fragen und Antworten verwalten · hat ständig ca. **400** Fragen **plus** Antworten im Speicher · bietet die Möglichkeit, **eigene** Fragen einzugeben, und zwar mit einem **komfortablen** Editor · besitzt eine **Supergrafik** mit Window-Technik · ist vollkommen **menuegesteuert** · 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 man 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!

für CPC 464 · 664 · 6128

Best.-Nr. 106 **3" Diskette** **29,– DM***

CYRUS II SCHACH



Das bewährte Programm für CPC's

Wahlweise 3D oder 2D Display, Einstellbare Spielstärke und viele weitere Extra-Features (mit deutscher Bedienungsanleitung).

zum Sonderpreis **Kassette** **12,95 DM***

Best.-Nr. 106

CPC Power-Spiele-Paket!

18 tolle Spiele für Ihren CPC 464, 664 oder 6128 auf 3"-Disketten zum Knüllerpreis von nur nur solange Vorrat reicht

Best.-Nr. 102 **4 Disketten** **70,– DM***

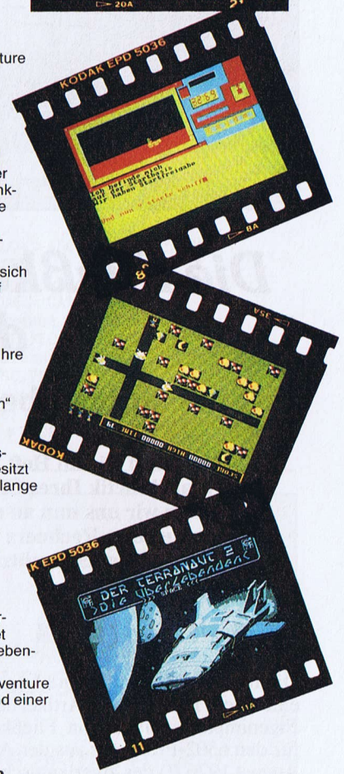
Fantastic Four

Das CPC Spielepaket enthält vier Super-Programme zum kleinen Preis

Die neue CPC-Spielebox enthält vier ausgesuchte TOP-Programme der Spitzenklasse. Da ist für jeden das richtige dabei!



COCKAINE
– ist ein echtes Weltraum-Actionspiel mit allem, was zu dieser Art von Software gehört. Die Erde kann das immer größer werdende Ozonloch nicht mehr verkraften, die Expedition COCKAIGNE soll in fernen Galaxien neue Lebensräume erkunden. Als Leiter dieser Expedition haben Sie alle Hände voll zu tun.....



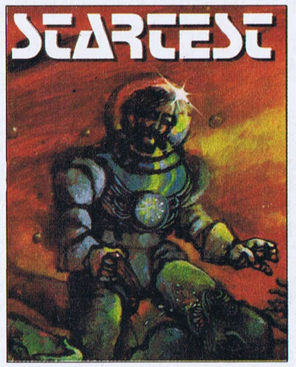
TERRANAUT I
– deutsches Science-Fiction-Adventure mit hervorragender Grafik. Vor 50 Jahren startete das letzte Generationen-Raumschiff TERRA von der Erde, um das Planetensystem Sirius zu kolonisieren. Durch eine kosmische Katastrophe verschwand die TERRA spurlos. Erst Jahre später fing eine Hyperfunkstation einen Funkspruch auf- die TERRA wurde in eine andere Galaxis katapultiert! Übernehmen Sie die Rolle des Commanders von Thaifun, dem ersten Galaxien-Kreuzer, und begeben Sie sich auf die Suche nach dem Raumschiff TERRA.

FRUITS
– Als Bauerssohn Frank jun. ist es Ihre Aufgabe, die Ernte der Felder einzufahren. Bei ihrem Vorhaben werden Sie allerdings von den "bösen Nasen" am Gelingen gehindert, zudem sind einige Feldfrüchte ziemlich ungesund. Fruits ist ein Geschicklichkeits- und Actionsspiel der Extraklasse, besitzt eine ausgefeilte Grafik und gewährt lange Spielfreude.

TERRANAUT II
– die Überlebenden. Nachdem Sie den ersten Teil gut überstanden haben, kommt sogleich die nächste Aufgabe auf Sie zu. Sie haben den Planeten gefunden, auf dem das vermißte Raumschiff TERRA gestrandet ist. Nun gilt es nach etwaigen Überlebenden der Katastrophe zu forschen. Terranaut II ist ein deutsches Textadventure mit vielen interessanten Features und einer spannenden Story.

Für alle CPCs nur als 3"Diskette

Best.-Nr. 1011 **49,– DM***



In den Tiefen der Galaxis wartet die letzte Herausforderung auf Sie

Der Zeitherr, ein Wesen aus den Tiefen des Weltraums, hat einen komischen Wettbewerb ausgeschrieben, um etwas Zerstreuung in sein unendlich langes Leben zu bringen:

Den STARTEST.

Nur wenige können die Prüfungen bestehen, aber wer es schafft, dem winkt eine Verlängerung seiner Lebensspanne.

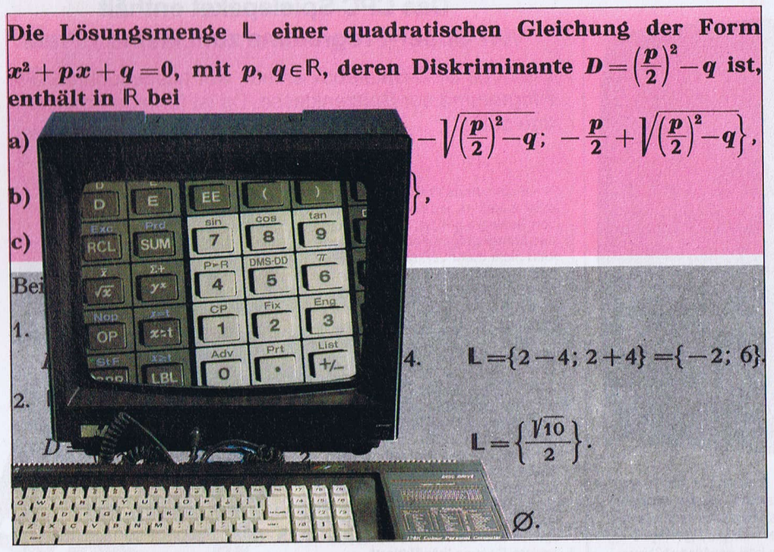
für CPC 464 · 664 · 6128

Kassette **24,– DM***
Diskette 3" **29,– DM***

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 3,– DM bzw. für das Ausland 5,– DM Porto und Verpackung.

Bitte benutzen Sie den Coupon auf Seite 129.

DMV Verlag · Postfach 250 · Fuldaer Str. 6 · 3440 Eschwege



Die Fließkomma-Arithmetik des CPC

Interessantes über die CPCs 464/664/6128

Nachdem wir im Heft 7/87 einen Grundlagenartikel über die Integerarithmetik Ihres CPC auf den Seiten 42 – 45 gebracht haben, wollen wir uns nun an diesem Ort auch noch die Fließkomma-Routinen unseres Rechners vornehmen und auch wieder den einen oder anderen nützlichen Tip vom Stapel lassen.

Intro: The REAL thing

Die Impulse dazu stammen übrigens von einem sehr interessanten Artikel über eine Eigenimplementation von Fließkommas für den 6502-Prozessor aus der Ausgabe August 1976 (!) des amerikanischen Magazins "Dr. Dobbs Journal of Computer Calisthenics & Orthodontia". Geschrieben hat diesen Artikel (man lese und staune!) Steve Wozniak, der ehemalige Chef der Apple Computer Company.

For You: Ready To Use

Was auch schon für die Integer- Arithmetik gilt, ist hier nicht anders: Schwierige Routinen wollten uns die Systemprogrammierer des CPCs schon von Haus aus anbieten. Im ROM sind also bereits alle wichtigen Einsprünge vorhanden, die u.a. auch der Locomotive BASIC Interpreter benutzt.

Damit kommen wir wieder zu einem Berg von Daten und Informationen, auf die wir nun eingehen werden müssen, da diese weder im 'CPC Intern' von Data Becker noch im Firmwarehandbuch von Schneider genauer erklärt sind:

Im unteren ROM (Lower ROM) befindet sich vor den sogenannten Characters (Matrizen des Systemzeichensatzes) der FLO (Floating Point Pack), mit dem man die wesentlichen Probleme, die bei der Realisierung von Fließkommas auftauchen können, lösen kann. "Die wesentlichen" heißt hier, daß zum Beispiel Prozeduren für die Ein- und Ausgabe, somit auch für die formatierte Zahlenausgabe, fehlen. Im BASIC-ROM kann der interessierte User solche Routinen finden. Doch dazu später, keine Sorge!

Die Einsprünge

Der FLO Pack-Bereich beim CPC 464 geht von #2E18 (dezimal 11800) bis #3707 (dezimal 14087), sodann folgt hier der schon beschriebene Integer Pack. Bei CPC664 und CPC6128 liegt der FLO Pack ab der Adresse #2F91 (dezimal 12177) bis #37FF (dezimal 14335). Fast alle arithmetischen Funktionen, die hier im Betriebssystem-ROM stehen, können über eine Sprungtabelle im RAM angesprungen werden. Auf Wunsch können Sie also eigene Routinen direkt in das System einbinden. Sie brauchen nur den entsprechenden Vektor-Eintrag auf eine eigene Routine im RAM "verbiegen". Die nötigen Grundlagen dazu sollen später in unserem Artikel noch vermittelt werden.

Doch wo steht denn nun diese Tabelle? Beim CPC 464 geht sie von #BD3D (dezimal 48445) bis #BDA2 (dezimal 48546) und bei den beiden größeren Brüdern 664 und 6128 von #BD61 (dezimal 48481) bis #BDC1 (dezimal 48577). Daraus ergeben sich 32 Einsprünge (mit je 3 Bytes) für CPC 664/6128 und 35 Einsprünge für den CPC 464. Abbildung 1 zeigt eine Liste der Einsprünge mit ihren Namen (nach dem Data Becker-ROM Listing).

Aus dieser Tabelle wird ersichtlich, daß bei den neueren CPC-Versionen 664 und 6128 einiges umgestellt worden ist und zwei Einsprünge (zumindest im RAM) nun fehlen, nämlich "Zahl mit 2↑A multiplizieren" und eine zweite Subtraktionsroutine. Was es nun mit den einzelnen Einsprüngen genau auf sich hat, das haben wir später in Form einer weiteren Tabelle penibel zusammengestellt.

CPC Intern – eine Reise ins Innere

Aber über die Grundzüge dieser Arithmetik haben wir noch kein Wort verloren: Grund genug, dies nun schleunigst zu tun, bevor die ersten Leser beleidigt sind. Die Fließkomma-Arithmetik des Locomotive-BASIC Interpreters benutzt, wie schon gesagt, obige Routinen. Daraus lassen sich über die Eigenschaften mit Hilfe der täglichen Erfahrungen mit BASIC einige Aussagen machen: Diese FLO-Implementation hat maximal neun Nachkommastellen Genauigkeit und einen Rechenbereich von -1E38 bis +1E38 aufzuweisen. Trigonometrische Berechnungen lassen sich im Bogenmaß wie auch im Gradmaß (RAD und DEG) durchführen. Die Konstante PI ist, wie auch ein ausgeklügelter Zufallszahlengenerator schon implementiert. Auf beide kann der Benutzer auch in Assembler zugreifen.

Die Gleitkomma-Darstellung
Das Grundprinzip ist es, Dezimalbrüche in einem festen Format darzustellen: Zu der oben beschriebenen Präzision von neun Nachkommastellen sind pro Zahl fünf Bytes nötig. Davon wird ein Byte zur Speicherung des Exponenten (inklusive dessen Vorzeichen) verwendet.

Die Mantisse
Die restlichen vier Bytes stellen (abzüglich eines Bits für das Vorzeichen der Zahl) mit 31 Bits die normalisierte Mantisse dar. Wir haben eine Genauigkeit von (2↑31)-1. Eine Darstellung mit doppelter Genauigkeit (double precision), wie bei manchen MSX-Computern oder mit MBASIC von der Firma Microsoft, ist nicht möglich. Die Wertigkeit der Bytes 0 bis 3 steigt stetig an: Das Bit 0 des Bytes 0 stellt das Least Significant Bit (LSB), das Bit 6 von Byte 3 das Most Significant Bit (MSB) dar; das Bit 7 von Byte 3 speichert das Vorzeichen (0=positive Zahl, 1= negative Zahl!). Daß diese single precision- Darstellung so ihre Tücken hat, dürfte wohl klar sein. Deshalb sollte der folgende Rat auch wirklich ernst genommen werden: Bei Schleifen mit sehr kleinen Zählwerten sollte man nie auf Gleichheit prüfen (siehe PC International 9/85, Seite 66f).

Der Exponent
Und das Byte 4 schließlich stellt den Exponenten dar (Bit 7 enthält das Signum des Exponenten, die Bits 0 bis 6 stellen den eigentlichen Exponenten dar). Das gesetzte Signum-(Vorzeichen-)Bit des Exponenten bedeutet eine positive Hochzahl (und umgekehrt negativ). Man kann diese Hochzahl also ganz leicht berechnen:

```
Exponent=Offset+128 (hex. #80)
```

Daraus ergeben sich folgende Beispielwerte:

```
#00 repräsentiert E-127
#80 repräsentiert E0
#81 repräsentiert E+1
#82 repräsentiert E+2
#FF repräsentiert E+128
```

Und es ergibt sich ein maximaler Darstellungsbereich von plus/minus 2E127 (dezimal 10↑38).

Binärcode-Darstellung: ein Beispiel
Als Beispiel nehmen wir die Fließkommazahl 0.01 und wir wollen diese nun in Binärcode-Schreibweise darstellen:

Es ergeben sich die fünf hexadezimalen Bytes 3D 0A D7 23 7A. Und in binärer Schreibweise:

```
- Byte 0 (LSB) : 00111101
- Byte 1 : 00001010
- Byte 2 : 11010111
- Byte 3 (MSB) : 00100011
- Byte4 (Exponent) :01111010
```

Und wie kommt man von hier zu der Zahl 0.01? Die Zahl berechnet sich wie folgt:

```
0.01 = (2↑(-2)+2↑(-6)+2↑.(-7)+2↑(-8)+
       2↑(-9)+2↑(-1)+2↑(-13)+2↑(-14)+
       2↑(-15)+ 2↑(-20)+ 2↑(-22)+ 2↑
       (-26)+2↑(-27)+2↑(-28)+2↑(-29)+
       2↑(-31)+1)* 2↑(-7)
```

wobei das Gleichheitszeichen im Prinzip durch ein Ungefährzeichen ersetzt werden müßte, denn mathematisch sind die Ausdrücke links und rechts vom Gleichheitszeichen meist nur annähernd gleich!

Fließkomma-Analyse
Zur Analyse von Fließkommazahlen auf der Schneider-CPC-Serie kann folgendes Programm dienen:

```
1000 'FLO examiner
1010 'ER 29.06.1987
1020:
1030 INPUT"Zahl=";z
1040 '"Klammeraffe"=FLO-Adresse
1050 address=@z
1060 FORi=0TO4
1070 p(i)=PEEK(address+i)
1080 NEXTi
1090 PRINT"Zahl ist";
1100 PRINT(1-2*SGN(p(3)AND128))*2↑(p
     (4)-129)*(1+((p(3)AND127)+(p(2)
     +(p(1)+p(0)/256)/256)/256)/128)
1110 PRINT"Aufbau:"
1120 FOR i=0 TO 3
1130 PRINT BIN$(p(i),8);" ";
1140 NEXTi
1150 PRINT"Exponent: ";
1160 PRINT BIN$(p(4),8)
1170 RUN
```

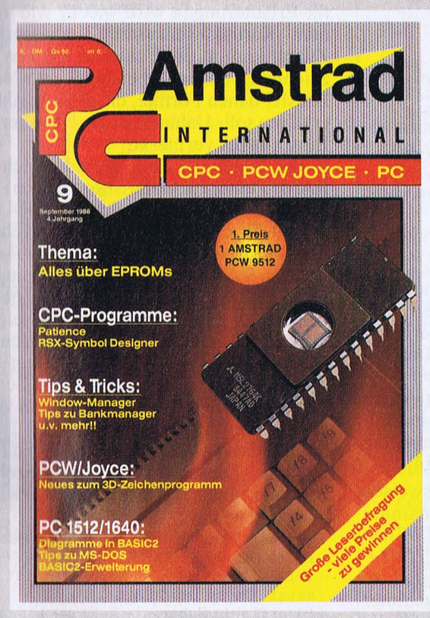
Dieses Programm zeigt auch, wie die interne Darstellung zur Ausgabe aufbereitet wird (Zeile 1100!). Der interne Aufbau wird anschließend im formatierten 8-Bit-Binärformat auf dem Bildschirm dargestellt.

Wir outputten in Assembler...

Wie Sie sehen, ist die Gleitkomma-Aufbereitung alleine schon ein schönes Stück Arbeit. Deshalb soll an dieser Stelle ein Tip folgen, wie man Fließkommazahlen nach ASCII wandelt, die dann ausgegeben werden können:

```
1000
1010 ; FLO->ASCII adaption CPC464
1020 ; STR$
1030 ; upper rom öffnen
1040 FLOASC: CALL #B900
1050 ; adresse fließkomma
1060 LD HL,FLO1
1070 ; adresse string
1080 LD BC,FLO2
1090 ; umwandlung in ascii
1100 ; bei 6128: #e393 !
1110 JP #E2C8
```

PC AMSTRAD International



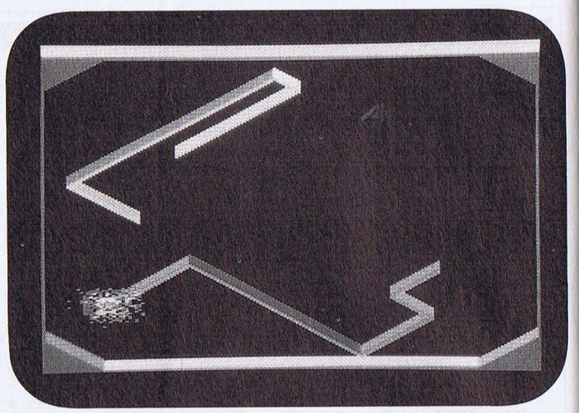
– die Zeitschrift, die Sie lesen sollten, wenn Sie einen AMSTRAD CPC, PCW oder PC besitzen.
Jeden Monat erhalten Sie aktuelle Informationen, interessante Anwendungsprogramme, spannende und ansprechend gestaltete Spiele, sowie eine Menge an Programmiertips und -tricks.

PC AMSTRAD International – für alle, die mehr für Ihren Computer wollen.

DMV Daten- und Medienverlag

DMV · Postfach 250 · Fuldaer Str. 6 · 3440 Eschwege

TRON 3D



3D-Perspektive überzeugend dargestellt.

Ein actionreiches Spiel mit dreidimensionaler Grafik und Stereosound

"Oh nein, nicht schon wieder TRON!" werden jetzt wahrscheinlich einige von Ihnen denken. Aber keine Angst, bei TRON 3D sieht und hört man wesentlich mehr als zwei Linien, die sich mit undefinierbaren Geräuschen entweder nervtötend langsam oder unspielbar schnell Über den Bildschirm quälen.

Schon der Name des Spieles verrät, daß es sich hier um etwas Besonderes handeln muß. Durch geschickten Einsatz der 16 Farben des CPC entsteht eine dreidimensionale Grafik, die bei diesem Spiel bisher ihresgleichen sucht. Und was einem hier für ein Sound geboten wird, kann sich auch hören lassen: ein Stereosound (für jeden Spieler ein eigener Kanal), der realistischer nicht sein könnte.

Für die, die das Spielprinzip noch nicht kennen, sei es hier nochmal erklärt: das Spielfeld ist eine Arena, die durch Mauern abgegrenzt wird. Zwei Spieler gleichzeitig treten gegeneinander an. Ihre Startpositionen und Fahrtrichtungen werden durch Zufall ausgewählt. Sie bewegen sich nun das ganze Spiel über und lassen sich nicht bremsen. Dabei hinterlassen sie dort, wo sie schon gefahren sind, eine Mauer, so daß sie also eine Spur zeichnen. Sie können sich dabei in vier Richtungen bewegen, die durch vorher definierte Tasten angewählt werden. Aufgabe ist es nun, zu versuchen, nicht in eine Mauer zu fahren und gleichzeitig den Gegner dazu zu zwingen, diesen Fehler zu begehen. Auch die Begrenzungsmauer darf nicht angefahren werden. Achtung: man darf auch nicht auf seiner eigenen Spur zurückfahren; wenn Sie also z.B. gerade nach links unten gefahren sind, dürfen sie nicht gleich danach wieder nach rechts oben fahren.

Wenn Sie das Spiel mit RUN"TRON" gestartet haben, dann wird das Hauptmenü aufgebaut. Sie können wählen zwischen "Spiel beginnen", "Tasten definieren" und "Geschwindigkeit". Durch Drücken der entsprechenden Zifferntaste für die Geschwindigkeit können Sie umschalten zwischen "langsam", "mittel" und "schnell". Wenn Sie "Tasten definieren" wählen, erscheint zuerst in der Farbe des ersten Spielers das Definitionsmenü. Sie sehen die jeweilige Fahrtrichtung angezeigt. Darunter sehen Sie die bisher eingestellten Tasten. In der untersten Zeile erscheint nun ein Cursor, und sie müssen die gewünschten Tasten für die durch die Pfeile in der obersten Zeile angegebene Fahrtrichtung drücken. Danach wechselt das Menü in die Farbe des zweiten Spielers, und dieser muß ebenso seine Tasten eingeben. Als zweckmäßig hat sich erwiesen: Spieler 1: <a>, <l>, <m>, <x> Spieler 2: <f4>,<f9>,<f5>,<f2> Dann entspricht die Lage der Tasten in ihrem Verhältnis zueinander genau den vier Fahrtrichtungen, und Sie können beim Spielen die Finger auf diesen Tasten liegen lassen - wichtig für ein schnelles Reagieren!

Wenn Sie nun "Spiel beginnen" wählen, wird das Spielfeld aufgebaut, und die Aufforderung "Fertigmachen" erscheint. Durch Tastendruck startet dann das Spiel, und Sie beginnen sich von der vom Computer ausgewählten Position in der vom Computer ausgewählten Richtung zu bewegen. Der Fahrer mit der hellen Oberkante ist Spieler 1, der mit der dunklen ist Spieler 2. Jetzt können Sie spielen!

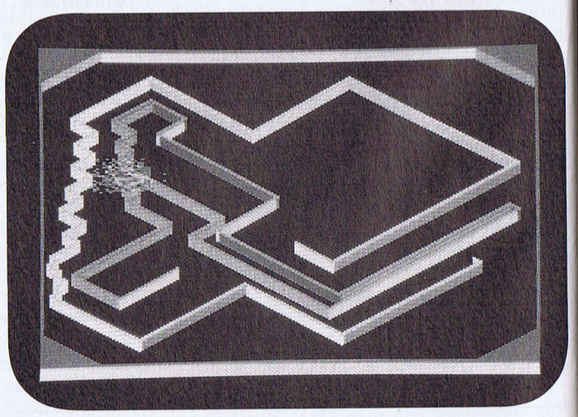
Baut einer der Spieler einen Unfall, dann gibt es eine farbenprächtige Explosion, und auf Tastendruck wird das Bild "zersetzt". Sie sehen dann eine Zwischenbilanz und können wählen, ob Sie noch ein Spiel wagen möchten (mit Leertaste) oder ob Sie ins Hauptmenü zurück möchten (mit <ESC>). Im letzteren Fall wird der Zähler zurückgesetzt.

Vor dem Vergnügen steht aber noch die Arbeit, und so muß auch erst noch das Spiel erstellt werden. Als erstes tippen Sie das Programm "TRON" ab und speichern es unter diesem Namen auf Ihrem Datenträger ab. Dann tippen Sie den Datalader "TRON.DAT" ab und speichern ihn ebenfalls (Kassettenbenutzer speichern ihn bitte auf einem anderen Band!).

Nun starten Sie ihn mit RUN, worauf das eigentliche Programm als Maschinencode erzeugt und automatisch abgespeichert wird (Kassettenbenutzer müssen, bevor sie den Datalader starten, wieder die Kassette mit dem Startprogramm "TRON" einlegen, und zwar so, daß der erzeugte Maschinencode direkt dahinter abgespeichert wird!). Jetzt können Sie das Spiel mit RUN"TRON" starten und sich wilde Schlachten liefern!

Viel Spaß!

(Sebastian Belzner/sr)



Tron 3D macht einfach Spaß.

Für 464-664-6128

```
10 MEMORY &8FFF                                   [207]
20 LOAD"tron.bin",&9000                           [2088]
30 CALL &9000                                     [445]
```

```
1000 '************************************ [1460]
1010 '* TRON.HEX - DATA-Lader von 'CPC' * [2006]
1020 '************************************ [1460]
1030 '                                    [117]
1040 DATA 9000,CD,5C,96,21,6A,98,06,10,3E, [3112]
00,E5,C5,F5,46,48,CD,757D
1050 DATA 9010,32,BC,F1,C1,E1,23,3C,10,F1, [2880]
01,00,00,CD,38,BC,3E,A32D
1060 DATA 9020,01,CD,0E,BC,21,00,00,22,80, [3206]
98,22,82,98,3E,03,CD,84AB
1070 DATA 9030,90,BB,21,29,9C,06,DC,7E,23, [3889]
CD,5A,BB,10,F9,06,DC,6FE1
1080 DATA 9040,7E,23,CD,5A,BB,10,F9,3A,E1, [2408]
9D,C3,71,90,CD,18,BB,DFA0
1090 DATA 9050,FE,31,CA,D2,91,FE,32,CA,AB, [3257]
90,FE,33,CA,62,90,C3,952B
1100 DATA 9060,4D,90,3E,03,CD,90,BB,3A,E1, [1842]
9D,3C,FE,03,20,02,3E,4B02
1110 DATA 9070,00,32,E1,9D,21,00,12,22,68, [2837]
98,21,E2,9D,FE,00,28,6120
1120 DATA 9080,16,21,00,08,22,68,98,21,E9, [2444]
9D,FE,01,28,09,21,00,2C34
1130 DATA 9090,03,22,68,98,21,F0,9D,E5,21, [2392]
08,16,CD,75,BB,E1,06,8B1E
1140 DATA 90A0,07,7E,23,CD,5A,BB,10,F9,C3, [3239]
4D,90,3E,01,CD,90,BB,3493
1150 DATA 90B0,21,0D,01,CD,75,BB,21,F7,9D, [3078]
06,F0,7E,23,CD,5A,BB,95DD
1160 DATA 90C0,10,F9,21,0E,1D,CD,75,BB,3E, [4768]
31,CD,5A,BB,2A,97,97,06B8
1170 DATA 90D0,ED,5B,99,97,CD,23,91,22,97, [2524]
97,ED,53,99,97,3E,02,8447
1180 DATA 90E0,CD,90,BB,21,0D,01,CD,75,BB, [3068]
21,F7,9D,06,F0,7E,23,3289
1190 DATA 90F0,CD,5A,BB,10,F9,21,0E,1D,CD, [3897]
75,BB,3E,32,CD,5A,BB,408A
1200 DATA 9100,2A,A0,97,ED,5B,A2,97,CD,23, [2818]
91,22,A0,97,ED,53,A2,BFA8
1210 DATA 9110,97,21,0D,01,CD,75,BB,06,F0, [3535]
3E,20,CD,5A,BB,10,F9,C3F1
1220 DATA 9120,C3,4D,90,E5,21,10,0F,CD,75, [2509]
BB,E1,E5,7D,CD,2A,BB,0D6C
1230 DATA 9130,CD,5A,BB,CD,9A,91,7A,CD,2A, [2119]
BB,CD,5A,BB,CD,9A,91,3DE4
1240 DATA 9140,E1,7C,CD,2A,BB,CD,5A,BB,CD, [1928]
9A,91,7B,CD,2A,BB,CD,3199
1250 DATA 9150,5A,BB,21,11,0F,CD,75,BB,3E, [3169]
5F,CD,5A,BB,3E,08,CD,FC38
1260 DATA 9160,5A,BB,CD,A4,91,6F,CD,9A,91, [3126]
3E,5F,CD,5A,BB,3E,08,546C
1270 DATA 9170,CD,5A,BB,CD,A4,91,57,CD,9A, [3661]
91,3E,5F,CD,5A,BB,3E,DA54
1280 DATA 9180,08,CD,5A,BB,CD,A4,91,67,CD, [2629]
9A,91,3E,5F,CD,5A,BB,5D31
1290 DATA 9190,3E,08,CD,5A,BB,CD,A4,91,5F, [3233]
C9,06,04,3E,20,CD,5A,28F6
1300 DATA 91A0,BB,10,F9,C9,E5,D5,C5,3E,00, [2512]
06,50,F5,CD,1E,BB,20,FBD9
1310 DATA 91B0,09,F1,3C,10,F6,C1,D1,E1,18, [2535]
EA,F1,F5,CD,2A,BB,CD,97B0
1320 DATA 91C0,5A,BB,F1,C1,D1,21,00,80,22, [3633]
FC,94,F5,CD,FB,94,F1,4B9A
1330 DATA 91D0,E1,C9,11,DC,05,ED,53,AD,97, [3088]
ED,53,B6,97,2A,68,98,C2D6
1340 DATA 91E0,22,FC,94,3E,64,CD,19,93,C6, [2816]
32,5F,16,00,CB,03,CB,B3DE
1350 DATA 91F0,12,ED,53,9E,97,3E,64,CD,19, [1629]
93,C6,32,5F,16,00,CB,B4FE
1360 DATA 9200,03,CB,12,ED,53,A7,97,3E,64, [2466]
CD,19,93,C6,32,5F,16,C93E
1370 DATA 9210,00,CB,03,CB,12,ED,53,9C,97, [3100]
3E,64,CD,19,93,C6,9B,49FE
1380 DATA 9220,5F,16,00,CB,03,CB,12,ED,53, [2629]
```

Listing "TRON 3D"

```
A5,97,3E,03,CD,19,93,5F68
1390 DATA 9230,C6,01,32,9B,97,3E,03,CD,19, [3061]
93,C6,01,32,A4,97,3E,09E8
1400 DATA 9240,00,CD,0E,BC,01,50,00,11,0C, [2239]
98,21,BC,97,ED,B0,06,30B2
1410 DATA 9250,06,C5,21,0C,98,CD,D4,92,C1, [3204]
10,F6,06,21,C5,21,34,167A
1420 DATA 9260,98,CD,D4,92,C1,10,F6,21,00, [2625]
00,11,00,00,CD,C0,BB,6544
1430 DATA 9270,3E,0C,CD,DE,BB,06,03,21,00, [1964]
00,11,80,02,C5,CD,F9,AF4A
1440 DATA 9280,BB,11,00,00,21,02,00,CD,C3, [3068]
BB,11,7F,FD,21,00,00,40A4
1450 DATA 9290,CD,F9,BB,21,02,00,11,00,00, [4020]
CD,C3,BB,C1,10,D8,21,A504
1460 DATA 92A0,00,00,11,80,02,CD,F9,BB,3E, [1939]
0F,CD,90,BB,21,0C,05,4429
1470 DATA 92B0,CD,75,BB,21,5C,98,06,0C,7E, [2048]
23,CD,5A,BB,10,F9,CD,1F09
1480 DATA 92C0,18,BB,21,0C,05,CD,75,BB,06, [3754]
0C,3E,20,CD,5A,BB,10,6E83
1490 DATA 92D0,F9,C3,5F,93,06,04,C5,5E,23, [2575]
56,23,4E,23,46,2B,0B,6092
1500 DATA 92E0,0B,71,23,70,03,03,23,7E,CD, [2574]
DE,BB,E5,60,69,CD,F6,C7E2
1510 DATA 92F0,BB,E1,23,C1,10,E0,06,04,C5, [3119]
5E,23,56,23,4E,23,46,CC4E
1520 DATA 9300,2B,03,03,71,23,70,0B,0B,23, [2209]
7E,CD,DE,BB,E5,60,69,E9B7
1530 DATA 9310,CD,F6,BB,E1,23,C1,10,E0,C9, [3087]
D5,E5,F5,ED,5F,6F,67,E8DD
1540 DATA 9320,CD,3D,93,ED,5B,95,97,19,7C, [2332]
B7,20,01,24,7D,B7,20,DD3E
1550 DATA 9330,01,2C,22,95,97,F1,6F,CD,4D, [2616]
93,E1,D1,C9,C5,D5,5D,E8DA
1560 DATA 9340,16,00,6A,40,29,30,01,19,10, [3150]
FA,D1,C1,C9,C5,AF,06,F7B7
1570 DATA 9350,08,CB,14,17,95,30,01,85,3F, [2464]
10,F6,CB,14,C1,C9,2A,D4DD
1580 DATA 9360,97,97,ED,5B,99,97,CD,EC,93, [2733]
FE,00,28,0B,78,32,9B,C7B6
1590 DATA 9370,97,11,14,05,ED,53,AD,97,3A, [4203]
9B,97,2A,9E,97,ED,5B,D5E7
1600 DATA 9380,9C,97,CD,13,94,22,9E,97,ED, [3667]
53,9C,97,47,3E,01,CD,A9C9
1610 DATA 9390,3B,94,CD,FB,94,2A,A0,97,ED, [3921]
5B,A2,97,CD,EC,93,FE,365B
1620 DATA 93A0,00,28,0B,78,32,A4,97,11,14, [2709]
05,ED,53,B6,97,3A,A4,CBE2
1630 DATA 93B0,97,2A,A7,97,ED,5B,A5,97,CD, [3430]
13,94,22,A7,97,ED,53,3D56
1640 DATA 93C0,A5,97,47,3E,02,CD,3B,94,CD, [2556]
FB,94,2A,AD,97,2B,3E,C943
1650 DATA 93D0,03,BC,20,03,21,84,03,22,AD, [2589]
97,2A,B6,97,2B,3E,03,F522
1660 DATA 93E0,BC,20,03,21,84,03,22,B6,97, [2786]
C3,5F,93,7C,06,03,E5,D4DB
1670 DATA 93F0,CD,1E,BB,E1,C0,7D,06,04,E5, [2946]
CD,1E,BB,E1,C0,7A,06,C135
1680 DATA 9400,01,E5,CD,1E,BB,E1,C0,7B,06, [4330]
02,E5,CD,1E,BB,E1,C0,37C8
1690 DATA 9410,3E,00,C9,FE,01,20,07,23,23, [3098]
13,13,13,13,C9,FE,02,715B
1700 DATA 9420,20,07,2B,2B,13,13,13,13,C9, [3796]
FE,03,20,07,2B,2B,1B,A8AF
1710 DATA 9430,1B,1B,1B,C9,23,23,1B,1B,1B, [3353]
1B,C9,F5,C5,2B,2B,2B,12CC
1720 DATA 9440,2B,E5,23,23,23,23,D5,CD,DE, [3397]
BB,E5,D5,CD,F0,BB,FE,373A
1730 DATA 9450,01,CA,05,95,FE,02,CA,05,95, [3024]
FE,03,CA,05,95,D1,E1,E3C2
1740 DATA 9460,E5,D5,CD,EA,BB,D1,E1,2B,2B, [1715]
E5,D5,CD,F0,BB,FE,01,8B76
1750 DATA 9470,CA,05,95,FE,02,CA,05,95,FE, [2315]
03,CA,05,95,D1,E1,D5,C751
1760 DATA 9480,E5,CD,EA,BB,E1,D1,2B,2B,06, [1636]
06,3E,00,C5,E5,D5,F5,00F0
1770 DATA 9490,CD,F0,BB,FE,03,28,5E,FE,02, [2370]
28,5A,FE,01,28,56,F1,F517
1780 DATA 94A0,3C,D1,E1,2B,2B,C1,10,E4,FE, [2517]
00,20,05,D1,E1,C1,F1,3719
1790 DATA 94B0,C9,3D,D1,E1,CB,3A,CB,1B,CB, [4503]
3A,CB,1B,CB,3C,CB,1D,B98B
1800 DATA 94C0,F5,7D,C1,48,06,00,91,4F,F1, [2868]
32,A9,97,F1,FE,02,28,5174
```

Listing "TRON 3D"

```
1810 DATA 94D0,04,3E,03,18,02,3E,07,C5,E5, [3437]
21,A9,97,46,80,D5,CD,24CB
1820 DATA 94E0,2C,BC,D1,C1,E1,CD,62,BC,21, [2252]
AA,97,CD,AA,BC,21,B3,574A
1830 DATA 94F0,97,CD,AA,BC,C9,F1,C1,C1,C1, [3838]
18,AD,01,00,09,0B,3E,FC5E
1840 DATA 9500,00,B8,20,FA,C9,CD,C6,BB,06, [4112]
0F,23,1B,1B,1B,10,FA,EE5F
1850 DATA 9510,CB,3C,CB,1D,CB,3A,CB,1B,CB, [3310]
3A,CB,1B,3E,18,BD,38,7E77
1860 DATA 9520,04,28,02,2E,18,3E,C7,BD,30, [3208]
02,2E,C7,3E,F0,BB,30,B94D
1870 DATA 9530,02,1E,00,3E,8A,BB,30,02,1E, [1891]
8A,16,00,CD,1D,BC,E5,96C2
1880 DATA 9540,CD,A7,BC,3E,01,21,8D,98,CD, [3039]
BC,BC,21,84,98,CD,AA,753A
1890 DATA 9550,BC,21,00,00,22,FC,94,E1,E5, [2926]
11,9D,98,CD,0F,96,CD,0200
1900 DATA 9560,FB,94,E1,E5,11,BD,99,CD,0F, [2958]
96,CD,FB,94,E1,11,DD,4082
1910 DATA 9570,9A,CD,0F,96,CD,FB,94,CD,03, [2831]
BB,CD,18,BB,CD,F0,95,918B
1920 DATA 9580,3E,01,CD,0E,BC,F1,F1,F1,F1, [3259]
F1,3E,01,CD,90,BB,21,716E
1930 DATA 9590,7D,96,06,C8,7E,CD,5A,BB,23, [3327]
10,F9,21,01,12,CD,75,8208
1940 DATA 95A0,BB,F1,F5,C6,30,CD,5A,BB,F1, [2843]
FE,01,20,06,21,80,98,FF7D
1950 DATA 95B0,34,18,04,21,82,98,34,21,05, [2261]
01,CD,75,BB,2A,80,98,B1C0
1960 DATA 95C0,CD,2A,96,21,05,1D,CD,75,BB, [2451]
2A,82,98,CD,2A,96,21,2A70
1970 DATA 95D0,07,01,CD,75,BB,21,45,97,06, [2853]
50,7E,CD,5A,BB,23,10,6562
1980 DATA 95E0,F9,CD,18,BB,FE,20,CA,D2,91, [3414]
FE,FC,CA,1F,90,18,F1,2018
1990 DATA 95F0,11,00,C0,06,06,DD,21,7A,98, [4281]
62,6B,36,00,23,23,23,C67A
2000 DATA 9600,23,23,23,7C,B7,20,F4,DD,5E, [3756]
00,DD,23,10,EB,C9,06,D936
2010 DATA 9610,18,C5,06,0C,1A,13,AE,77,23, [2566]
10,F9,CD,26,BC,B7,D5,F337
2020 DATA 9620,11,0C,00,ED,52,D1,C1,10,E8, [1719]
C9,F5,D5,E5,11,10,27,44DE
2030 DATA 9630,CD,4E,96,11,E8,03,CD,4E,96, [3724]
11,64,00,CD,4E,96,11,0DCD
2040 DATA 9640,0A,00,CD,4E,96,1E,01,CD,4E, [3591]
96,E1,D1,F1,C9,AF,ED,51B3
2050 DATA 9650,52,38,03,3C,18,F9,19,C6,30, [2242]
C3,5A,BB,3E,FE,21,09,5A54
2060 DATA 9660,9C,CD,A8,BB,3E,FD,21,11,9C, [3010]
CD,A8,BB,3E,FC,21,19,C6CA
2070 DATA 9670,9C,CD,A8,BB,3E,FB,21,21,9C, [3118]
CD,A8,BB,C9,20,20,20,EA39
2080 DATA 9680,20,20,20,20,20,20,53,70,69, [3499]
65,6C,65,72,20,20,2C,7373
2090 DATA 9690,20,50,65,63,68,20,67,65,68, [2383]
61,62,74,21,20,20,20,DDA5
2100 DATA 96A0,20,20,20,20,20,20,20,20,20, [2311]
20,20,20,20,20,20,20,FFFF
2110 DATA 96B0,20,20,20,20,20,20,20,20,20, [3089]
20,20,20,20,20,20,20,FFFF
2120 DATA 96C0,20,20,20,20,20,20,20,20,20, [3198]
20,20,20,20,50,6C,61,FEE6
2130 DATA 96D0,79,65,72,20,31,3A,20,20,20, [3018]
20,20,20,20,20,20,20,9FC8
2140 DATA 96E0,20,20,20,20,20,20,20,20,20, [2919]
50,6C,61,79,65,72,20,ED17
2150 DATA 96F0,32,3A,20,20,20,20,20,20,20, [3608]
20,20,20,20,20,20,20,7FF0
2160 DATA 9700,20,20,20,20,20,20,20,20,20, [2538]
20,20,20,20,20,20,20,FFFF
2170 DATA 9710,20,20,20,20,20,20,20,20,20, [2258]
20,20,20,20,30,30,30,FF8F
2180 DATA 9720,30,30,20,43,72,61,73,68,73, [3355]
20,20,20,20,20,20,20,9C76
2190 DATA 9730,20,20,20,20,20,20,20,20,20, [2857]
30,30,30,30,20,43,F85C
2200 DATA 9740,72,61,73,68,73,20,20,20,20, [2843]
20,20,20,20,20,20,53,C7B9
2210 DATA 9750,50,43,20,3D,20,4E,6F,63,68, [2967]
20,27,6E,65,20,52,75,A846
2220 DATA 9760,6E,64,65,20,20,20,20,20,20, [3922]
20,20,20,20,20,20,20,5FC1
2230 DATA 9770,20,20,20,20,20,20,20,20,45,53, [2342]
43,20,3D,20,45,6E,64,BB23
```

Listing "TRON 3D"

```
2240 DATA 9780,65,20,64,65,72,20,53,61,69, [2596]
73,6F,6E,20,20,20,20,A593
2250 DATA 9790,20,20,20,20,20,65,89,08,02, [2483]
01,00,00,00,00,00,00,8F5F
2260 DATA 97A0,45,26,3F,24,00,00,00,00,00, [2952]
00,0A,00,00,14,05,00,A134
2270 DATA 97B0,0F,05,00,11,00,00,14,05,00, [1939]
0F,05,00,00,00,36,00,FE0B
2280 DATA 97C0,0B,4E,00,10,00,0B,30,02,10, [3332]
00,0C,7E,02,36,00,0D,40B2
2290 DATA 97D0,7E,02,5A,01,0D,30,02,80,01, [3256]
0D,4E,00,80,01,0C,00,F228
2300 DATA 97E0,00,5A,01,0B,00,00,36,00,03, [3435]
4E,00,10,00,03,30,02,6F78
2310 DATA 97F0,10,00,03,7E,02,36,00,03,7E, [2048]
02,66,01,03,30,02,8C,780F
2320 DATA 9800,01,03,4E,00,8C,01,03,00,00, [2442]
66,01,03,00,00,00,00,FB9D
2330 DATA 9810,00,00,00,00,00,00,00,00,00, [2043]
00,00,00,00,00,00,00,0000
2340 DATA 9820,00,00,00,00,00,00,00,00,00, [3091]
00,00,00,00,00,00,00,0000
2350 DATA 9830,00,00,00,00,00,00,00,00,00, [2552]
00,00,00,00,00,00,00,0000
2360 DATA 9840,00,00,00,00,00,00,00,00,00, [3189]
00,00,00,00,00,00,00,0000
2370 DATA 9850,00,00,00,00,00,00,00,00,00, [2810]
00,00,00,46,45,52,54,03D4
2380 DATA 9860,49,47,4D,41,43,48,45,4E,00, [3118]
08,00,18,0B,06,0A,0C,0FE3
2390 DATA 9870,0E,10,12,14,08,16,0E,10,1A, [3320]
1A,03,05,02,01,04,00,1FAC
2400 DATA 9880,00,00,00,00,01,01,00,BC,02, [2640]
0F,0F,C8,00,04,01,7F,BFCD
2410 DATA 9890,02,00,10,05,7F,FF,14,01,FF, [2488]
14,00,00,00,00,00,00,0783
2420 DATA 98A0,00,00,00,00,00,00,00,00,00, [2790]
00,00,00,00,00,00,00,0000
2430 DATA 98B0,00,00,00,00,00,00,00,00,00, [2376]
00,00,00,00,00,00,00,0000
2440 DATA 98C0,00,00,00,44,00,00,88,00,00, [2787]
88,00,00,00,00,00,00,7205
2450 DATA 98D0,00,88,00,88,00,00,00,00,00, [2885]
00,00,88,00,00,00,00,88AA
2460 DATA 98E0,44,00,00,00,00,00,00,00,44,00, [2127]
88,44,00,44,00,44,00,6C0A
2470 DATA 98F0,00,00,00,00,88,88,CC,88,8C, [3643]
88,00,00,00,00,44,44,14CB
2480 DATA 9900,4C,CC,8C,CC,CC,CC,00,00,00, [2180]
00,00,88,CC,8C,CC,8C,1DC9
2490 DATA 9910,8C,CC,88,00,00,00,00,44,CC, [2737]
8C,4C,0C,4C,CC,88,00,0864
2500 DATA 9920,00,00,00,CC,CC,0C,0C,0C,4C, [3335]
8C,88,00,00,00,00,CC,90C6
2510 DATA 9930,CC,0C,0C,48,0C,CC,CC,00,44, [3731]
00,00,CC,4C,8C,0C,48,E6A2
2520 DATA 9940,4C,CC,88,00,00,00,44,44,CC, [2326]
0C,48,0C,0C,CC,88,00,A284
2530 DATA 9950,00,00,00,CC,8C,0C,84,0C,0C, [3150]
4C,CC,00,00,00,88,CC,9955
2540 DATA 9960,CC,0C,48,0C,0C,4C,4C,00,44, [2900]
00,00,44,CC,8C,0C,0C,2A69
2550 DATA 9970,4C,CC,CC,00,00,00,00,88,CC, [3080]
8C,0C,8C,CC,CC,88,00,400C
2560 DATA 9980,00,00,00,44,CC,4C,4C,CC,CC, [1957]
4C,88,00,00,00,44,44,20CF
2570 DATA 9990,CC,CC,8C,CC,CC,CC,00,00,00, [3763]
00,00,CC,CC,4C,CC,4C,1A09
2580 DATA 99A0,8C,CC,00,00,00,00,00,44,CC, [2669]
CC,44,CC,CC,88,00,00,1175
2590 DATA 99B0,00,00,00,00,88,44,00,00,88, [1837]
00,44,00,00,00,00,00,1C85
2600 DATA 99C0,00,00,00,00,00,00,00,00,00, [3457]
00,00,00,00,00,00,00,0000
2610 DATA 99D0,00,00,00,00,00,00,00,00,00, [2459]
00,00,00,00,00,00,00,0000
2620 DATA 99E0,00,00,88,00,44,00,00,44,00, [2428]
00,00,00,00,00,00,44,6457
2630 DATA 99F0,00,44,00,00,00,00,00,00,00, [2458]
00,00,00,88,88,00,00,0671
2640 DATA 9A00,00,00,00,00,00,00,00,44,44, [2531]
00,00,00,00,88,00,00,6420
2650 DATA 9A10,00,88,00,44,CC,88,44,4C,00, [2613]
00,00,00,00,88,00,88,C68A
2660 DATA 9A20,CC,4C,00,44,88,00,00,00,00, [2780]
44,88,88,4C,88,00,44,08F1
```

Listing "TRON 3D"


```
2670 DATA 9A30,44,00,00,00,00,44,00,44,44, [3178]
04,00,88,44,00,00,00,7D83
2680 DATA 9A40,00,00,CC,8C,00,88,04,88,4C, [2937]
00,00,00,00,44,88,CC,C6DF
2690 DATA 9A50,4C,00,04,08,CC,88,00,00,00, [3533]
44,8C,00,08,88,00,CC,420E
2700 DATA 9A60,CC,00,00,00,00,88,44,04,04, [3494]
04,04,08,44,00,00,00,AD44
2710 DATA 9A70,00,00,88,00,08,04,04,0C,44, [2940]
88,00,00,00,44,88,00,5411
2720 DATA 9A80,08,00,00,00,8C,88,00,00,00, [2667]
00,CC,00,8C,88,00,00,5FC2
2730 DATA 9A90,44,88,00,00,00,44,44,CC,00, [3453]
00,00,88,CC,00,00,00,5AE1
2740 DATA 9AA0,00,00,00,00,00,00,00,8C,CC, [2584]
00,00,00,00,88,CC,44,E9FC
2750 DATA 9AB0,88,08,00,88,88,00,00,00,00, [3874]
44,88,8C,CC,88,CC,08,CD5A
2760 DATA 9AC0,88,00,00,00,00,00,88,44,88, [3190]
44,00,44,00,00,00,00,0505
2770 DATA 9AD0,00,00,44,00,00,00,88,00,00, [3890]
00,00,00,00,00,00,00,9009
2780 DATA 9AE0,00,00,00,88,00,00,00,00,00, [2959]
00,00,00,00,00,00,00,8008
2790 DATA 9AF0,00,00,00,00,00,00,00,00,00, [3428]
00,00,00,00,00,00,00,0000
2800 DATA 9B00,00,00,00,00,00,00,00,00,00, [3654]
00,00,00,00,00,88,44,0154
2810 DATA 9B10,44,00,00,00,88,00,00,00,00, [1815]
00,00,00,00,00,88,00,4136
2820 DATA 9B20,00,88,00,00,00,00,00,88,00, [2973]
88,00,00,00,00,44,00,AAAA
2830 DATA 9B30,00,00,00,00,08,04,00,44,00, [2696]
00,00,00,00,00,44,00,1488
2840 DATA 9B40,00,00,04,00,00,08,88,00,00, [2688]
44,00,00,04,00,00,04,A125
2850 DATA 9B50,00,00,00,00,00,00,00,04,00, [3552]
04,00,08,00,00,00,00,0580
2860 DATA 9B60,00,00,44,00,00,00,40,00,00, [3337]
44,00,00,00,00,00,00,1108
2870 DATA 9B70,04,80,00,00,04,00,44,00,00, [2775]
00,00,08,00,00,00,00,A8A2
2880 DATA 9B80,00,00,00,00,00,00,00,88,00, [3147]
00,80,04,00,88,00,00,9A60
2890 DATA 9B90,00,00,88,04,04,00,00,00,04, [1716]
00,88,00,00,00,00,00,7311
2900 DATA 9BA0,00,00,40,00,00,00,00,00,00, [2610]
00,00,00,00,80,00,00,0208
2910 DATA 9BB0,00,00,44,00,00,00,00,88,08, [2922]
00,00,08,8C,00,00,00,08E8
2920 DATA 9BC0,00,44,00,00,00,08,08,00,00, [2989]
44,00,00,00,00,00,00,2111
2930 DATA 9BD0,00,44,00,04,00,00,00,00,00, [2920]
00,00,00,00,00,04,00,4019
2940 DATA 9BE0,88,00,44,00,00,44,88,44,88, [3109]
44,00,00,00,00,00,00,914C
2950 DATA 9BF0,00,00,00,00,00,44,00,00,00, [2642]
00,44,88,00,00,00,00,1001
2960 DATA 9C00,00,00,00,08,00,00,00,00,00, [3070]
FE,FC,F8,F8,FC,CE,87,AAAB
2970 DATA 9C10,03,7F,3F,1F,1F,3F,73,E1,C0, [3278]
03,87,CE,FC,F8,F8,FC,2AD4
2980 DATA 9C20,FE,C0,E1,73,3F,1F,1F,3F,7F, [4229]
96,9A,9A,9A,9A,9A,9A,92A3
2990 DATA 9C30,9A,9A,9A,9A,9A,9A,9A,9A,9A, [1823]
9A,9A,9A,9A,9A,9A,9A,0000
3000 DATA 9C40,9A,9A,9A,9A,9A,9A,9A,9A,9A, [3254]
9A,9A,9A,9A,9A,9A,9A,0000
3010 DATA 9C50,9C,95,3E,3E,3E,3E,3E,3E,3E, [4104]
3E,3E,3E,3E,3E,53,55,3F35
3020 DATA 9C60,50,45,52,54,52,4F,4E,20,2D, [2817]
20,33,44,3C,3C,3C,3C,4DC1
3030 DATA 9C70,3C,3C,3C,3C,3C,3C,3C,3C,95, [4230]
93,9A,9A,9A,9A,9A,9A,67A2
3040 DATA 9C80,9A,9A,9A,9A,9A,9A,9A,9A,9A, [2331]
9A,9A,9A,9A,9A,9A,9A,0000
3050 DATA 9C90,9A,9A,9A,9A,9A,9A,9A,9A,9A, [2251]
9A,9A,9A,9A,9A,9A,9A,0000
3060 DATA 9CA0,99,20,20,20,20,20,20,20,20, [2899]
20,20,20,20,20,20,20,7FA3
3070 DATA 9CB0,20,20,20,20,20,20,20,20,20, [3060]
20,20,20,20,20,20,20,FFFF
3080 DATA 9CC0,20,20,20,20,20,20,20,20,20, [2772]
96,9A,9A,9A,9A,9A,9A,C869
3090 DATA 9CD0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [3125]
9A,9A,9A,9A,9A,9A,9A,0000
```

Listing "TRON 3D"

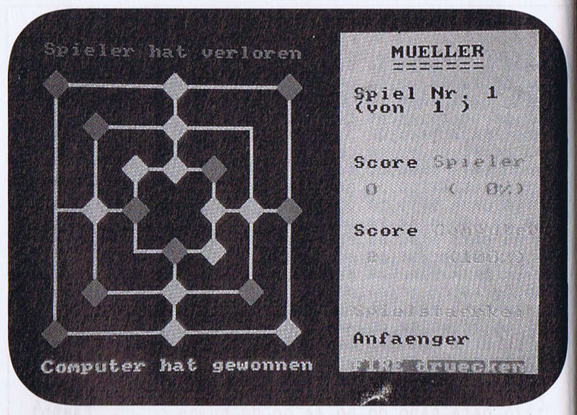
```
3100 DATA 9CE0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [3883]
9A,9A,9A,9A,9A,9A,9A,0000
3110 DATA 9CF0,9C,95,20,20,20,20,20,20,20, [2234]
20,20,31,20,2D,20,53,BEDB
3120 DATA 9D00,70,69,65,6C,20,62,65,67,69, [2642]
6E,6E,65,6E,20,20,20,2228
3130 DATA 9D10,20,20,20,20,20,20,20,20,95, [2805]
95,20,20,20,20,20,20,883F
3140 DATA 9D20,20,20,20,32,20,2D,20,54,61, [3090]
73,74,65,6E,20,64,65,A7D3
3150 DATA 9D30,66,69,6E,69,65,72,65,6E,20, [3690]
20,20,20,20,20,20,20,4BC0
3160 DATA 9D40,95,95,20,20,20,20,20,20,20, [2631]
20,20,33,20,2D,20,54,3EF8
3170 DATA 9D50,65,6D,70,6F,3A,20,4D,49,54, [4050]
54,45,4C,20,20,20,20,81A0
3180 DATA 9D60,20,20,20,20,20,20,20,20,95, [2667]
95,20,20,20,20,20,20,883F
3190 DATA 9D70,20,20,20,20,20,20,20,20,20, [2777]
20,20,20,20,20,20,20,FFFF
3200 DATA 9D80,20,20,20,20,20,20,20,20,20, [3060]
20,20,20,20,20,20,20,FFFF
3210 DATA 9D90,95,95,20,20,20,20,20,20,20, [2826]
20,20,42,69,74,74,65,3A5D
3220 DATA 9DA0,20,54,61,73,74,65,20,64,72, [3062]
75,65,63,6B,65,6E,21,2CED
3230 DATA 9DB0,20,20,20,20,20,20,20,20,95, [2134]
93,9A,9A,9A,9A,9A,9A,93A9
3240 DATA 9DC0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [4016]
9A,9A,9A,9A,9A,9A,9A,0000
3250 DATA 9DD0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [2589]
9A,9A,9A,9A,9A,9A,9A,0000
3260 DATA 9DE0,99,01,4C,41,4E,47,53,41,4D, [2633]
4D,49,54,54,45,4C,20,E1EE
3270 DATA 9DF0,53,43,48,4E,45,4C,4C,96,9A, [1382]
9A,9A,9A,9A,9A,9A,9A,C041
3280 DATA 9E00,9A,9A,9A,9A,9A,9A,9A,9A,9A, [3254]
9A,9A,9A,9A,9A,9A,9A,0000
3290 DATA 9E10,9A,9A,9A,9A,9A,9A,9A,9A,9A, [2435]
9A,9A,9A,9A,9A,9C,95,0003
3300 DATA 9E20,54,61,73,74,65,6E,20,64,65, [3354]
66,69,6E,69,65,72,65,F6A4
3310 DATA 9E30,6E,2C,20,53,70,69,65,6C,65, [2971]
72,20,20,3A,20,20,20,9B0F
3320 DATA 9E40,20,20,20,20,20,20,95,95,52, [2536]
69,63,68,74,75,6E,67,0471
3330 DATA 9E50,3A,20,20,20,20,FE,20,20,20, [2824]
20,FD,20,20,20,20,FC,9C8D
3340 DATA 9E60,20,20,20,20,FB,20,20,20,20, [3008]
20,20,20,20,20,95,95,2626
3350 DATA 9E70,20,20,20,20,20,41,6C,74,3A, [3627]
20,20,20,20,20,20,20,BAFE
3360 DATA 9E80,20,20,20,20,20,20,20,20,20, [2728]
20,20,20,20,20,20,20,FFFF
3370 DATA 9E90,20,20,20,20,20,20,20,95,95,20, [2092]
20,20,20,20,4E,65,75,2199
3380 DATA 9EA0,3A,20,20,20,20,20,20,20,20, [2716]
20,20,20,20,20,20,20,FFF2
3390 DATA 9EB0,20,20,20,20,20,20,20,20,20, [2574]
20,20,20,20,20,95,93,FE26
3400 DATA 9EC0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [3838]
9A,9A,9A,9A,9A,9A,9A,0000
3410 DATA 9ED0,9A,9A,9A,9A,9A,9A,9A,9A,9A, [2627]
9A,9A,9A,9A,9A,9A,9A,0000
3420 DATA 9EE0,9A,9A,9A,9A,9A,9A,99,00,00, [4084]
00,00,00,00,00,00,00,EA76
3430 DATA *ENDE*                              [502]
3440 adr=&9000:zeile=14:MEMORY adr-1          [2805]
3450 READ d$:IF d$="*ENDE*"THEN 3560          [1208]
3460 pr=0                                     [117]
3470 FOR i=1 TO 16                            [317]
3480 READ a$:a=VAL("&"+a$)                    [883]
3490 POKE adr,a:adr=adr+1                     [717]
3500 pr=pr*2:IF pr>65535 THEN pr=pr-65535     [2485]
3510 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+65    [1890]
536
3520 NEXT i                                   [375]
3530 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 TH    [2610]
EN pr2=pr2+65536
3540 IF pr<>pr2 THEN PRINT"Pruefsummenfehl    [3615]
er in Zeile";zeile:STOP
3550 zeile=zeile+1:GOTO 3450                  [847]
3560 SAVE"TRON.BIN",B,&9000,&EE6              [811]
3570 PRINT d$:END                             [346]
```

Listing "TRON 3D"

MÜLLER –

DAS INTELLIGENTE SPIEL!!!

Bei diesem Spiel geht es nicht etwa darum, Mehl zu mahlen, wie man vielleicht meinen könnte. Vielmehr handelt es sich hierbei um ein Strategiespiel für Denker.



Müller - eine Kombination aus Mühle und Drei gewinnt. Einfach gut…

Das Spiel erklärt sich selbst, da die Anleitung im Hauptmenü des Programms abgerufen werden kann, deshalb hier nur eine kurze Beschreibung: Auf einem Mühle-Spielfeld muß man nach den Regeln von 'Drei gewinnt' Spielsteine setzen. Gespielt wird gegen den Computer. Dieser hat drei verschiedene Spielstärken. Der Computer setzt seine Steine durch eine programmierte 'Künstliche Intelligenz'. Es wäre leicht möglich gewesen, den Computer so intelligent zu programmieren, daß man nicht mehr gewinnen kann. Dadurch würde das Spiel aber sehr schnell langweilig werden. Deshalb hat der Computer eine gewisse Fehlerquote bei seinen Zügen, die von seiner Spielstärke abhängt.

Das Programm ist mit RENUM in Zehnerschritten neu durchnumeriert worden, wodurch es leicht mit Hilfe der automatischen Zeilennumerierung (nach Eingabe des AUTO-Befehls) eingetippt werden kann. Für Abtippfaule gibt es folgende Möglichkeit, die Abtipparbeit zu verkürzen: Das Programm braucht nur bis einschließlich der Zeile 3600 abgetippt werden, danach müssen die Zeilen 1940 und 2070 gelöscht werden. Das Programm ist nun 1,5 KB kürzer, es fehlen jedoch zwei Menüpunkte (die Anleitung und die Eingabe von Problemstellungen). Mit der Menüfunktion 'Problemstellung' kann man Spielstellungen eingeben und von dieser aus dann weiterspielen. Das Spiel läßt sich durchweg mit dem Joystick steuern!

(Jürgen Altfeld/sr)

für 464-664-6128

```
10 REM     Juergen Altfeld 1988                  [1319]
20 RANDOMIZE TIME                                [1777]
30 DEFINT a-e,g-z                                [739]
40 GOSUB 2650                                    [817]
50 GOSUB 2780                                    [831]
60 GOSUB 2090                                    [865]
70 GOSUB 2160                                    [875]
80 GOTO 1710                                     [357]
90 score1=0:score2=0                             [1478]
100 game=1                                       [326]
110 GOSUB 2760                                   [899]
120 GOSUB 2930                                   [849]
130 IF game>1 THEN GOSUB 3180                    [825]
140 IF (game-beginn)/2=INT((game-beginn)/2 [3510]
) THEN 210
150 a=INT(RND*4):IF a=0 THEN a=4                 [2053]
160 IF a=1 THEN a=10                             [1058]
170 IF a=2 THEN a=13                             [394]
180 IF a=3 THEN a=19                             [831]
190 GOSUB 1660                                   [879]
200 GOSUB 3550:IF a THEN 1490                    [1397]
210 GOSUB 3220                                   [949]
220 CALL 42525                                   [403]
230 b=PEEK(42524)                                [1315]
240 IF b=254 THEN 1100                           [1137]
250 IF b=255 THEN 1380                           [1422]
260 GOSUB 3550:IF a THEN 1490                    [1397]
270 IF (b AND 15)=0 THEN 320                     [1140]
280 a=0                                          [324]
290 IF PEEK(feld(a))=0 THEN POKE feld(a),1 [5322]
:CALL 42525:IF PEEK(42524)=255 THEN GOSUB
1660:GOTO 1380 ELSE POKE feld(a),0
300 a=a+1                                        [725]
310 GOTO 290                                     [411]
320 IF (b AND 240)/16=0 THEN 370                 [1745]
330 a=0                                          [324]
340 IF PEEK(feld(a))=0 THEN POKE feld(a),1 [6872]
:CALL 42525:IF ((PEEK(42524)AND 240)/16)<(
b AND 240)/16 THEN GOSUB 1660:GOTO 200 ELS
E POKE feld(a),0
350 a=a+1                                        [725]
360 GOTO 340                                     [464]
370 IF level>1 THEN 480                          [702]
380 IF level=3 THEN 910                          [1022]
```

Listing "MÜLLER"

```
390 b=INT(RND*2):a=set(nr,b)                     [1115]
400 IF PEEK(feld(a))>0 THEN a=set(nr,b XOR [2297]
 1)
410 IF PEEK(feld(a))>0 THEN 440                  [1885]
420 GOSUB 1660                                   [879]
430 GOTO 200                                     [429]
440 a=INT(RND*24)                                [1445]
450 IF PEEK(feld(a))>0 THEN 440                  [1885]
460 GOSUB 1660                                   [879]
470 GOTO 200                                     [429]
480 a=0                                          [324]
490 IF PEEK(feld(a))=0 THEN POKE feld(a),1 [5918]
:CALL 42525:IF (PEEK(42524) AND 15)>1 THEN
 GOSUB 1660:GOTO 200 ELSE POKE feld(a),0
500 a=a+1                                        [725]
510 IF a<24 THEN 490                             [1053]
520 IF level>1 THEN 580                          [698]
530 a=0                                          [324]
540 IF PEEK(feld(a))=0 THEN POKE feld(a),1 [5558]
6:CALL 42525:IF ((PEEK(42524)AND 240)/16)>
1 THEN GOSUB 1660:GOTO 200 ELSE POKE feld(
a),0
550 a=a+1                                        [725]
560 IF a<24 THEN 540                             [1059]
570 GOTO 380                                     [488]
580 a=0                                          [324]
590 IF PEEK(feld(a))=0 THEN POKE feld(a),1 [5804]
:CALL 42525:IF (PEEK(42524) AND 15)=1 THEN
 630 ELSE POKE feld(a),0
600 a=a+1                                        [725]
610 IF a<24 THEN 590                             [1057]
620 IF level=3 THEN 380 ELSE 530                 [1345]
630 b=0                                          [312]
640 IF PEEK(feld(b))=0 THEN POKE feld(b),1 [5990]
6:CALL 42525:IF PEEK(42524)>0 THEN POKE fe
ld(b),0 ELSE 670
650 b=b+1                                        [569]
660 IF b<24 THEN 640 ELSE POKE feld(a),0:G [2264]
OTO 600
670 c=0                                          [348]
680 IF PEEK(feld(c))=0 THEN POKE feld(c),1 [9756]
:CALL 42525:IF (PEEK(42524) AND 15)<2 THEN
 POKE feld(c),0 ELSE POKE feld(c),0:POKE f
eld(b),0:GOSUB 1660:GOTO 200
690 c=c+1                                        [701]
700 IF c<24 THEN GOTO 680                        [1255]
710 POKE feld(b),0:POKE feld(a),0:GOTO 600 [2182]
720 b=0                                          [312]
730 a(b)=0                                       [390]
```

Listing "MÜLLER"


```
740 IF PEEK(feld(a(b)))=0 THEN POKE feld(a [2852]
(b)),16 ELSE 790
750 CALL 42525:c=(PEEK(42524)AND 240)/16   [1049]
760 IF c=1 AND b<8 THEN b=b+1:GOTO 830     [2281]
770 IF c>1 THEN FOR a=0 TO b:POKE feld(a(a [2957]
)),0:NEXT:c=1:RETURN
780 POKE feld(a(b)),0                      [1014]
790 a(b)=a(b)+1                            [463]
800 IF a(b)<24 THEN 740                    [542]
810 IF b=0 THEN c=0:RETURN                 [1970]
820 b=b-1:POKE feld(a(b)),0:GOTO 880       [2747]
830 a(b)=0                                 [390]
840 IF PEEK(feld(a(b)))=0 THEN POKE feld(a [2908]
(b)),1 ELSE 880
850 CALL 42525:IF PEEK(42524)>0 THEN 870   [1152]
860 b=b+1:IF b<9 THEN 730 ELSE b=b-1       [2045]
870 POKE feld(a(b)),0                      [1014]
880 a(b)=a(b)+1                            [463]
890 IF a(b)<24 THEN 840                    [459]
900 b=b-1:POKE feld(a(b)),0:GOTO 790       [2749]
910 GOSUB 720                              [933]
920 IF c=0 THEN 390                        [1125]
930 FOR a=0 TO 8:b(a)=a(a)                 [1000]
940 IF a>b THEN b(a)=24                    [1250]
950 NEXT                                   [350]
960 e=b                                    [514]
970 d=INT(RND*(e+1)):IF b(d)=24 THEN 970   [2025]
980 POKE feld(b(d)),1                      [825]
990 GOSUB 720                              [933]
1000 POKE feld(b(d)),0                     [826]
1010 IF c=0 THEN a=b(d):GOTO 460           [1897]
1020 b(d)=24                               [774]
1030 a=0                                   [324]
1040 IF b(a)<24 THEN 970                   [667]
1050 a=a+1:IF a<=e THEN 1040               [1332]
1060 d=set(nr,0):IF PEEK(feld(d))=0 THEN P [5543]
OKE feld(d),1:GOSUB 720:IF c=0 THEN a=d:GO
TO 460 ELSE POKE feld(d),0
1070 d=set(nr,1):IF PEEK(feld(d))=0 THEN P [5557]
OKE feld(d),1:GOSUB 720:IF c=0 THEN a=d:GO
TO 460 ELSE POKE feld(d),0
1080 GOTO 440                              [444]
1090 END                                   [110]
1100 PEN 2                                 [548]
1110 LOCATE 3,2:PRINT"Spieler hat gewonnen [2693]
"
1120 PEN 3                                 [547]
1130 LOCATE 3,25:PRINT"Computer hat verlor [4039]
en"
1140 score1=score1+2                       [1646]
1150 GOSUB 3180                            [1148]
1160 game=game+1                           [644]
1170 LOCATE #1,2,25:PEN #1,2:PRINT#1,CHR$( [4211]
24)"FIRE druecken"CHR$(24);
1180 GOSUB 2360                            [819]
1190 IF game>spiele THEN 1210              [971]
1200 GOTO 110                              [423]
1210 WINDOW #1,1,25,1,25:PAPER#1,0:CLS#1   [2118]
1220 PEN 1:LOCATE 8,2:PRINT"Gesamtsieger:" [1996]
1230 LOCATE 8,3:PRINT STRING$(13,"=")      [1817]
1240 PEN 2:PRINT:PRINT                     [849]
1250 PRINT:PRINT"Spieler hat von"spiele"Sp [2453]
ielen"
1260 PRINT score1/2"gewonnen ("USING"###"; [5560]
(score1*100/(score1+score2));:PRINT"%)"
1270 PEN 3                                 [547]
1280 PRINT:PRINT:PRINT                     [1082]
1290 PRINT"Computer hat"score2/2"Spiele"   [3249]
1300 PRINT"gewonnen ("USING"###";(score2*1 [3936]
00/(score1+score2));:PRINT"%)"
1310 PRINT:PRINT:PRINT:PRINT:PRINT         [2848]
1320 PEN 1:PRINT"Gesamtsieger ist somit:"  [2361]
1330 IF score1=score2 THEN PRINT"KEINER (P [3832]
ATT)":GOTO 1360
1340 IF score1>score2 THEN PEN 2:PRINT"DER [4789]
 SPIELER":GOTO 1360
1350 PEN 3:PRINT"DER COMPUTER"             [2057]
1360 GOSUB 2360                            [819]
1370 GOTO 70                               [396]
1380 PEN 2                                 [548]
1390 LOCATE 3,2:PRINT"Spieler hat verloren [2813]
"
1400 PEN 3                                 [547]
1410 LOCATE 3,25:PRINT"Computer hat gewonn [3143]
en"
1420 score2=score2+2                       [1900]
1430 GOSUB 3180                            [1148]
```

Listing "MÜLLER"

```
1440 game=game+1                           [644]
1450 LOCATE #1,2,25:PEN #1,2:PRINT#1,CHR$( [4211]
24)"FIRE druecken"CHR$(24);
1460 GOSUB 2360                            [819]
1470 IF game>spiele THEN 1210              [971]
1480 GOTO 110                              [423]
1490 CALL 42525                            [403]
1500 a=PEEK(42524)                         [1210]
1510 IF a=254 THEN 1100                    [1164]
1520 IF a=255 THEN 1100                    [986]
1530 PEN 2                                 [548]
1540 LOCATE 3,2:PRINT"Spieler ";           [1700]
1550 PEN 1:PRINT"und ";:PEN 3:PRINT"Comput [1571]
er"
1560 PEN 1:LOCATE 3,25                     [1024]
1570 PRINT"haben gewonnen (Patt)"          [3334]
1580 score1=score1+1                       [1643]
1590 score2=score2+1                       [1901]
1600 GOSUB 3180                            [1148]
1610 game=game+1                           [644]
1620 LOCATE #1,2,25:PEN #1,2:PRINT#1,CHR$( [4211]
24)"FIRE druecken"CHR$(24);
1630 GOSUB 2360                            [819]
1640 IF game>spiele THEN 1210              [971]
1650 GOTO 110                              [423]
1660 FOR b=0 TO 20                         [1043]
1670 LOCATE x(a),y(a):PEN 2:PRINT stein$:L [4318]
OCATE x(a),y(a):PEN 3:PRINT stein$
1680 NEXT                                  [350]
1690 POKE feld(a),1                        [1026]
1700 RETURN                                [555]
1710 LOCATE 10,9:PEN 3:PRINT CHR$(243)     [1460]
1720 GOSUB 2360                            [819]
1730 IF a=1 THEN PEN 1:LOCATE 10,9:PRINT"- [3290]
":GOTO 2030
1740 IF a=2 THEN PEN 1:LOCATE 10,9:PRINT"- [2620]
":GOTO 1760
1750 IF a=16 THEN GOTO 90 ELSE 1720        [1502]
1760 LOCATE 10,11:PEN 3:PRINT CHR$(243)    [1456]
1770 GOSUB 2360                            [819]
1780 IF a=1 THEN PEN 1:LOCATE 10,11:PRINT" [2861]
-":GOTO 1710
1790 IF a=2 THEN PEN 1:LOCATE 10,11:PRINT" [3356]
-":GOTO 1830
1800 IF a=16 THEN PEN 3:LOCATE 13,12+begin [5250]
n:PRINT" ":IF beginn=0 THEN beginn=1 ELSE
beginn=0
1810 LOCATE 13,12+beginn:PRINT"*"          [1750]
1820 GOTO 1770                             [361]
1830 LOCATE 10,15:PEN 3:PRINT CHR$(243)    [1529]
1840 GOSUB 2360                            [819]
1850 IF a=1 THEN PEN 1:LOCATE 10,15:PRINT" [2122]
-":GOTO 1760
1860 IF a=2 THEN PEN 1:LOCATE 10,15:PRINT" [3988]
-":GOTO 1900
1870 IF a=16 THEN spiele=spiele+1:IF spiel [3493]
e=10 THEN spiele=1
1880 LOCATE 30,15:PEN 1:PRINT CHR$(24);spi [1821]
ele;CHR$(24)
1890 GOTO 1840                             [395]
1900 LOCATE 10,17:PEN 3:PRINT CHR$(243)    [2696]
1910 GOSUB 2360                            [819]
1920 IF a=1 THEN PEN 1:LOCATE 10,17:PRINT" [3247]
-":GOTO 1830
1930 IF a=2 THEN PEN 1:LOCATE 10,17:PRINT" [3297]
-":GOTO 1960
1940 IF a=16 THEN GOTO 3770                [1393]
1950 GOTO 1910                             [381]
1960 LOCATE 10,19:PEN 3:PRINT CHR$(243)    [1642]
1970 GOSUB 2360                            [819]
1980 IF a=1 THEN PEN 1:LOCATE 10,19:PRINT" [3340]
-":GOTO 1900
1990 IF a=2 THEN PEN 1:LOCATE 10,19:PRINT" [3386]
-":GOTO 2030
2000 IF a=16 THEN LOCATE 13,19+level:PRINT [5701]
" ":level=level+1:IF level=4 THEN level=1
2010 LOCATE 13,19+level:PEN 3:PRINT"*"     [1223]
2020 GOTO 1970                             [337]
2030 LOCATE 10,24:PEN 3:PRINT CHR$(243)    [2022]
2040 GOSUB 2360                            [819]
2050 IF a=1 THEN PEN 1:LOCATE 10,24:PRINT" [4461]
-":GOTO 1960
2060 IF a=2 THEN PEN 1:LOCATE 10,24:PRINT" [2895]
-":GOTO 1710
2070 IF a=16 THEN 3610                     [1527]
2080 GOTO 2040                             [355]
2090 spiele=1                              [837]
```

Listing "MÜLLER"

```
2100 stein$=CHR$(214)+CHR$(215)+CHR$(10)+C [3460]
HR$(8)+CHR$(8)+CHR$(213)+CHR$(212)
2110 zeiger1$=CHR$(214)+CHR$(9)+CHR$(9)+CH [8262]
R$(215)+CHR$(8)+CHR$(8)+CHR$(8)+CHR$(8)+CH
R$(10)+CHR$(10)+CHR$(10)+CHR$(213)+CHR$(9)
+CHR$(9)+CHR$(212)
2120 zeiger2$=" "+CHR$(9)+CHR$(9)+" "+CHR$ [5125]
(8)+CHR$(8)+CHR$(8)+CHR$(8)+CHR$(10)+CHR$(
10)+CHR$(10)+" "+CHR$(9)+CHR$(9)+" "
2130 level=1                                [567]
2140 beginn=0                               [386]
2150 RETURN                                 [555]
2160 MODE 1                                 [506]
2170 INK 0,0:INK 1,13:INK 2,6:INK 3,11      [999]
2180 PAPER 0:BORDER 0:CLS                   [1596]
2190 LOCATE 17,1:PEN 2:PRINT"MUELLER"       [1719]
2200 LOCATE 17,2:PEN 3:PRINT STRING$(7,"=" [2605]
):PEN 1
2210 LOCATE 1,5:PRINT"(C) and Design by "; [3594]
:PEN 3:PRINT"Toasti"
2220 LOCATE 10,9:PEN 1:PRINT"- Spielbeginn [3683]
"
2230 LOCATE 10,11:PRINT"- Erster Zug durch [2483]
"
2240 LOCATE 15,12:PRINT"Blau (Computer)"    [2016]
2250 LOCATE 15,13:PRINT"Rot (Spieler)"      [1857]
2260 LOCATE 10,15:PRINT"- Anzahl der Spiel [3793]
e:";CHR$(24);spiele;CHR$(24)
2270 LOCATE 10,17:PRINT"- Anleitung"        [1921]
2280 LOCATE 10,19:PRINT"- Spielstaerke des [5168]
 Computers:"
2290 LOCATE 15,20:PRINT"Anfaenger"          [1947]
2300 LOCATE 15,21:PRINT"Fortgeschrittener" [3658]
2310 LOCATE 15,22:PRINT"Profi"              [1673]
2320 LOCATE 10,24:PRINT"- Stellung eingebe [2770]
n"
2330 LOCATE 13,12+beginn:PEN 3:PRINT"*"     [2288]
2340 LOCATE 13,19+level:PRINT"*"            [1440]
2350 RETURN                                 [555]
2360 a=JOY(0):IF a<>0 THEN 2360             [2725]
2370 a=JOY(0):a$=INKEY$                     [1952]
2380 IF a$<>"" AND c=-1 THEN 2410           [1294]
2390 IF a=0 THEN 2370                       [523]
2400 a$=INKEY$:IF a$<>"" THEN 2400          [1937]
2410 RETURN                                 [555]
2420 DATA 00,21,04,A6,06,08,11,00,  234     [1206]
2430 DATA 00,7E,23,86,23,86,23,CD,  704     [1716]
2440 DATA AE,A6,10,F5,3A,04,A6,21,  862     [1599]
2450 DATA 0D,A6,86,21,19,A6,86,CD,  876     [1858]
2460 DATA AE,A6,3A,07,A6,21,0E,A6,  784     [1268]
2470 DATA 86,21,16,A6,86,CD,AE,A6,  1034    [1140]
2480 DATA 3A,0A,A6,21,0F,A6,86,21,  615     [1634]
2490 DATA 13,A6,86,CD,AE,A6,3A,05,  927     [1607]
2500 DATA A6,21,08,A6,86,21,0B,A6,  717     [1960]
2510 DATA 86,CD,AE,A6,3A,14,A6,21,  956     [1805]
2520 DATA 17,A6,86,21,1A,A6,86,CD,  887     [1110]
2530 DATA AE,A6,3A,0C,A6,21,1O,A6,  791     [1094]
2540 DATA 86,21,15,A6,86,CD,AE,A6,  1033    [1080]
2550 DATA 3A,09,A6,21,11,A6,86,21,  616     [1239]
2560 DATA 18,A6,86,CD,AE,A6,3A,06,  933     [976]
2570 DATA A6,21,12,A6,86,21,1B,A6,  743     [1538]
2580 DATA 86,CD,AE,A6,7B,CB,27,CB,  1247    [2013]
2590 DATA 27,CB,27,CB,27,B2,32,1C,  779     [1973]
2600 DATA A6,C9,FE,02,20,02,14,C9,  878     [1028]
2610 DATA FE,20,20,02,1C,C9,FE,03,  806     [1426]
2620 DATA 20,03,E1,18,06,FE,30,CO,  784     [1509]
2630 DATA E1,18,06,3E,FF,32,1C,A6,  816     [1339]
2640 DATA C9,3E,FE,32,1C,A6,C9,00,  962     [1498]
2650 RESTORE 2420:MEMORY 42499:dat=0:sz=0: [2195]
dz=2420
2660 FOR f=42524 TO 42707                   [1249]
2670 READ byte$:dat=dat+1                   [1348]
2680 byte=VAL(CHR$(38)+byte$):sz=sz+byte    [1381]
2690 POKE f,byte                            [116]
2700 IF dat<8 AND f<42707 THEN 2740         [1252]
2710 READ chksum                            [1222]
2720 IF chksum<>sz THEN PRINT "Fehler in Z [3056]
eile :";dz
2730 dz=dz+10:sz=0:dat=0                    [843]
2740 NEXT                                   [350]
2750 RETURN                                 [555]
2760 FOR a=0 TO 24:POKE 42500+a,0:NEXT      [2080]
2770 RETURN                                 [555]
2780 RESTORE 2820:DIM x(23),y(23),feld(23) [3181]
,set(23,1),a(8),b(8)
2790 FOR a=0 TO 23:READ x(a),y(a):feld(a)= [4537]
42500+a:NEXT
2800 FOR a=0 TO 23:READ set(a,0),set(a,1): [3086]
NEXT
```

Listing "MÜLLER"

```
2810 RETURN                                 [555]
2820 DATA 3,4, 12,4, 21,4                   [693]
2830 DATA 6,7, 12,7, 18,7                   [748]
2840 DATA 9,10, 12,10, 15,10                [856]
2850 DATA 3,13, 6,13, 9,13, 15,13, 18,13,  [2225]
21,13
2860 DATA 9,16, 12,16, 15,16                [238]
2870 DATA 6,19, 12,19, 18,19                [1137]
2880 DATA 3,22, 12,22, 21,22                [335]
2890 DATA 1,9, 0,2, 1,14, 4,10, 3,5, 4,13  [1430]
2900 DATA 7,11, 6,8, 7,12, 0,21, 3,18, 6,1 [1627]
5, 8,17
2910 DATA 5,20, 2,23, 11,16, 15,17, 12,16, [1807]
 10,19
2920 DATA 18,20, 13,19, 9,22, 21,23, 14,22 [2089]
2930 MODE 1:PAPER 0:CLS                     [1469]
2940 WINDOW #1,26,40,1,25:PAPER #1,1:CLS # [3265]
1
2950 PEN #1,0                               [466]
2960 LOCATE #1,5,2:PRINT#1,"MUELLER"        [2404]
2970 LOCATE #1,5,3:PRINT#1,STRING$(7,"=")   [1114]
2980 LOCATE #1,2,5:PRINT#1,"Spiel Nr. 1":P [3838]
RINT#1," (von "spiele")"
2990 LOCATE #1,2,10:PRINT#1,"Score ";:PEN# [4640]
1,2:PRINT#1,"Spieler"
3000 LOCATE #1,2,12:PRINT#1,"0.0    (  0%) [2819]
":PEN#1,0
3010 LOCATE #1,2,15:PRINT#1,"Score ";:PEN# [3309]
1,3:PRINT#1,"Computer"
3020 LOCATE #1,2,17:PRINT#1,"0.0    (  0%) [2638]
":PEN #1,0
3030 LOCATE #1,2,21:PEN #1,3:PRINT#1,"Spie [3135]
lstaerke:"
3040 PRINT#1:PEN #1,0:IF level=1 THEN PRIN [3826]
T#1," Anfaenger"
3050 IF level=2 THEN PRINT#1," Fortgeschri [2618]
tt."
3060 IF level=3 THEN PRINT#1," Profi"       [2117]
3070 ORIGIN 46,46:MOVE 0,0:DRAWR 290,0,1    [2044]
3080 DRAWR 0,290:DRAWR -290,0:DRAW 0,0      [1405]
3090 MOVE 2,2:DRAWR 286,0:DRAWR 0,286:DRAW [3739]
R -286,0:DRAW 2,2
3100 MOVE 144,0:DRAWR 0,290:MOVE 146,0:DRA [4370]
WR 0,290
3110 MOVE 0,144:DRAWR 290,0:MOVE 0,146:DRA [2276]
WR 290,0
3120 MOVE 48,48:DRAWR 194,0:DRAWR 0,194:DR [3404]
AWR -194,0:DRAWR 0,-194
3130 MOVE 50,50:DRAWR 190,0:DRAWR 0,190:DR [2677]
AWR -190,0:DRAWR 0,-190
3140 MOVE 96,96:DRAWR 98,0:DRAWR 0,98:DRAW [2962]
R -98,0:DRAWR 0,-98
3150 WINDOW #2,10,15,11,16:CLS #2           [1922]
3160 MOVE 98,98:DRAWR 94,0:DRAWR 0,94:DRAW [2853]
R -94,0:DRAWR 0,-94
3170 RETURN                                 [555]
3180 PEN #1,0:LOCATE #1,11,5:PRINT#1,game   [1598]
3190 PEN #1,2:LOCATE #1,2,12:PRINT#1,USING [5833]
"##";score1;:PRINT#1,"     (";USING"###";(
score1/(score1+score2))*100
3200 PEN #1,3:LOCATE #1,2,17:PRINT#1,USING [6693]
"##";score2;:PRINT#1,"     (";USING"###";(
score2/(score1+score2))*100
3210 RETURN                                 [555]
3220 nr=0                                   [324]
3230 LOCATE x(nr)-1,y(nr)-1:PEN 2:PRINT ze [3474]
iger1$
3240 GOSUB 2360                             [819]
3250 LOCATE x(nr)-1,y(nr)-1:PRINT zeiger2$ [2561]
3260 a$=LOWER$(a$):IF (a$="s" OR a$="c") A [2457]
ND c=-1 THEN RETURN
3270 IF a=1 THEN GOTO 3330                  [1653]
3280 IF a=2 THEN GOTO 3370                  [705]
3290 IF a=4 THEN GOTO 3410                  [707]
3300 IF a=8 THEN GOTO 3450                  [1149]
3310 IF a=16 THEN GOTO 3490                 [1409]
3320 GOTO 3230                              [443]
3330 a=23                                   [533]
3340 IF x(a)=x(nr) AND y(a)<y(nr) THEN nr= [3187]
a:GOTO 3230
3350 a=a-1                                  [721]
3360 IF a<0 THEN GOTO 3230 ELSE 3340        [1543]
3370 a=0                                    [324]
3380 IF x(a)=x(nr) AND y(a)>y(nr) THEN nr= [2361]
a:GOTO 3230
3390 a=a+1                                  [725]
3400 IF a>23 THEN GOTO 3230 ELSE 3380       [1731]
3410 a=23                                   [533]
```

Listing "MÜLLER"


```
3420 IF x(a)<x(nr) AND y(a)=y(nr) THEN nr= [2267]
a:GOTO 3230
3430 a=a-1                                  [721]
3440 IF a<0 THEN GOTO 3230 ELSE 3420        [1511]
3450 a=0                                    [324]
3460 IF x(a)>x(nr) AND y(a)=y(nr) THEN nr= [3332]
a:GOTO 3230
3470 a=a+1                                  [725]
3480 IF a>23 THEN GOTO 3230 ELSE 3460       [1954]
3490 IF PEEK(feld(nr))<>0 THEN IF c=-1 THE [3715]
N GOSUB 3750:GOTO 3230:ELSE PRINT CHR$(7):
GOTO 3230
3500 POKE feld(nr),16                       [1023]
3510 LOCATE x(nr),y(nr):PEN 2               [1448]
3520 PRINT stein$                           [456]
3530 IF c=-1 THEN GOTO 3230                 [1608]
3540 RETURN                                 [555]
3550 a=0                                    [324]
3560 IF PEEK(feld(a))=0 THEN a=0:GOTO 3600 [1415]
3570 a=a+1                                  [725]
3580 IF a<24 THEN 3560                      [1174]
3590 a=1                                    [327]
3600 RETURN                                 [555]
3610 GOSUB 2760                             [899]
3620 GOSUB 2930                             [849]
3630 LOCATE 1,1:PRINT"Farbwechsel: Stein a [2803]
uf"
3640 PRINT"Stein setzen"                    [2544]
3650 LOCATE 1,25:PRINT"C/S - Start Comp./S [2016]
pieler";
3660 c=-1                                   [533]
3670 GOSUB 3220                             [949]
3680 LOCATE 1,1:PRINT STRING$(25," "):PRIN [1794]
T STRING$(25," ")
3690 LOCATE 1,25:PRINT STRING$(25," ");     [1678]
3700 c=0:game=1:score1=0:score2=0           [2152]
3710 IF a$="c" THEN 220                     [762]
3720 CALL 42525:b=PEEK(42524)               [983]
3730 IF b=255 THEN 1380                     [1422]
3740 IF b=254 THEN 1100 ELSE 200            [1942]
3750 IF PEEK(feld(nr))=1 THEN POKE feld(nr [7262]
),16:LOCATE x(nr),y(nr):PEN 2:PRINT stein$
ELSE POKE feld(nr),1:LOCATE x(nr),y(nr):PE
N 3:PRINT stein$
3760 RETURN                                 [555]
3770 MODE 1:PAPER 0:PEN 2:CLS               [1750]
3780 PRINT TAB(17);"MUELLER"                [608]
3790 PRINT TAB(17);"=======                 [1248]
3800 PRINT:PRINT                            [743]
3810 PEN 1                                  [549]
3820 PRINT"MUELLER ist ein Spiel, das      [2838]
 mit dem";
3830 PRINT"Brettspiel MUEHLE verwandt i    [4034]
st (daher";
3840 PRINT"der Name). Der Spieler bzw. der [4624]
 Computer";
3850 PRINT"setzen abwechselnd einen Spiel [5190]
stein auf";
3860 PRINT"das Spielfeld. Gespielt wird [3655]
 solange,";
3870 PRINT"bis das Spielfeld voll ist      [5014]
 oder der";
3880 PRINT"Spieler (bzw. Computer) gewonne [3504]
n hat."
3890 PRINT"Sieger ist, wer drei eigene Sp [2347]
ielsteine";
3900 PRINT"in eine durch eine Linie verbun [5590]
dene Rei-";
3910 PRINT"he gesetzt hat."                 [1847]
3920 PRINT"Es koennen auch bis zu 9 Spiele [4088]
 nachein-";
3930 PRINT"ander gespielt werden. Fuer jed [4528]
es gewon-";
3940 PRINT"nene Spiel gibt es 2 Punkte fue [4544]
r den Ge-";
3950 PRINT"winner. Bei einem Unentschiede [5169]
n (Spiel-";
3960 PRINT"feld voll) bekommt jeder Spie [3601]
ler einen";
3970 PRINT"Punkt."                          [1096]
3980 PRINT"Gesamtsieger ist, wer die hoehe [5280]
re Punkt-";
3990 PRINT"zahl erreicht hat."              [1955]
4000 LOCATE 11,25:PEN 3:PAPER 2             [565]
4010 PRINT"Bitte FIRE druecken"             [2383]
4020 GOSUB 2360                             [819]
4030 GOTO 70                                [396]
```

Listing "MÜLLER"

Aus dem Sybex-Angebot

Einführung in WordStar
Der Bestseller zum populärsten Textverarbeitungsprogramm wurde für die Besitzer des CPC überarbeitet. Und damit wichtige Hilfe und Nachschlagewerk bei der Arbeit mit WordStar und MailMerge auf dem CPC. Neben der klaren Einführung in den effektiven Umgang mit WordStar gibt es u.a. auch wertvolle Hinweise für die Installation von Druckern und zu Systempatches.
280 Seiten/40 Abb. Best.-Nr. 421 DM 42,–*

Arbeiten mit dBasell
dBasell ist im PC-Bereich wohl eines der leistungsstärksten Datenbankprogramme. Benutzern eines Schneider CPC vermittelt ein echter Experte in diesem Buch alle Kenntnisse, die für den erfolgreichen Einsatz von dBasell wichtig sind. Z.B.: Installation von und Programmieren mit dBasell, Editieren von Dateien mit Wordstar, Tips und Tricks. Jeder Lernschritt wird durch praxisgerechte Beispiele ergänzt. Und zwar so, daß dem Leser die Umsetzung dann wirklich problemlos möglich ist. Ein Buch, das in jeder Arbeitsphase weiterhilft.
272 Seiten/m Abb. Best.-Nr. 422 DM 48,–*

CPC-Bücherkiste

Aus dem Data Becker-Angebot

CPC 6128/664 Intern
Blicken Sie hinter die Kulissen des CPC 664 und des CPC 6128. Kaum ein anderes Autorenteam hat sich so intensiv mit diesen Rechnern auseinandergesetzt: vom Prozessor bis hin zum speziellen Schnittstellenbaustein. Alles wird erklärt und dokumentiert. Natürlich auch das Betriebssystem mit all den wichtigen Facts und Hinweisen, die man braucht. Hier finden Sie die Information, die ein Profi erwarten kann.
456 Seiten Best.-Nr. 411 DM 69,–*

Das Floppybuch zum CPC
Was man alles aus der DDI-1 des CPC 464, CPC 664 und CPC 6128 holen kann, zeigt dieses Buch auf eindrucksvolle Weise. Neben den nötigen Erklärungen und einem ausführlichen DOS-Listing gibt es zahlreiche Utilities: eine komfortable Dateiverwaltung, einen Disk-Manager. Selbst CP/M-Grundlagen und die relative Dateiverwaltung werden erklärt. So findet wirklich jeder CPC-Besitzer in diesem Buch einen wertvollen Ratgeber.
422 Seiten Best.-Nr. 412 DM 49,–*

Das CP/M-Trainingsbuch zum CPC
Beherrschen Sie CP/M. Dieses Buch hilft Ihnen dabei. Von den ersten Schritten bis zum perfekten Umgang. Dabei werden natürlich die Versionen 2.2 und 3.0 für Schneider CPC 464, 664 und 6128 berücksichtigt. Dieses CP/M-Trainingsbuch bietet ein wenig mehr als andere: zum Beispiel Hilfsprogramme, mit denen Sie in der Lage sind, auch fremde Diskettenformate zu lesen oder Submit-Dateien zu erstellen.
260 Seiten Best.-Nr. 413 DM 49,–*

CPC Tips und Tricks Band 2
Der 2. Band aus der Tips und Tricks-Reihe ist für alle CPC-Besitzer interessant: Egal ob Sie nun einen 464, 664 oder 6128 besitzen. Schreiben Sie eigene Befehlserweiterungen oder einen Maskengenerator. Lernen Sie wichtige Systemroutinen kennen. Erfahren Sie, wie man Programme beschleunigt, und viele andere Dinge, die im täglichen Umgang mit dem Rechner fast unverzichtbar sind. Mit diesem Buch holt man noch mehr aus seinem CPC.
250 Seiten Best.-Nr. 414 DM 39,–*

Das Maschinensprachebuch zum CPC
Wer seinen CPC wirklich beherrschen will, der muß sich mit dem Thema Maschinensprache beschäftigen. Von den Grundlagen bis zur Programmierung des Z80-Prozessors. Das Maschinensprachebuch zum CPC hilft Ihnen von Anfang an. Mit einer genauen Beschreibung aller Befehle und ausführlichen Beispielen, mit Hinweisen zur Benutzung der Systemroutinen und einem Assembler/Disassembler sowie einem Monitor zum Abtippen. So macht der Einstieg Spaß.
330 Seiten Best.-Nr. 415 DM 39,–*

Das große Grafikbuch zum CPC
Dieses Buch ist für alle, die bisher dachten, spektakuläre Grafik auf dem CPC sei nicht möglich. Zwei Top-Autoren beweisen das Gegenteil: Mit CPC-Chart - dem Diagrammgenerator, mit Destroyed - dem Arcade-Game, mit CPCs World - dem 3-D-Animationsprogramm, mit Vektorgrafik, mit Sprites... Ja. Sie haben richtig gelesen : wir reden von den Grafikmöglichkeiten Ihres CPC - inklusive 6128 und Joyce.
589 Seiten Best.-Nr. 416 DM 49,–*

Programmwissen pur im Westentaschenformat

Führer zum CPC Best.-Nr. 451 DM 19,80 *
Führer zum CP/M Best.-Nr. 452 DM 19,80 *

Bücher für den CPC

464 BASIC-Trainingsbuch
BASIC auf dem CPC 464 ? Kein Problem ! Denn mit diesem Trainingsbuch lernen Sie nicht nur Schritt für Schritt die einzelnen Befehle und ihre Anwendungen kennen, sondern bekommen auch gleich einen sauberen Programmierstil mit allem, was dazu gehört: von der Problemanalyse über den Datenflußplan bis zum fertigen Programm. Dazu viele Übungsaufgaben mit Lösungen und zahlreichen Beispielen. Schlichtweg unentbehrlich!
285 Seiten Best.-Nr: 460 DM 39,–*

Das BASIC-Buch zum 6128
BASIC macht Spaß. Man muß es nur richtig erklärt bekommen. Und genau das tut das große BASIC-Buch zum CPC 6128. In diesem Buch steckt mehr als Einsteigerwissen: Variablen, Zahlensysteme, Bits und Bytes, Tokens, Stringbearbeitung, Sortierung, Laufschrift, selbstdefinierte Zeichen, Windows, Fehlerbehandlung, Kopierschutz, Grafiken, Soundprogrammierung, relative Dateien .... Das verstehen wir unter Vielfalt.
276 Seiten Best.-Nr. 461 DM 39,–*

464 Tips & Tricks
Hardwareaufbau des CPC 464, Betriebssystem und BASIC-Tokens, Bildschirmaufbau, Anwendungen der Window-Möglichkeiten, eine komplette Dateiverwaltung, Soundeditor - CPC Tips & Tricks bringt Ihnen diese und andere Ideen in Haus. Und damit das Ganze auch Spaß macht, kommen noch eine Reihe spannender Spiele dazu. Mit diesem Buch holt man mehr aus dem 464.
271 Seiten Best.-Nr: 462 DM 39,–*

464 Peeks & Pooks
Wer PEEKS und POKES zum CPC 464 kennen und anwenden will, der findet hier umfassende Information! Sie reicht vom Adreßbereich des Prozessors über Betriebssystem und Interpreter bis hin zur Einführung in die Maschinensprache. Dazu Programmierhilfen, Routinen sowie reichlich Material zu den Themen Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC und RAM-Pages!
180 Seiten Best.-Nr: 463 DM 29,–*

CPC Hardwareerweiterungen
Speziell für den Hobbyelektroniker, der mehr aus seinem CPC machen möchte! Von nützlichen Tips zur Platinenherstellung über Adreßdecodierung, Adapterkarten und Interfaces bis zur EPROM-Programmierboard und -Programmiernetzteil oder Motorsteuerung für Gleich- und Schrittschaltmotoren werden machbare Erweiterungen ausführlich und praxisnah beschrieben.
445 Seiten Best.-Nr: 464 DM 49,–*

464 Intern
Wirklich alle Geheimnisse des CPC 464 lüftet dieses Standardwerk: Neben dem kommentierten BASIC-ROM-Listing enthält es Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinenutzung, Character-Generator, u.v.m. Für den fortgeschrittenen Basic-Programmierer, für den Assembler-Programmierer ein absolutes Muß!
548 Seiten Best.-Nr: 465 DM 69,–*

Zu beziehen über:

DMV Verlag · Postfach 250 · 3440 Eschwege

* Unabhängig von der Anzahl der bestellten Bücher berechnen wir für das Inland 3,– DM bzw. für das Ausland 5,– DM Porto und Verpackung. Bitte benutzen Sie unseren Bestellcoupon auf Seite 157.

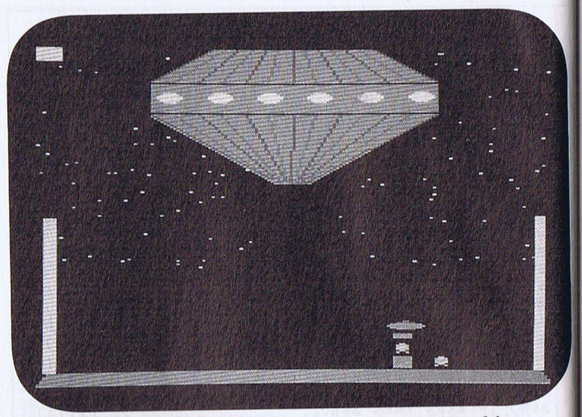
COSMIC

Angriff aus dem Weltall

Die Cylonen bereiten eine größere Offensive gegen die intergalaktische Weltmacht vor und haben zu diesem Zweck bereits mehrere Planeten erobert, auf denen sie sich mit Hilfe von Klystrongeneratoren versorgen. Sie, der Captain der COSMIC-STAR, haben den Auftrag erhalten, diese Generatoren zu stehlen.

Jeder dieser Planeten - es sind insgesamt vier - besitzt zwei dieser Generatoren, die Sie mit Hilfe eine BEAM-STRAHLS in ihr Raumschiff beamen sollen. Aber passen Sie auf, denn die Cylonen sehen nicht untätig zu, sondern versuchen Sie mit einem Laser zu treffen. Aber bevor Sie sich in die Nähe des Planeten begeben können, müssen Sie sich erst einmal durch einen Asteroidengürtel durchkämpfen, um danach ihr Mutterschiff verlassen zu können und die Generatoren zu stehlen. Aber wenn Sie denken, das Sie auf dem Planeten vor Asteroiden sicher sind, so haben Sie sich getäuscht, denn wenn ein Warnton ertönt, sollten Sie sich so schnell wie möglich in ihr Mutterschiff zurückbegeben, abwarten und rechtzeitig in die richtige Richtung schießen.

Nun zur Steuerung:
Im Asteroidengürtel: Den Joystick in die Richtung drücken, woher die Asteroiden kommen, also rechts, links, oben oder unten. Dabei sollten Sie aufpassen, daß Sie nicht in die falsche Richtung schießen, dies kostet eine Menge Energie.



Nach erfolgreichem Abschneiden gilt es, die Konzentration auf das Andockmanöver zu lenken.

Auf dem Planeten:
Raumschiff steuern, Joystick rechts, links, oben oder unten beamen - Joystick nach unten ziehen und Feuerknopf drücken (und gedrückt halten bis der Generator im Raumschiff verschwunden ist.) Dies geht aber nur, wenn Sie sich nicht zu weit vom und genau über dem Generator befinden.
Zurückkehren ins Mutterschiff - einfach in die Öffnung unterm Mutterschiff fliegen.
Asteroiden abschießen - Joystick rechts links (geht nur wenn man sich im Mutterschiff befindet)

Nachdem man den Asteroiden abgeschossen hat, geht es sofort beim nächsten Planeten weiter.
Das Spiel endet, wenn man:
a) dreimal von einem Asteroiden getroffen worden ist,
b) wenn die Energie aufgebraucht ist,
c) alle Planeten gerettet hat. Letzteres ist dabei das endgültige Ende des Spiels.
Möglichkeiten der Auffrischung der Energie: Für jeden aufgesammelten Generator gibt es Zusatzenergie. Viel Glück!

(Dirk Baumert/sr)

für 464-664-6128

```
1000 '******************                    [669]
1010 '*   C O S M I C   *                    [892]
1020 '*     S T A R     *                    [93]
1030 '*       by        *                    [227]
1040 '*  Dirk  Baumert  *                    [1172]
1050 '*       1988      *                    [336]
1060 '* programmed for  *                    [819]
1070 '(C)PC INTERNATIONAL                    [1080]
1080 '******************                    [669]
1090 '                                       [117]
1100 'Titelbild und Initalisierung           [2064]
1110 '                                       [117]
1120 IF PEEK(&A000)=0 THEN SYMBOL AFTER 25   [7106]
6:MEMORY &3FFF:LOAD"!cosmic.spr",&9000:LOA
D"!cosmic.mc"
1130 ENT 1,20,1,1:ENT 2,100,-5,1:ENT 3,5,1   [3129]
0,1:ENT 4,10,-1,1
1140 ENV 1,16,-1,2                           [468]
1150 GOSUB 3270                              [1170]
1160 OUT &BC00,2:OUT &BD00,49                [1291]
1170 OUT &BC00,1:OUT &BD00,46                [800]
1180 OUT &BC00,7:OUT &BD00,34                [1188]
1190 OUT &BC00,6:OUT &BD00,33                [913]
1200 FOR i=1 TO 300                          [318]
1210 PLOT INT(RND*640),INT(RND*400),INT(RN   [2306]
D*2)+1
1220 NEXT::INK 1,26:INK 2,13:PEN 1:SOUND 1   [3628]
,0,400,1,2,,1
1230 LOCATE 8,17:PRINT" Dirk Baumert";       [2612]
1240 FOR w=1 TO 2000:NEXT                    [1334]
1250 LOCATE 8,17:PRINT"  presents   ";       [2342]
1260 FOR w=1 TO 2000:NEXT                    [1334]
1270 LOCATE 8,17:PRINT" COSMIC-STAR ";:SOU   [2790]
ND 1,900,40,15,2,4,15
1280 WHILE INKEY$<>"":a$=a$+INKEY$:WEND      [3116]
1290 WHILE INKEY$="":WEND                    [1095]
1300 OUT &BC00,2:OUT &BD00,48                [1288]
1310 OUT &7F00,16:OUT &7F00,0                [996]
1320 OUT &BC00,2:OUT &BD00,49                [1291]
1330 OUT &7F00,16:OUT &7F00,0                [996]
1340 OUT &BC00,2:OUT &BD00,50                [1270]
1350 OUT &BC00,2:OUT &BD00,49                [1291]
1360 WEND                                    [390]
1370 GOSUB 3270                              [1170]
1380 OUT &BC00,2:OUT &BD00,46:OUT &BC00,1:   [3099]
OUT &BD00,40
1390 OUT &BC00,7:OUT &BD00,30:OUT &BC00,6:   [4172]
OUT &BD00,25:OUT &BC00,4:OUT &BD00,38
1400 FOR a=1 TO 200:PLOT INT(RND*640),INT(   [3419]
RND*400),INT(RND*2)+1:NEXT
1410 x=0:FOR a=100 TO 180:MOVE 300-x,a:DRA   [3301]
W 340+x,a,3:x=x+2:NEXT
1420 FOR a=180 TO 220:MOVE 140,a:DRAW 500,   [1435]
a:NEXT
1430 x=0:FOR a=220 TO 300:MOVE 140+x,a:DRA   [3601]
W 500-x,a:x=x+2:NEXT
1440 x=300:pe=14:FOR a=140 TO 500 STEP 30:   [8445]
MOVE x,300:DRAW a,220,pe:MOVE x,100:DRAW a
,180:x=x+3:IF pe=14 THEN pe=15 ELSE pe=14
1450 NEXT                                    [350]
1460 PRINT CHR$(23)CHR$(1):TAG:PLOT 700,1,   [8292]
2:FOR a=148 TO 500 STEP 64:MOVE a,208:PRIN
T CHR$(231);:NEXT:TAGOFF:PRINT CHR$(23)CHR
$(0)
1470 MOVE 140,182:DRAWR 360,0,0:MOVE 140,2   [2929]
18:DRAWR 360,0
1480 LOCATE 1,1:PEN 1:PRINT"Energie:":LOCA   [4175]
TE 14,1:PRINT"120"
1490 CALL &A07E:CLS                          [1329]
1500 FOR a=1 TO 200:PLOT INT(RND*640),INT(   [3419]
RND*400),INT(RND*2)+1:NEXT
1510 CALL &BC02:INK 0,0:BORDER 0:INK 2,13:   [5250]
INK 1,26:INK 14,0,6:INK 15,6,0:INK 7,12:IN
K 9,13:INK 10,9
1520 meteor1=&9000:meteor2=&9078:meteor3=&   [3657]
90F0:meteor4=&9190
1530 EXP1=&9230:RAUMS=&92D0:mann=&9334:bea   [3711]
m=&936B:leer=&945E
1540 energie=120:lives=3                     [1551]
1550 dif=2                                   [229]
1560 FOR w=1 TO 200:SOUND 1,500-w,20,15:CA   [2729]
LL &A090:NEXT
1570 '                                       [117]
1580 'Hauptprogramm Level 1                  [1904]
1590 '                                       [117]
1600 GOSUB 3500:LOCATE 14,1:PRINT USING"##   [4141]
#";energie:EVERY 5,1 GOSUB 1990
1610 IF treffer THEN 2140 ELSE jo=JOY(0):I   [2539]
F jo=0 THEN dr=0
1620 IF jo=1 AND dr=0 THEN dr=1:GOTO 1700    [2164]
1630 IF jo=2 AND dr=0 THEN dr=1:GOTO 1740    [1341]
1640 IF jo=4 AND dr=0 THEN dr=1:GOTO 1780    [2697]
1650 IF jo=8 AND dr=0 THEN dr=1:GOTO 1820    [1918]
1660 GOTO 1610                               [353]
1670 '                                       [117]
1680 'Schuss nach oben                       [674]
1690 '                                       [117]
1700 MOVE 320,304:DRAWR 0,96,1:GOSUB 2240:   [3028]
GOSUB 1860:DRAWR 0,-96,0:GOTO 1610
1710 '                                       [117]
1720 'Schuss nach unten                      [834]
1730 '                                       [117]
1740 MOVE 320,96:DRAWR 0,-96,1:GOSUB 2240:   [2747]
GOSUB 1860:DRAWR 0,96,0:GOTO 1610
1750 '                                       [117]
1760 'Schuss nach links                      [691]
1770 '                                       [117]
1780 MOVE 136,200:DRAWR -136,0,1:GOSUB 224   [3838]
0:GOSUB 1860:DRAWR 136,0,0:GOTO 1610
1790 '                                       [117]
1800 'Schuss nach rechts                     [758]
1810 '                                       [117]
1820 MOVE 504,200:DRAWR 136,0,1:GOSUB 2240   [5507]
:GOSUB 1860:DRAWR -136,0,0:GOTO 1610
1830 '                                       [117]
1840 'Testen ob Meteor getroffen             [1414]
1850 '                                       [117]
1860 DI:IF (jo AND ri)=1 THEN GOTO 1950      [2343]
1870 IF (jo AND ri)=2 THEN GOTO 1950         [768]
1880 IF (jo AND ri)=4 THEN GOTO 1950         [1578]
1890 IF (jo AND ri)=8 THEN GOTO 1950         [1850]
1900 energie=energie-5:EI:GOTO 2190          [2004]
1910 RETURN                                  [555]
1920 '                                       [117]
1930 'Meteor getroffen                       [1756]
1940 '                                       [117]
1950 DI:SOUND 130,0:CALL &A000,exp1,xm,ym,   [9370]
8,20:CALL &BD19:CALL &BD19:CALL &BD19:CALL
 &A000,leer,xm,ym,8,20:bew=0:energie=energ
ie-1:tref=tref+1:EI:GOTO 2190
1960 '                                       [117]
1970 'Bewegung von Meteoriten                [2715]
1980 '                                       [117]
1990 IF bew=0 THEN wahl=INT(RND*4)+1:bew=1   [2602]
 ELSE GOTO 2020
2000 IF wahl=1 THEN ri=1:xm=37:ym=1-dif EL   [4385]
SE IF wahl=2 THEN ri=4:xm=76-dif/2:ym=84
2010 IF wahl=3 THEN ri=8:xm=75+dif/2:ym=90   [3455]
 ELSE IF wahl=4 THEN ri=2:xm=37:ym=179+dif
2020 SOUND 2,500,10,15,0,2:ON wahl GOSUB 2   [5376]
030,2050,2070,2090:RETURN
2030 ym=ym+dif:IF ym>=28 THEN treffer=-1:q   [5953]
=REMAIN(1) ELSE CALL &A000,meteor1,37,ym,6
,20
2040 RETURN                                  [555]
2050 xm=xm+dif/2:IF xm>=90 THEN treffer=-1   [5866]
:q=REMAIN(1) ELSE CALL &A000,meteor4,xm,84
,8,20
2060 RETURN                                  [555]
2070 xm=xm-dif/2:IF xm<=62 THEN treffer=-1   [6573]
:q=REMAIN(1) ELSE CALL &A000,meteor3,xm,90
,8,20
2080 RETURN                                  [555]
2090 ym=ym-dif:IF ym<=150 THEN treffer=-1:   [7041]
q=REMAIN(1) ELSE CALL &A000,meteor2,37,ym,
6,20
2100 RETURN                                  [555]
2110 '                                       [117]
2120 'Cosmic Star getroffen                  [1139]
2130 '                                       [117]
2140 SOUND 1,1000,100,15,,3,15:CALL &A0C9:   [3886]
q=REMAIN(1):EI
2150 lives=lives-1:IF lives<=0 THEN 3360 E   [2700]
LSE 1600
2160 '                                       [117]
2170 'Energie anzeigen                       [1445]
2180 '                                       [117]
2190 IF energie<=0 THEN 3360 ELSE LOCATE 1   [3129]
4,1:PRINT USING"###";energie
2200 IF tref=dif*8 THEN q=REMAIN(1):GOTO 2   [3677]
250 ELSE GOTO 1910
2210 '                                       [117]
2220 'Schuss Sound                           [742]
2230 '                                       [117]
2240 SOUND 1,90,20,15,,1:RETURN              [2786]
2250 '                                       [117]
2260 'Level 2                                [1122]
2270 '                                       [117]
2280 CALL &A119:LOCATE 14,1:PRINT USING"##   [8133]
#";energie:PEN 2:LOCATE 6,12:PRINT"LANDING
 ON":PEN 1:LOCATE 11-l/2,14:PRINT n$:FOR w
=1 TO 2000:NEXT:MODE 0
2290 POKE &A0C7,&80:POKE &A0C8,&7F:FOR a=1   [5023]
 TO 130:SOUND 2,500-a,20,15:CALL &A090:NEX
T
2300 FOR a=1 TO 14:MOVE a,a:DRAW 640-a,a,9   [2599]
:NEXT
2310 raumx=35:raumy=83:lasery=16:laser=100   [5786]
:m1x=INT(RND*65)+4:m2x=INT(RND*65)+4
2320 zeit=165-(dif*20):mann1=0:mann2=0:mr=   [4742]
0:treffer=0:GOSUB 3000
2330 EVERY 6,1 GOSUB 2460:EVERY 10 GOSUB 2   [1874]
590
2340 IF treffer THEN 3210 ELSE DI:jo=JOY(0   [4404]
):CALL &A000,raums,raumx,raumy,10,10:IF jo
=0 THEN 2420
2350 IF jo=1 AND raumy>83 THEN raumy=raumy   [2990]
-1
2360 IF (jo=1 AND raumy=83 AND raumx>=34 A   [2615]
ND raumx<=36) THEN EI:GOTO 2860
2370 IF jo=2 AND raumy<160 THEN raumy=raum   [2333]
y+1
2380 IF jo=2 AND raumy=83 THEN EI:GOSUB 29   [2394]
70
2390 IF jo=4 AND raumx>5 THEN raumx=raumx-   [2889]
1
2400 IF jo=8 AND raumx<65 THEN raumx=raumx   [2152]
+1
2410 IF jo=18 AND raumy>100 THEN EI:GOTO 2   [2619]
710
2420 EI:GOTO 2340                            [601]
2430 '                                       [117]
2440 'Laser aufbauen                         [2548]
2450 '                                       [117]
2460 IF lasery<>200 THEN MOVE 10,lasery:DR   [11078]
AWR 10,0,1:MOVE 620,lasery:DRAWR 10,0:lase
ry=lasery+2:MOVE 10,lasery:DRAWR 10,0,1:MO
VE 620,lasery:DRAWR 10,0:lasery=lasery+2:R
ETURN
2470 laser=laser+dif:IF laser>=200 THEN la   [3218]
ser=60
2480 IF w<8 THEN w=w+1:RETURN ELSE w=0       [2233]
2490 SOUND 1,2,10,15,0,4,1                   [1281]
2500 MOVE 24,laser:DRAWR 594,0,1:GOSUB 254   [3536]
0:CALL &BD19:DRAWR -594,0,0:RETURN
2510 '                                       [117]
2520 'Testen ob Raumschiff getroffen         [1632]
2530 '                                       [117]
2540 IF laser<400-(raumy*2) AND laser>400-   [6356]
(raumy*2)-15 THEN treffer=-1:q=REMAIN(1)
2550 RETURN                                  [555]
2560 '                                       [117]
2570 'Bewegung von Maennchen                 [2257]
2580 '                                       [117]
2590 IF zeit=0 THEN SOUND 4,250,-10,0,1      [2080]
2600 zeit=zeit-1                             [1941]
2610 IF zeit=-20 THEN GOSUB 3060             [1137]
2620 q=INT(RND*2)+1                          [352]
2630 IF q=1 AND m1x>4 THEN m1x=m1x-1         [1449]
2640 IF q=2 AND m1x<65 THEN m1x=m1x+1        [1530]
2650 q1=INT(RND*2)+1                         [549]
2660 IF q1=1 AND m2x>4 THEN m2x=m2x-1        [1586]
2670 IF q1=2 AND m2x<65 THEN m2x=m2x+1       [1836]
2680 IF NOT mann1 THEN CALL &A000,mann,m1x   [2678]
,181,5,11
2690 IF NOT mann2 THEN CALL &A000,mann,m2x   [3112]
,181,5,11
2700 RETURN                                  [555]
2710 IF raumx+2=m1x AND NOT mann1 THEN mx=   [8598]
m1x:my=181:q=REMAIN(0) ELSE IF raumx+2=m2x
 AND NOT mann2 THEN mx=m2x:my=181:q=REMAIN
(0) ELSE GOTO 2340
2720 bl=my-raumy:WHILE JOY(0)=18 AND my>ra   [3125]
umy+10:SOUND 2,1000,2,4
2730 CALL &A000,beam,mx+1,raumy+10,3,bl:CA   [3411]
LL &A000,mann,mx,my,5,11:my=my-0.5:WEND
2740 CALL &A000,leer,mx+1,raumy+10,3,bl      [1472]
2750 IF my<=raumy+10 AND raumx+2=m1x THEN    [3189]
mann1=-1
```

Listing "COSMIC"

```
2760 IF my<=raumy+10 AND raumx+2=m2x THEN mann2=-1 [3105]
2770 EVERY 10 GOSUB 2590:GOTO 2340 [1418]
2780 ' [117]
2790 'Meteor bewegen [1209]
2800 ' [117]
2810 SOUND 2,500,10,15,0,2:ON wahl GOTO 2820,2830 [3915]
2820 xm=xm-dif/2:IF xm<=62 THEN treffer=-1:q=REMAIN(1):RETURN ELSE CALL &A000,meteor3,xm,20,8,20:RETURN [5899]
2830 xm=xm+dif/2:IF xm>=10 THEN treffer=-1:q=REMAIN(1):RETURN ELSE CALL &A000,meteor4,xm,20,8,20:RETURN [6736]
2840 IF jo=8 AND mr=8 THEN q=REMAIN(1):CALL &A000,exp1,xm,20,8,20:GOTO 3140 ELSE GOTO 2920 [4213]
2850 IF jo=4 AND mr=4 THEN q=REMAIN(1):CALL &A000,exp1,xm,20,8,20:GOTO 3140 ELSE GOTO 2920 [6041]
2860 ' [117]
2870 'Raumschiff kehrt ins Mutterschiff zu rueck [3829]
2880 ' [117]
2890 y=10:r=raums:wa=1:WHILE y>1 [2753]
2900 CALL &A000,r,raumx,83,10,y:y=y-1:r=r+10:WHILE wa<10:wa=wa+1:WEND:wa=1 [5353]
2910 WEND [390]
2920 IF treffer THEN 3210 ELSE jo=JOY(0):IF jo=0 THEN dr=0:GOTO 2920 [3727]
2930 IF jo=2 THEN GOSUB 3000:GOTO 2340 [2611]
2940 IF jo=4 AND dr=0 THEN dr=1:DI:MOVE 136,340:DRAWR -136,0,1:GOSUB 2240:MOVE 136,340:DRAWR -136,0,0:EI:GOTO 2850 [6310]
2950 IF jo=8 AND dr=0 THEN dr=1:DI:MOVE 504,340:DRAWR 136,0,1:GOSUB 2240:MOVE 504,340:DRAWR 136,0,0:EI:GOTO 2840 [6894]
2960 GOTO 2920 [339]
2970 ' [117]
2980 'Raumschiff verlaesst Mutterschiff [2336]
2990 ' [117]
3000 wa=1:r=raums+100:y=wa:WHILE y<10:CALL &A000,r,35,83,10,y:r=r-10:y=y+1 [5977]
3010 WHILE wa<10:wa=wa+1:WEND:wa=1:WEND [2001]
3020 raumx=35:raumy=83:RETURN [3653]
3030 ' [117]
3040 'Meteor recht oder links [1957]
3050 ' [117]
3060 q=REMAIN(0)+REMAIN(1) [1023]
3070 wahl=INT(RND*2)+1 [1593]
3080 IF wahl=1 THEN mr=8:xm=75+dif/2 [1295]
3090 IF wahl=2 THEN mr=4:xm=0-dif/2 [1444]
3100 EVERY 6,1 GOSUB 2810:RETURN [926]
3110 ' [117]
3120 'Getroffen [769]
3130 ' [117]
3140 CALL &BD19:CALL &BD19:CALL &BD19:CALL &A000,leer,xm,20,8,20:IF mann1 THEN energie=energie+10 [7293]
3150 IF mann2 THEN enrgie=enrgie+10 [1543]
3160 CALL &A119:dif=dif+1 [820]
3170 GOTO 3310 [451]
3180 ' [117]
3190 'Laser trifft Raumschiff [1517]
3200 ' [117]
3210 q=REMAIN(1)+REMAIN(0):IF zeit=-20 THEN GOTO 2140 ELSE SOUND 4,0,25,15,,2,15 :CALL &A000,exp1,raumx,raumy,8,20:CALL &BD19:CALL &BD19:CALL &BD19:FOR w=1 TO 1000:NEXT:CALL &A000,leer,raumx,raumy,8,20 [11950]
3220 treffer=0:energie=energie-15 [2210]
3230 CALL &A119:GOTO 3170 [831]
3240 ' [117]
3250 'Bildschirm dunkel [1352]
3260 ' [117]
3270 MODE 0:FOR i=0 TO 15:INK i,0:NEXT:BORDER 0:RETURN [3264]
3280 ' [117]
3290 'Testen ob Energie=0 [682]
3300 ' [117]
3310 IF energie<=0 THEN GOSUB 3360 'the END [2442]
3320 EI:GOTO 1600 [605]
3330 ' [117]
3340 'The END [376]
3350 ' [117]
3360 q=REMAIN(0)+REMAIN(1) [1023]
3370 CALL &A119:LOCATE 14,1:PRINT"  0":WINDOW #1,7,15,8,16:PAPER#1,7:CLS#1 [3946]
```

Listing "COSMIC"

```
3380 LOCATE #1,1,5:PRINT #1,"GAME OVER" [1624]
3390 FOR a=500 TO 1000 STEP 4:SOUND 1,a,1,13:NEXT [1940]
3400 FOR w=1 TO 2000:NEXT [1334]
3410 LOCATE #1,1,5:PRINT #1,"Once More" [2032]
3420 LOCATE #1,1,6:PRINT #1,"  (J/N)  " [2655]
3430 k$=UPPER$(INKEY$):IF k$="" THEN 3430 [1357]
3440 IF k$="J" THEN RUN [1364]
3450 IF k$="N" THEN MODE 2:CALL &BC02:END [1285]
3460 GOTO 3430 [515]
3470 ' [117]
3480 'Planetenname anzeigen [888]
3490 ' [117]
3500 IF dif=2 THEN n$="CYLONIEN" [837]
3510 IF dif=3 THEN n$="REBACA" [1276]
3520 IF dif=4 THEN n$="MORBUS" [1186]
3530 IF dif=5 THEN n$="HONOURUS" [847]
3540 IF dif=6 THEN 3620 [814]
3550 CALL &A119:LOCATE 14,1:PRINT USING"##";energie:bew=0:tref=0:treffer=0 [4650]
3560 PEN 2:l=LEN(n$):LOCATE 8,12:PRINT"Rescue" [3351]
3570 LOCATE 11-l/2,14:PEN 1:PRINT n$:FOR w=1 TO 2000:NEXT [2806]
3580 CALL &A119:RETURN [1278]
3590 ' [117]
3600 'Alles geschaft ..... [927]
3610 ' [117]
3620 DI:CALL &A119:LOCATE 1,1:PRINT"  ";  [2064]
3630 PEN 14:LOCATE 4,10:PRINT"CONGRATULATION" [2670]
3640 PEN 6:LOCATE 4,12:PRINT"   you have  " [2921]
3650 PEN 12:LOCATE 4,14:PRINT"    rescue  " [4104]
3660 PEN 2:LOCATE 4,16:PRINT" all Plantets " [3195]
3670 FOR a=1 TO 3:SOUND 1,200,25,15:SOUND 1,0:SOUND 1,200,25,15:SOUND 1,0:SOUND 1,150,50,15:SOUND 1,0:NEXT [6007]
3680 WINDOW #1,7,15,8,16:PAPER#1,7:CLS#1 [2774]
3690 LOCATE #1,1,5:PRINT #1,"Once More" [2032]
3700 LOCATE #1,1,6:PRINT #1,"  (J/N)  " [2655]
3710 k$=UPPER$(INKEY$):IF k$="" THEN 3710 [1565]
3720 IF k$="J" THEN RUN [1364]
3730 IF k$="N" THEN MODE 2:CALL &BC02:PEN 1:END [1546]
3740 GOTO 3710 [499]
```

```
1000 '*************** [345]
1010 '*  Data-Lader * [850]
1020 '*     zu      * [298]
1030 '*  Cosmic.mc  * [970]
1040 '*************** [345]
1050 ' [117]
1060 ' [117]
1070 ' [117]
1080 MEMORY &9FFF [134]
1090 FOR adr=&A000 TO &A124 [1128]
1100 READ a$:sum=sum+VAL("&"+a$) [1839]
1110 POKE adr,VAL("&"+a$) [718]
1120 NEXT [350]
1130 IF sum<>35903 THEN PRINT"Data-ERROR !!!":END [3628]
1140 PRINT"Absaven: Press a key !":CALL &BB18 [2923]
1150 SAVE"COSMIC.MC",b,&A000,&125 [1667]
1160 END [110]
1170 ' [117]
1180 DATA DD,46,00,DD,4E,02,DD,66 [1654]
1190 DATA 04,DD,6E,06,DD,56,09,DD [1379]
1200 DATA 5E,08,E5,CD,23,A0,C5,06 [1624]
1210 DATA 00,EB,ED,B0,EB,C1,E1,24 [1199]
1220 DATA 10,F0,C9,D5,7C,E5,E6,07 [976]
1230 DATA 21,44,A0,85,6F,56,E1,5D [1838]
1240 DATA D5,7C,E6,F8,CB,3F,CB,3F [1204]
1250 DATA 21,4C,A0,85,6F,5E,23,56 [1609]
1260 DATA E1,19,D1,C9,00,08,10,18 [1318]
1270 DATA 20,28,30,38,00,C0,50,C0 [1223]
1280 DATA A0,C0,F0,C0,40,C1,90,C1 [1019]
1290 DATA E0,C1,30,C2,80,C2,D0,C2 [1411]
1300 DATA 20,C3,70,C3,C0,C3,10,C4 [1512]
1310 DATA 60,C4,B0,C4,00,C5,50,C5 [1072]
1320 DATA A0,C5,F0,C5,40,C6,90,C6 [1441]
1330 DATA E0,C6,30,C7,80,C7,21,00 [1978]
1340 DATA C0,11,00,40,01,FF,3F,ED [1387]
```

Listing "COSMIC"


```
1350 DATA B0,21,80,7F,22,C7,A0,C9 [969]
1360 DATA CD,19,BD,21,80,F7,11,80 [1340]
1370 DATA FF,06,C7,C5,D5,E5,01,50 [1678]
1380 DATA 00,ED,B0,E1,D1,CD,29,BC [840]
1390 DATA EB,CD,29,BC,EB,C1,10,EB [432]
1400 DATA 2A,C7,A0,E5,11,00,C0,01 [842]
1410 DATA 50,00,ED,B0,E1,01,00,80 [1731]
1420 DATA CD,29,BC,22,C7,A0,C9,80 [1521]
1430 DATA 7F,1E,03,21,00,C0,01,00 [1672]
1440 DATA 40,55,7E,1F,E6,77,CB,1A [1625]
1450 DATA 30,02,F6,08,CB,5A,28,02 [1512]
1460 DATA F6,80,56,A2,77,23,0B,78 [1139]
1470 DATA B1,20,E7,51,01,00,40,2B [878]
1480 DATA 7E,17,E6,EE,CB,12,30,02 [1711]
1490 DATA F6,10,CB,62,28,02,F6,01 [1263]
1500 DATA 56,A2,77,0B,78,B1,20,E7 [974]
1510 DATA 1D,20,C0,21,00,C0,01,FF [1302]
1520 DATA 3F,11,01,C0,AF,77,ED,B0 [1260]
1530 DATA C9,21,00,40,11,00,C0,01 [1252]
1540 DATA FF,3F,ED,B0,C9,00,00,00 [1270]
```

```
1000 '*********** [502]
1010 '*Data-LADER* [850]
1020 '*    zu    * [298]
1030 '*COSMIC.SPR* [1491]
1040 '*********** [502]
1050 ' [117]
1060 ' [117]
1070 ' [117]
1080 MEMORY &8FFF [207]
1090 FOR adr=&9000 TO &9551 [916]
1100 READ a$:POKE adr,VAL("&"+a$) [1712]
1110 sum=sum+VAL("&"+a$) [1393]
1120 NEXT [350]
1130 IF sum<>86540 THEN PRINT"DATA-ERROR !":END [1801]
1140 PRINT "Absaven: Press a key !":CALL &BB18 [2923]
1150 SAVE"COSMIC.SPR",b,&9000,&552 [1710]
1160 END [110]
1170 ' [117]
1180 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1190 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1200 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1210 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1220 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1230 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1240 DATA 00,00,00,54,FC,00,00,00,00,FC [669]
1250 DATA C3,A8,54,A8,FC,C3,C3,D6,54,D6 [1749]
1260 DATA C3,4B,C3,5E,E9,C3,4B,C3,4B,D6 [2092]
1270 DATA E9,C3,C3,C3,C3,A8,FC,C3,87,C3 [1132]
1280 DATA C3,A8,00,E9,C3,C3,D6,00,00,54 [1975]
1290 DATA E9,FC,D6,00,00,00,54,A8,A8,00 [845]
1300 DATA 00,FC,A8,FC,FC,A8,54,E9,D6,E9 [1890]
1310 DATA C3,D6,54,C3,C3,C3,C3,D6,E9,C3 [1647]
1320 DATA 87,87,87,D6,E9,87,87,C3,C3,D6 [1227]
1330 DATA 54,C3,C3,C3,C3,FC,00,FC,87,4B [1524]
1340 DATA C3,A8,00,54,E9,C3,D6,00,00,00 [2031]
1350 DATA 54,D6,FC,00,00,00,54,FC,00,00 [1620]
1360 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1370 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1380 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1390 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1400 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1410 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1420 DATA 00,00,00,00,00,00,00,00,00,54 [840]
1430 DATA A8,00,00,00,00,00,54,E9,A8,00 [1965]
1440 DATA 00,00,00,00,54,C3,D6,00,00,00 [1603]
1450 DATA 00,00,54,C3,D6,00,00,00,00,00 [525]
1460 DATA AD,87,D6,00,00,00,00,00,E9,C3 [1559]
1470 DATA D6,00,00,00,00,00,E9,87,D6,00 [1459]
1480 DATA 00,00,00,00,FC,C3,A8,00,00,00 [1798]
1490 DATA 00,00,54,C3,A8,00,00,00,00,00 [1287]
1500 DATA 54,C3,A8,00,00,00,00,00,54,4B [1371]
1510 DATA A8,00,00,00,00,00,54,C3,D6,00 [1957]
1520 DATA 00,00,00,00,54,C3,C3,00,00,00 [1709]
1530 DATA 00,00,E9,87,C3,A8,00,00,00,00 [1326]
1540 DATA E9,C3,C3,A8,00,00,00,00,E9,4B [1211]
1550 DATA D6,00,00,00,00,00,54,C3,A8,00 [1976]
1560 DATA 00,00,00,00,54,C3,A8,00,00,00 [1786]
1570 DATA 00,00,54,FC,00,00,00,00,00,00 [1394]
1580 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1590 DATA 00,00,00,00,FC,00,00,00,00,00 [1716]
1600 DATA 00,54,D6,00,00,00,00,00,00,54 [1633]
1610 DATA C3,A8,00,00,00,00,00,FC,C3,A8 [2654]
1620 DATA 00,00,00,00,00,E9,C3,D6,00,00 [1440]
1630 DATA 00,00,00,E9,C3,C3,00,00,00,00 [689]
1640 DATA 00,AD,87,D6,00,00,00,00,54,C3 [1107]
```

Listing "COSMIC"

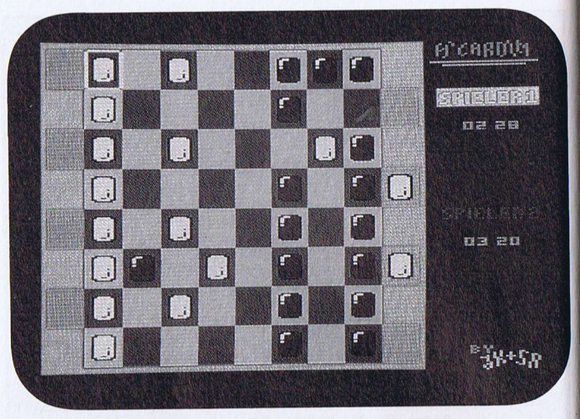
```
1650 DATA C3,FC,00,00,00,00,54,C3,C3,A8 [1594]
1660 DATA 00,00,00,00,54,4B,4B,A8,00,00 [1951]
1670 DATA 00,00,00,E9,D6,00,00,00,00,00 [1196]
1680 DATA 00,E9,D6,00,00,00,00,00,00,E9 [1045]
1690 DATA 4B,A8,00,00,00,00,00,E9,C3,A8 [2341]
1700 DATA 00,00,00,00,00,00,E9,C3,FC,00,00 [1511]
1710 DATA 00,00,00,FC,87,D6,00,00,00,00 [1397]
1720 DATA 00,54,C3,D6,00,00,00,00,00,54 [1332]
1730 DATA E9,A8,00,00,00,00,00,00,FC,A8 [1125]
1740 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1750 DATA 00,00,0A,00,00,00,00,00,00,0A [925]
1760 DATA 0A,05,00,00,00,00,00,0A,0A,05 [1712]
1770 DATA 00,00,00,05,05,00,0A,05,00,00 [1874]
1780 DATA 00,0A,05,00,0A,05,00,0A,00,0A [1478]
1790 DATA 05,0F,0A,0A,05,00,00,0F,0F,33 [1598]
1800 DATA 27,0A,05,00,05,0A,05,33,33,33 [1799]
1810 DATA 0A,00,00,00,1B,62,91,33,0A,00 [1630]
1820 DATA 05,00,1B,62,91,27,00,00,00,0A [1413]
1830 DATA 1B,62,91,27,0F,00,00,05,1B,62 [1249]
1840 DATA 91,27,00,0F,00,00,05,62,91,33 [1278]
1850 DATA 0A,00,00,00,05,33,33,27,0F,00 [1409]
1860 DATA 00,00,0A,0F,33,0A,05,0A,00,05 [1081]
1870 DATA 00,00,0F,0A,00,0A,00,0A,00,05 [1082]
1880 DATA 00,0A,00,00,00,00,00,0A,00,05 [2191]
1890 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1900 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1910 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1920 DATA 00,00,00,00,00,01,02,00,00,00 [1076]
1930 DATA 00,00,00,01,03,03,02,00,00,00 [1580]
1940 DATA 00,00,04,0C,0C,0C,0C,08,00,00 [1655]
1950 DATA 00,00,0C,0C,0C,0C,0C,0C,00,00 [1411]
1960 DATA 00,00,04,0C,0C,0C,0C,08,00,00 [1655]
1970 DATA 00,00,00,15,3F,3F,2A,00,00,00 [1731]
1980 DATA 00,00,00,00,00,00,00,00,00,00 [822]
1990 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2000 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2010 DATA 00,00,00,00,00,00,00,F3,00,00 [1195]
2020 DATA 00,45,F3,8A,00,00,51,F3,A2,00 [1225]
2030 DATA 00,51,F3,A2,00,00,00,CF,00,00 [1329]
2040 DATA 00,45,00,8A,00,00,00,00,00,00 [1483]
2050 DATA 00,00,00,00,00,C3,C3,C3,C3,C3 [906]
2060 DATA C3,C3,C3,C3,C3,C3,C3,C3,C3,C3 [2488]
2070 DATA C3,C3,63,C3,C3,C3,93,C3,C3,C3 [1978]
2080 DATA C3,63,C3,C3,C3,93,C3,C3,C3,C3 [1274]
2090 DATA C3,C3,C3,C3,C3,C3,63,C3,C3,C3 [2084]
2100 DATA C3,C3,C3,C3,93,C3,C3,C3,63,C3 [2345]
2110 DATA C3,C3,C3,63,C3,C3,C3,C3,C3,C3 [2249]
2120 DATA C3,C3,C3,C3,C3,93,C3,C3,C3,C3 [2082]
2130 DATA C3,C3,C3,C3,C3,C3,C3,93,C3,C3 [2508]
2140 DATA 63,C3,C3,C3,C3,C3,C3,C3,63,C3 [996]
2150 DATA C3,C3,C3,C3,C3,63,93,C3,C3,C3 [1871]
2160 DATA C3,C3,93,C3,C3,C3,63,C3,C3,C3 [2028]
2170 DATA C3,C3,C3,63,93,C3,C3,C3,C3,C3 [2296]
2180 DATA C3,C3,C3,C3,C3,C3,C3,93,C3,C3 [2283]
2190 DATA C3,C3,C3,C3,C3,C3,C3,C3,63,C3 [2345]
2200 DATA C3,C3,C3,C3,C3,93,C3,93,C3,C3 [1833]
2210 DATA C3,C3,63,C3,C3,C3,63,C3,C3,C3 [2124]
2220 DATA C3,C3,C3,C3,C3,C3,C3,C3,63,C3 [2345]
2230 DATA 63,C3,C3,C3,C3,C3,C3,C3,C3,C3 [1153]
2240 DATA C3,93,C3,93,C3,C3,C3,C3,C3,C3 [1651]
2250 DATA C3,93,C3,93,C3,C3,C3,C3,C3,C3 [1651]
2260 DATA C3,C3,C3,C3,C3,C3,93,C3,C3,C3 [2286]
2270 DATA C3,C3,C3,C3,C3,C3,93,C3,C3,C3 [2319]
2280 DATA C3,63,C3,C3,C3,C3,C3,93,C3,C3 [1023]
2290 DATA C3,C3,93,C3,C3,C3,C3,C3,00,00 [1727]
2300 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2310 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2320 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2330 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2340 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2350 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2360 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2370 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2380 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2390 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2400 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2410 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2420 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2430 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2440 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2450 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2460 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2470 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2480 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2490 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2500 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2510 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2520 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2530 DATA 00,00,00,00,00,00,00,00,00,00 [822]
2540 DATA 00,00,00,00,00,00,00,00,00,00 [822]
```

Listing "COSMIC"

H'CARDIN

Ein Strategiespiel für zwei Spieler (von 7 bis 77 Jahre)

Der Planet Ni'drac kreist um den Stern Merak im Sternbild des Großen Bären. Seine Bewohner sind leidenschaftliche Spieler. Eines der beliebtesten Spiele dort ist H'Cardin. Es finden alljährlich Wettkämpfe statt, die jedesmal eines der bedeutendsten Spektakel sind.



H'Cardin - wer gewinnt das knifflige Spiel?

Aus zweifelhaften Quellen erhielten wir vor wenigen Wochen unter mysteriösen Umständen die Spielregeln dieses außergewöhnlichen Spieles.

Wir wollten es natürlich den Lesern nicht vorenthalten, und machten uns sofort an die Arbeit. Hier ist das Ergebnis:

Es spielen zwei Spieler gegeneinander. Jeder von beiden hat 16 Spielsteine, die es gilt, in die 4 Felder hinter die gegnerischen Reihen zu plazieren. Jeder Stein kann in alle acht Richtungen um ein Feld bewegt werden, also vertikal, horizontal und diagonal.

Genauso ist es möglich, über einen beliebigen Stein zu springen. Es sind auch Mehrfachsprünge möglich. Wird durch Zug ein gegnerischer Stein von zwei eigenen Steinen eingeschlossen, so wird er vom Feld entfernt.

Spieler 1 (weiß) steuert mit den Cursortasten sowie der Leertaste, Spieler 2 (schwarz) mit dem Joystick.

Viel Spaß beim Spielen!
(Steffen Rau&Jens Köplinger/sr)

für 464-664-6128

```
10 '    +---------------+                                [1177]
20 '  :                 :                                [255]
30 ' :   H ' C a r d i n   :                              [115]
40 '  :                 :                                [255]
50 '    +---------------+                                [1177]
60 '                                                     [117]
70 ' this programm was designed and writte               [4937]
n in 1988 by
80 '                                                     [117]
90 '   Jens Koeplinger  &  Steffen Rau                   [1566]
100 ' =================   =============                  [1959]
110 '                                                    [117]
120 '  ************************                          [978]
130 ' ***     Ladeprogramm     ***                       [1680]
140 '  ************************                          [978]
150 '                                                    [117]
160 ' +---------------+                                  [1177]
170 ' :  Titelgrafik  :                                  [1061]
180 ' +---------------+                                  [1177]
190 '                                                    [117]
200 MODE 1:INK 0,0:INK 1,6:INK 2,0:INK 3,2               [2512]
4:BORDER 0
210 PEN 2:LOCATE 1,25:PRINT"H'Cardin"CHR$(               [2829]
23)CHR$(3)
220 FOR f=3 TO 1 STEP-2                                  [1466]
230   FOR x=0 TO 126 STEP 2                              [943]
240     FOR y=0 TO 14 STEP 2                             [1487]
250       a=TEST(x,y)                                    [1239]
260       IF a=0 THEN 290                                [524]
270       MOVE 150-f*3+y+x*3+g,300+y*3+f*4               [2382]
:DRAWR 2,4,f
280       MOVER 0,-4:DRAWR 2,4:MOVER 0,-4:               [2032]
DRAWR 2,4
290     NEXT y                                           [359]
300   NEXT x                                             [356]
310 NEXT f                                               [374]
```

Listing "H'CARDIN"

```
320 LOCATE 1,25:PRINT"          "CHR$(23)CHR             [2429]
$(0):INK 2,18
330 FOR g=0 TO 4 STEP 2                                  [709]
340   PLOT 100,290+g,2:DRAWR 464,0                       [711]
350   PLOT 104+g*2,286-g:DRAWR 464,0:DRAWR               [2896]
 0,6:PLOTR 2,0:DRAWR 0,-6
360 NEXT g                                               [377]
370 PLOTR 0,0,0:DRAWR-14,6                               [1568]
380 PEN 2:LOCATE 5,11:PRINT"this excellent               [4087]
 game was designed"
390 LOCATE 10,13:PRINT"and written in 1988               [4345]
 by"
400 PRINT CHR$(23)CHR$(1);                               [1919]
410 MOVE 158,160:DRAW 464,160,2:DRAW 464,4               [3477]
6:DRAW 158,46:DRAW 158,160
420 PRINT CHR$(23)CHR$(3);                               [1911]
430 PEN 3:LOCATE 13,17:PRINT"Jens Koepling               [3322]
er"
440 LOCATE 15,21:PRINT"Steffen Rau"                      [1155]
450 PEN 2:LOCATE 20,19:PRINT"&"                          [1571]
460 PLOT-10,10,1:TAG                                     [333]
470 MOVE 196,140:PRINT"Jens Koeplinger";                 [2619]
480 MOVE 228,76:PRINT"Steffen Rau";:TAGOFF               [1963]
490 LOCATE 26,25:PRINT"please wait ...";                 [2441]
500 '                                                    [117]
510 ' +-------------------------------------             [1966]
----+
520 ' :  MCode poken und Hauptprogramm lad               [3848]
en  :
530 ' +-------------------------------------             [1966]
----+
540 '                                                    [117]
550 SYMBOL AFTER 256:MEMORY &9FFF:adr=&A00               [3434]
0
560 FOR z=68 TO 130                                      [978]
570   summe=0                                            [552]
580   FOR g=1 TO 16                                      [805]
590     READ a$:a=VAL("&"+a$)                            [883]
600     summe=summe+a:POKE adr,a:adr=adr+1               [1485]
610   NEXT g                                             [377]
620   READ a:IF a<>summe THEN PRINT"Fehler               [3550]
 in"z*10:END
```

Listing "H'CARDIN"

```
630 NEXT z                                               [354]
640 RUN"h'main                                           [259]
650 '                                                    [117]
660 ' +++  Daten fuer MCode  +++                         [1685]
670 '                                                    [117]
680 DATA 06,10,0E,04,E5,1A,AE,77,23,13,0D,               [3187]
C2,05,A0,E1,CD, 1444
690 DATA 4C,A0,10,EE,C9,CD,1F,A0,C3,29,A0,               [2129]
4D,A3,F2,DB,2A, 2226
700 DATA 1D,A0,ED,5B,1B,A0,C3,00,A0,DD;66,               [3288]
03,DD,6E,02,DD, 1939
710 DATA 56,05,DD,5E,04,DD,E5,CD,1D,BC,DD,               [2749]
E1,DD,56,01,DD, 2257
720 DATA 5E,00,22,1D,A0,ED,53,1B,A0,C3,00,               [2440]
A0,D5,11,00,08, 1417
730 DATA 19,D2,58,A0,11,50,C0,19,D1,C9,04,               [2478]
0C,0C,08,0C,C0, 1447
740 DATA C0,0C,48,C0,C0,84,48,C0,C0,84,48,               [2147]
C0,C0,84,48,C0, 2232
750 DATA C0,84,48,C0,C0,84,48,C0,C0,84,48,               [2777]
C0,84,84,48,C0, 2292
760 DATA C0,84,48,C0,84,84,48,C0,84,84,48,               [2741]
C0,84,84,48,84, 2112
770 DATA 48,84,0C,C0,C0,0C,04,0C,0C,08,04,               [2748]
0C,0C,08,0C,C0, 888
780 DATA 0C,0C,48,0C,0C,0C,48,0C,0C,0C,48,               [1642]
0C,0C,0C,0C,0C, 372
790 DATA 0C,0C,48,0C,0C,0C,0C,0C,0C,0C,0C,               [2209]
0C,0C,0C,0C,0C, 252
800 DATA 0C,0C,0C,0C,0C,0C,0C,0C,0C,0C,0C,               [2950]
0C,0C,0C,0C,0C, 192
810 DATA 0C,0C,0C,0C,0C,0C,04,0C,0C,08,F0,               [2692]
00,A0,A0,00,00, 668
820 DATA 00,00,00,00,00,A0,A0,A0,A0,00,00,               [2071]
00,00,00,00,00, 640
830 DATA F0,00,50,00,00,00,00,00,00,00,00,               [1167]
A0,A0,50,00,40, 784
840 DATA 00,00,40,80,00,00,F0,00,50,40,40,               [2527]
00,80,40,40,00, 896
850 DATA 00,00,00,00,40,40,00,80,40,00,40,               [3044]
80,00,00,40,40, 640
860 DATA 40,40,C0,40,00,40,40,00,40,C0,40,               [2956]
80,00,80,40,00, 1152
870 DATA 40,40,00,00,40,00,C0,00,80,00,80,               [1771]
40,40,00,00,40, 832
880 DATA 00,80,80,00,00,80,40,80,00,00,00,               [2484]
80,80,80,00,80, 1088
890 DATA 80,80,80,00,00,40,80,80,00,00,40,               [2563]
80,80,80,00,00, 1152
900 DATA 80,80,80,00,00,00,00,80,80,00,00,               [2584]
80,80,00,00,00, 896
910 DATA 00,00,80,80,00,00,40,00,00,00,00,               [3068]
00,00,00,00,C3, 515
920 DATA C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,               [3525]
C3,C3,C3,C3,C3, 3120
930 DATA C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,               [3525]
C3,C3,C3,C3,C3, 3120
940 DATA 3C,96,69,96,96,3C,96,C3,96,3C,96,               [3530]
69,C3,96,C3,96, 2175
950 DATA C3,96,96,96,96,C3,96,C3,96,C3,96,               [1840]
96,C3,3C,C3,96, 2580
960 DATA C3,96,96,96,96,C3,96,C3,96,C3,96,               [2115]
96,C3,96,C3,C3, 2715
970 DATA 69,96,69,96,96,69,96,C3,96,69,96,               [2421]
69,C3,96,C3,C3, 2355
980 DATA 96,96,C3,96,96,C3,96,C3,96,C3,96,               [2938]
96,C3,96,C3,C3, 2715
990 DATA 96,96,C3,96,96,C3,96,C3,96,C3,96,               [2617]
96,C3,96,C3,96, 2670
1000 DATA 69,96,C3,96,96,3C,96,3C,96,3C,96               [2184]
,96,C3,3C,69,C3, 2085
1010 DATA C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,C3               [3525]
,C3,C3,C3,C3,C3, 3120
1020 DATA C3,C3,C3,C3,C3,C3,C3,C3,C3,C3,C3               [3837]
,C3,C3,C3,C3,3F, 2988
1030 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F               [1211]
,3F,3F,3F,3F,3F, 1008
1040 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F               [1211]
,3F,3F,3F,3F,3F, 1008
1050 DATA 0F,2F,1F,2F,2F,0F,2F,3F,2F,0F,2F               [2883]
,1F,3F,2F,3F,2F, 672
1060 DATA 3F,2F,2F,2F,2F,3F,2F,3F,2F,3F,2F               [3034]
,2F,3F,1F,1F,2F, 800
1070 DATA 3F,2F,2F,2F,2F,3F,2F,3F,2F,3F,2F               [2461]
,2F,3F,3F,1F,3F, 848
```

Listing "H'CARDIN"

```
1080 DATA 1F,2F,1F,2F,2F,1F,2F,3F,2F,1F,2F               [1892]
,1F,3F,2F,3F,3F, 736
1090 DATA 2F,2F,3F,2F,2F,3F,2F,3F,2F,3F,2F               [3030]
,2F,3F,1F,3F,3F, 848
1100 DATA 2F,2F,3F,2F,2F,3F,2F,3F,2F,3F,2F               [3186]
,2F,3F,1F,3F,2F, 832
1110 DATA 1F,2F,3F,2F,2F,0F,2F,0F,2F,0F,2F               [2937]
,2F,3F,0F,1F,3F, 640
1120 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F               [1211]
,3F,3F,3F,3F,3F, 1008
1130 DATA 3F,3F,3F,3F,3F,3F,3F,3F,3F,3F,3F               [1340]
,3F,3F,3F,3F,F3, 1188
1140 DATA A2,A2,A2,A2,A2,A2,A2,F3,A2,51,00               [2474]
,F3,00,51,00,51, 2025
1150 DATA 00,F3,A2,F3,A2,00,A2,F3,A2,A2,00               [2875]
,F3,A2,F3,A2,00, 2349
1160 DATA A2,51,A2,00,A2,F3,A2,A2,00,A2,A2               [2969]
,F3,A2,00,A2,00, 2025
1170 DATA A2,F3,A2,A2,00,F3,A2,00,A2,F3,A2               [2193]
,F3,A2,A2,00,F3, 2511
1180 DATA A2,A2,A2,F3,A2,F3,A2,00,A2,51,00               [2567]
,A2,00,A2,00,F3, 2106
1190 DATA A2,A2,A2,F3,A2,A2,A2,F3,A2,F3,A2               [2930]
,A2,A2,F3,A2,00, 2754
1200 DATA A2,F3,A2,F3,A2,A2,A2,A2,A2,A2,A2               [3190]
,F3,A2,00,00,A0, 2509
1210 DATA 00,00,00,00,00,00,00,00,A0,50,00               [2151]
,00,50,00,50,00, 400
1220 DATA 50,00,A0,00,F0,50,A0,A0,50,00,00               [3086]
,A0,50,50,00,A0, 1440
1230 DATA 50,50,50,50,50,50,50,50,A0,A0,A0               [3067]
,50,00,00,A0,50, 1440
1240 DATA 50,50,50,50,50,50,50,50,A0,A0,F0               [3192]
,00,50,00,50,F0, 1600
1250 DATA 50,50,50,50,50,50,50,A0,F0,50,00               [2954]
,50,00,50,50,50, 1360
1260 DATA A0,50,50,50,50,00,A0,A0,50,00,50               [3421]
,00,50,50,50,50, 1280
1270 DATA 50,50,50,50,00,A0,A0,50,00,50,50               [2745]
,50,50,50,50,50, 1280
1280 DATA 50,50,00,A0,A0,A0,50,00,00,A0,00               [3018]
,50,50,50,50,50, 1280
1290 DATA 00,A0,00,A0,00,A0,00,00,00,00,00               [3186]
,50,00,00,A0,00, 720
1300 DATA A0,00,A0,00,00,00,00,00,00,00,00               [2974]
,00,00,00,00,00, 320
```

```
10 '    +---------------+                                [1177]
20 '  :                 :                                [255]
30 ' :   H ' C a r d i n   :                              [115]
40 '  :                 :                                [255]
50 '    +---------------+                                [1177]
60 '                                                     [117]
70 '  ************************                           [1051]
80 ' ***     Hauptprogramm     ***                       [1683]
90 '  ************************                           [1051]
100 '                                                    [117]
110 ' +--------------------+                             [1102]
120 ' :  Initialisierung  :                              [1154]
130 ' +--------------------+                             [1102]
140 '                                                    [117]
150 LOCATE 24,25:PRINT"press any key ...";               [2539]
160 WHILE INKEY$<>"":WEND:WHILE INKEY$="":               [3680]
WEND
170 ENV 1,1,0,15,12,-1,30:ENV 2,1,0,10,1,-               [2945]
2,1,12,-1,30:ENV 3,1,0,5,14,-1,4
180 ENV 4,1,0,5,14,-1,2:ENT 1,100,-1,7                   [1838]
190 PAPER 0:PEN 1:CALL &BC02:BORDER 0:MODE               [3116]
 0
200 FOR g=0 TO 15                                        [660]
210   READ a:INK g,a                                     [1224]
220 NEXT g                                               [377]
230 DEFINT b-y:DIM feld(9,9),t(4,2),adr(2)               [6133]
,us(6,4),hs1(7),hs2(7),ton(65),sieg(1),z(4
,4),ab(4),rx(7),ry(7)
240 FOR s=1 TO 2                                         [1025]
250   FOR g=0 TO 4                                      [663]
260     READ t(g,s)                                      [1023]
270   NEXT g                                             [377]
```

Listing "H'CARDIN"

```
280 NEXT s                                                [365]
290 FOR g=0 TO 7                                          [668]
300   READ rx(g),ry(g)                                    [1401]
310 NEXT g                                                [377]
320 adr(0)=&A3E3:adr(1)=&A05A:adr(2)=&A09A                [2003]
330 DATA 1,26,0,13,24,26,0,12,3,26,0,0,0,0                [2328]
,26,9
340 DATA 0,1,2,8,47, 72,75,73,74,76                       [942]
350 DATA 0,1, 1,1, 1,0, 1,-1, 0,-1, -1,-1,                [1940]
 -1,0, -1,1
360 '                                                     [117]
370 ' +++  Bildschirm  +++                                [661]
380 '                                                     [117]
390 GOSUB 1040:GOSUB 1390                                 [1216]
400 PRINT CHR$(23);CHR$(1);:GOSUB 1240                    [2259]
410 '                                                     [117]
420 ' +---------+                                         [784]
430 ' :  Spiel  :                                         [237]
440 ' +---------+                                         [784]
450 '                                                     [117]
460 EVERY 50 GOSUB 2380                                   [1111]
470 FOR spieler=1 TO 2                                    [1344]
480   INK 10+spieler,2,0                                  [1626]
490   cx=(spieler-1)*7+1:cy=9-cx:GOSUB 162                [2291]
0
500   '                                                   [117]
510   ' +++  Tastatur  +++                                [1260]
520   '                                                   [117]
530   WHILE INKEY(t(4,spieler))>-1:WEND                   [2314]
540   x=0:y=0                                             [899]
550   IF(INKEY(t(0,spieler))>-1)AND cy<8 T                [3098]
HEN y=1
560   IF(INKEY(t(1,spieler))>-1)AND cx<9 T                [2869]
HEN x=1
570   IF(INKEY(t(2,spieler))>-1)AND cy>1 T                [2403]
HEN y=-1
580   IF(INKEY(t(3,spieler))>-1)AND cx>0 T                [1692]
HEN x=-1
590   IF INKEY(51)>-1 THEN GOSUB 2460                     [787]
600   IF(cx=9 OR cx=0)AND(y AND x=0)THEN y                [1343]
=0
610   IF(cx=8 AND x=1)THEN IF((cy+y)MOD 2)                [2929]
=0 THEN x=0
620   IF(cx=1 AND x=-1)THEN IF((cy+y)MOD 2                [2886]
)=1 THEN x=0
630   IF(x OR y)<>0 THEN cx=cx+x:cy=cy+y:G                [2167]
OSUB 1710
640   IF INKEY(t(4,spieler))=-1 THEN 540                  [2208]
650   '                                                   [117]
660   ' +++  Zug auf Gueltigkeit testen un                [3500]
d ausfuehren  +++
670   '                                                   [117]
680   a=feld(cx,cy)                                       [1707]
690   IF(a<>spieler)AND(flag1=0)THEN GOSUB                [4536]
 1910:GOTO 530
700   IF(flag1=1)AND(cx=mx)AND(cy=my)THEN                 [4296]
flag1=0:GOSUB 1660:GOSUB 1920:GOTO 530
710   IF flag2=0 THEN 730                                 [852]
720   IF(cx=mx)AND(cy=my)THEN sprung=0:GOS                [4688]
UB 1660:GOSUB 1920:GOTO 950
730   IF flag1>0 THEN IF a<>0 THEN GOSUB 1                [3470]
910:GOTO 530 ELSE 750
740   IF flag2=0 THEN GOSUB 1920:GOSUB 166                [3469]
0:flag1=1:mx=cx:my=cy:GOTO 530
750   GOSUB 1800                                          [883]
760   IF fehler THEN GOSUB 1910:GOTO 530                  [2611]
770   CALL &A029,mx*12+4,my*24-1,adr(spiel                [4014]
er):feld(mx,my)=0
780   CALL &A029,cx*12+4,cy*24-1,adr(spiel                [4000]
er):feld(cx,cy)=spieler
790   IF cx>0 AND cx<9 THEN GOSUB 1920:GOT                [3142]
O 810
800   GOSUB 1930:sieg(cx/9)=sieg(cx/9)+1:s                [4480]
prung=0:flag2=0:feld(cx,cy)=3
810   FOR g=0 TO 1                                        [666]
820     IF sieg(g)=4 THEN 2660                            [1731]
830   NEXT g                                              [377]
840   ax=mx:ay=my:GOSUB 1660                              [1387]
850   IF sprung THEN flag1=2:flag2=1:mx=cx                [4617]
:my=cy:ax=cx:ay=cy:GOSUB 1660
860   FOR g=0 TO 7                                        [668]
870     x=cx+rx(g):y=cy+ry(g)                             [1596]
880     IF x<1 OR x>8 OR y<1 OR y>8 THEN 9                [1436]
20
890     IF NOT(feld(x,y)=(3-spieler))THEN                 [1414]
920
```

Listing "H'CARDIN"

```
900     IF NOT(feld(x+rx(g),y+ry(g))=spiel                [2971]
er)THEN 920
910     sprung=0:feld(x,y)=0:CALL &A029,x*                [5078]
12+4,y*24-1,adr(3-spieler)
920   NEXT g                                              [377]
930   IF sprung THEN 530                                  [1163]
940   IF flag2 THEN GOSUB 1620                           [993]
950   GOSUB 1620:flag1=0:flag2=0:INK 10+sp                [2315]
ieler,0:GOSUB 2460
960 NEXT spieler:GOTO 470                                 [644]
970 '                                                     [117]
980 ' +-----------------------------+                     [1696]
990 ' :  Bildschirmaufbau (Feld)  :                       [1450]
1000 ' +-----------------------------+                    [1696]
1010 '                                                    [117]
1020 ' +++  Feld zeichnen  +++                            [1020]
1030 '                                                    [117]
1040 ORIGIN 0,0,56,438,390,8:CLG 8:f=4                    [1760]
1050 FOR g=1 TO 8                                         [698]
1060   FOR h=1 TO 8                                       [793]
1070     IF((g XOR h)AND 1)=0 THEN 1090                   [1526]
1080     GOSUB 1180                                       [911]
1090   NEXT h                                             [372]
1100 NEXT g                                               [377]
1110 CALL &BBBA:f=15                                      [633]
1120 FOR h=0 TO 9 STEP 9                                  [676]
1130   FOR g=1 TO 7 STEP 2                                [587]
1140     g=g-(h=9):GOSUB 1180:g=g+(h=9)                   [2063]
1150   NEXT g                                             [377]
1160 NEXT h                                               [372]
1170 RETURN                                               [555]
1180 y=390-(g-1)*48:x=h*48+8                              [1297]
1190 FOR y=y TO(y-46)STEP-2:PLOT x,y,f:DRA                [3154]
WR 44,0:NEXT y
1200 RETURN                                               [555]
1210 '                                                    [117]
1220 ' +++  Steine setzen  +++                            [1781]
1230 '                                                    [117]
1240 FOR g=1 TO 8                                         [698]
1250   feld(1,g)=1:feld(8,g)=2                            [2101]
1260   a=1-(((g-1)MOD 7)<>0):feld(a+1,g)=1                [3179]
:feld(8-a,g)=2
1270 NEXT g                                               [377]
1280 FOR y=1 TO 8                                         [1182]
1290   FOR x=1 TO 8                                       [706]
1300     CALL &A029,4+x*12,y*24-1,adr(feld                [2454]
(x,y))
1310   NEXT x                                             [356]
1320 NEXT y                                               [359]
1330 RETURN                                               [555]
1340 '                                                    [117]
1350 ' +-------------------------------+                  [1748]
1360 ' :  Bildschirmaufbau (Fenster)  :                   [1529]
1370 ' +-------------------------------+                  [1748]
1380 '                                                    [117]
1390 ORIGIN 0,0,496,640,400,0:CLG 2:CALL &                [2442]
BBBA
1400 h=10:b=15:adr=&A34D:zx=129:zy=199:GOS                [3153]
UB 1540
1410 h=11:b=16:adr=&A17F:zy=170:GOSUB 1540                [2114]
1420 adr=&A22F:zy=100:GOSUB 1540                          [1798]
1430 h=15:b=11:adr=&A0DA:zx=138:zy=14:GOSU                [1823]
B 1540
1440 h=5:b=2:adr=&A2DF                                    [1416]
1450 FOR zy=80 TO 150 STEP 70:zx=136                      [1595]
1460   FOR g=1 TO 4                                       [718]
1470     GOSUB 1540:zx=zx+4                               [1348]
1480     IF g=2 THEN zx=zx+2                              [628]
1490   NEXT g                                             [377]
1500 NEXT zy                                              [463]
1510 MOVE 506,374:DRAW 638,374,3                          [1352]
1520 MOVE 522,370:DRAW 622,370                            [940]
1530 POKE &A001,16:POKE &A003,4:RETURN                    [1965]
1540 POKE &A001,h:POKE &A003,b:CALL &A029,                [2299]
zx,zy,adr:RETURN
1550 '                                                    [117]
1560 ' +----------+                                       [933]
1570 ' :  Cursor  :                                       [403]
1580 ' +----------+                                       [933]
1590 '                                                    [117]
1600 ' +++  Init  +++                                     [455]
1610 '                                                    [117]
1620 ax=cx:ay=cy                                          [762]
1630 '                                                    [117]
```

Listing "H'CARDIN"


```
1640 ' +++  Cursor zeichnen  +++                          [2411]
1650 '                                                    [117]
1660 MOVE 8+ax*48,6+ay*48:DRAWR 44,0,spiel                [4193]
er:DRAWR 0,-46:DRAWR-44,0
1670 DRAWR 0,46:RETURN                                    [392]
1680 '                                                    [117]
1690 ' +++  Bewegen  +++                                  [1059]
1700 '                                                    [117]
1710 FOR g=1 TO 12                                        [793]
1720   GOSUB 1660:ax=ax+x/12:ay=ay+y/12:GO                [2051]
SUB 1660
1730 NEXT g                                               [377]
1740 RETURN                                               [555]
1750 '                                                    [117]
1760 ' +--------------------+                             [1102]
1770 ' :  Sprung moeglich  :                              [1124]
1780 ' +--------------------+                             [1102]
1790 '                                                    [117]
1800 fehler=0:sprung=0                                    [1159]
1810 x=ax-mx:y=ay-my:xx=ABS(x):yy=ABS(y):a                [4416]
=xx+yy
1820 IF(xx OR yy)<2 THEN IF flag2 THEN feh                [3929]
ler=1:RETURN ELSE RETURN
1830 IF a<>4 AND a<>2 THEN fehler=1:RETURN                [2906]
1840 IF feld(mx+x/2,my+y/2)=0 THEN fehler=                [2724]
1:RETURN
1850 sprung=1:RETURN                                      [1553]
1860 '                                                    [117]
1870 '  +---------+                                       [784]
1880 '  :  SOUND  :                                       [253]
1890 '  +---------+                                       [784]
1900 '                                                    [117]
1910 SOUND 7,1000,15,13:WHILE SQ(1)>10:WEN                [2579]
D:RETURN
1920 SOUND 7,50,-1,15,3:WHILE SQ(1)>10:WEN                [2409]
D:RETURN
1930 SOUND 7,119,12,15,1:SOUND 7,89,12,15,                [3213]
1:SOUND 7,71,12,15,1
1940 SOUND 7,60,24,15,3:SOUND 7,71,12,15,3                [4466]
:SOUND 7,60,48,15,1
1950 WHILE SQ(1)>10:WEND:RETURN                           [1619]
1960 '                                                    [117]
1970 ' +++  Daten fuer Schlussmelodie  +++                [1518]
1980 '                                                    [117]
1990 DATA 426,568,851,568,426,568,426,568,                [2980]
426,568,379,568,358,379,426,379
2000 DATA 426,568,716,758,851,568,426,568                 [1361]
2010 DATA 213,0,179,213,142,213,106,179,89                [2378]
,142,95,159,119,190
2020 DATA 1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,3,                [3061]
4,5,6,6,3,4,3,4,3,4,3,4,3,4,4,5,6
2030 DATA 1,6,1,5,6,1,3,4,4,3,4,1,1,6,1,6,                [3366]
6,6,6,6,6,6,6,6,6,6,2,2,5
2040 DATA 0,16,1,8,2,6,3,2,4,12,5,4,6,4,4,                [3395]
4,4,4,7,4,4,16,2,8,3,6,4,2,6,12,5,2,4
2050 DATA 2,7,1,4,1,7,2,4,1,7,1,4,8,7,2,4,                [3126]
16,5,6,5,2,5,4,6,2,4,2,6,6,6,2,6,4,4
2060 DATA 2,7,2,3,16,1,8,2,6,3,2,4,12,5,4,                [1768]
6,4,4,4,4,8,7,8,4,16
2070 '                                                    [117]
2080 ' +++  Schlussmelodie  +++                           [1930]
2090 '                                                    [117]
2100 flag=0:RESTORE 1990                                  [1135]
2110 FOR g=1 TO 6                                         [724]
2120   FOR h=1 TO 4                                       [813]
2130     READ us(g,h)                                     [794]
2140   NEXT h                                             [372]
2150 NEXT g                                               [377]
2160 FOR g=1 TO 7:READ hs1(g),hs2(g):NEXT                 [2542]
g
2170 FOR g=1 TO 65:READ ton(g):NEXT g                     [1865]
2180 us=1:s=20:SOUND 7,0,20,0:GOSUB 2240                  [2350]
2190 FOR t=1 TO 46:READ a,l                               [1470]
2200   SOUND 1,hs1(a),s*l,15,1                            [1876]
2210   SOUND 4,hs2(a),s*l,15,1                            [1094]
2220 NEXT t                                               [360]
2230 IF flag=1 THEN 2290 ELSE 2230                        [1201]
2240 FOR g=1 TO 4                                         [718]
2250   SOUND 2,us(ton(us),g),s,15,2                       [3258]
2260 NEXT g:AFTER s GOSUB 2240                            [813]
2270 us=us+1:IF us>65 THEN us=0:flag=1:SOU                [3349]
ND 2,851,80,15,2
2280 RETURN                                               [555]
2290 a=REMAIN(0)                                          [1296]
2300 WHILE SQ(1)>10:WEND:RETURN                           [1619]
2310 '                                                    [117]
```

Listing "H'CARDIN"

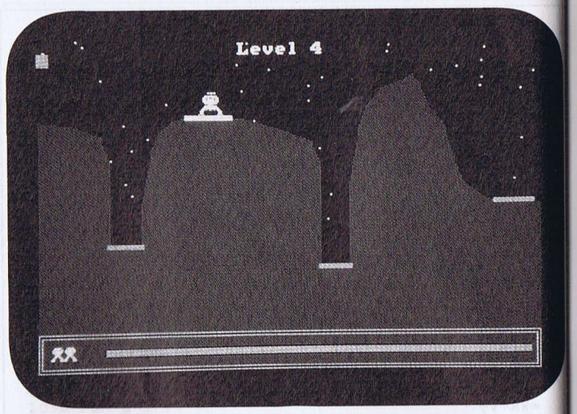
```
2320 ' +-------+                                          [621]
2330 ' :  Uhr  :                                          [372]
2340 ' +-------+                                          [621]
2350 '                                                    [117]
2360 ' +++  zaehlen  +++                                  [1051]
2370 '                                                    [117]
2380 DI:m=spieler:z(1,m)=z(1,m)+1:IF z(1,m                [3521]
)<10 THEN 2420
2390 z(1,m)=0:z(2,m)=z(2,m)+1:IF z(2,m)<6                 [3012]
THEN 2420
2400 z(2,m)=0:z(3,m)=z(3,m)+1:IF z(3,m)<10                [2640]
 THEN 2420
2410 z(3,m)=0:z(4,m)=z(4,m)+1:IF z(4,m)=6                 [2661]
THEN z(4,m)=0
2420 EI:RETURN                                            [416]
2430 '                                                    [117]
2440 ' +++  zeichnen  +++                                 [963]
2450 '                                                    [117]
2460 DI:m=h:h=5:b=2:zy=220-spieler*70                      [2348]
2470 FOR t=1 TO 4                                         [583]
2480   ab(t)=z(t,spieler+2)-z(t,spieler)                  [2390]
2490   IF ab(t)=0 THEN 2510                               [989]
2500   IF(ab(t)>4)OR(ab(t)<0 AND ab(t)>-5)                [3493]
THEN ab(t)=0.2 ELSE ab(t)=-0.2
2510 NEXT t                                               [360]
2520 v=0:zx=150                                           [1498]
2530 FOR t=1 TO 4                                         [583]
2540   IF z(t,spieler+2)=z(t,spieler)THEN                 [2825]
2610 ELSE v=1
2550   altadr=&A2DF+z(t,spieler+2)*10                     [2010]
2560   z(t,spieler+2)=ROUND(z(t,spieler+2)                [3573]
+ab(t),1)
2570   IF z(t,spieler+2)=10 THEN z(t,spiel                [2435]
er+2)=0:GOTO 2590
2580   IF z(t,spieler+2)<0 THEN z(t,spiele                [2713]
r+2)=9.8
2590   adr=altadr:GOSUB 1540                              [645]
2600   adr=&A2DF+z(t,spieler+2)*10:GOSUB 1                [3055]
540
2610   zx=zx-4:IF t=2 THEN zx=zx-2                        [1977]
2620 NEXT t                                               [360]
2630 IF v=1 THEN 2520                                     [937]
2640 GOSUB 1530:h=m:EI:RETURN                             [2288]
2650 CALL &BD19:RETURN                                    [449]
2660 '                                                    [117]
2670 ' +------------+                                     [1005]
2680 ' :  Gewonnen  :                                     [293]
2690 ' +------------+                                     [1005]
2700 '                                                    [117]
2710 GOSUB 2460:PAPER 2:SOUND 2,70,-10,0,4                [3084]
,1
2720 OUT &BC00,7:g=g XOR 1                                [1563]
2730 FOR t=30 TO 25 STEP-1:OUT &BD00,t:GOS                [2281]
UB 2650:NEXT t
2740 FOR t=24 TO 0 STEP-1                                 [1257]
2750   OUT &BD00,t:GOSUB 2650:WINDOW 1,20,                [2343]
t+1,t+1:CLS
2760 NEXT t                                               [360]
2770 FOR t=38 TO 31 STEP-1:OUT &BD00,t:GOS                [4909]
UB 2650:NEXT t
2780 INK 0,3:BORDER 3:PAPER 0:CLS:MODE 0                  [1320]
2790 a=REMAIN(0)                                          [1296]
2800 PEN 8:LOCATE 1,25:PRINT USING"#";g+1;                [1836]
2810 PEN 7:LOCATE 1,1:PRINT"SPIELER"                      [3084]
2820 FOR x=0 TO 28 STEP 4                                 [1204]
2830   FOR y=0 TO 16 STEP 2                               [1367]
2840     a=((TEST(x,y)=0)+1)*4:MOVE x*2+25                [2311]
6,y*4+334
2850     DRAWR 0,6,a:MOVER 4,0:DRAWR 0,-6                 [1771]
2860   NEXT y                                             [359]
2870 NEXT x                                               [356]
2880 LOCATE 12,3:PRINT"HAT":LOCATE 12,4:PR                [2605]
INT"GEWONNEN!"
2890 LOCATE 1,1:PRINT STRING$(9,11)                       [950]
2900 POKE &A001,16:POKE &A003,4                           [1628]
2910 FOR x=0 TO 150 STEP 10                               [1205]
2920   CALL &A029,x,87,adr(g+1):CALL &A029                [2788]
,152-x,151,adr(g+1)
2930 NEXT x                                               [356]
2940 GOSUB 2100                                           [871]
2950 RUN                                                  [243]
2960 '                                                    [117]
2970 ' +-----------+                                      [828]
2980 ' :  E n d e  :                                      [394]
2990 ' +-----------+                                      [828]
3000 '                                                    [117]
```

Listing "H'CARDIN"

MISSION WEGA

Ein Spiel nicht nur für Helden

Die Menschheit hat es geschafft, das 21. Jahrhundert zu erreichen, ohne sich selbst zu vernichten. Mit den Problemen ist man recht gut fertig geworden und es gibt eigentlich nur noch eine Gefahr, die die Menschheit bedroht - die Überbevölkerung. Um auch dieser Herr zu werden, begann man das All zu erforschen und zu besiedeln. Zuerst nur das eigene Sonnensystem, dann aber auch andere, weit entfernte Systeme.



Vielfältige Schwierigkeiten sind bei Mission Wega zu meistern.

Genauso ist es bei Wega 13 geschehen. Man hatte zuerst Vorposten errichtet und wollte dann nach und nach den Planeten besiedeln, zumindest aber die Bodenschätze ausbeuten. Dazu kam es aber nicht, denn dieser Planet ist vulkanisch noch sehr aktiv, und als man entdeckte, daß sich demnächst eine Reihe heftigster Ausbrüche ereignen würde, riefen die Forscher, die diesen Planeten auf eventuelle Bodenschätze untersuchen sollten, das nächstbeste Schiff um Hilfe an, denn die Zeit, die ein Bergungsschiff benötigen würde, den Planeten zu erreichen, würde nicht ausreichen. Natürlich übernehmen wieder einmal Sie die Rolle des hilfreichen Retters in der Not. Sie sind der einzige an Bord des Schiffes, der mit den ferngesteuerten Landekapseln so gut umgehen kann, daß noch Hoffnung auf Rettung der Wissenschaftler besteht. Mit Hilfe eines Radarschirmes scheint Ihre Aufgabe auch gar nicht mal so schwer, aber die Forscher haben sich die abgelegendsten Winkel ausgesucht, ihre Stationen zu errichten.

Auf dem Bildschirm stellt sich das Ganze dann so dar: Im unteren Teil können Sie erkennen, wieviel Treibstoff Sie noch zur Verfügung haben und wie viele Landekapseln noch in den Hangarn Ihres Schiffes stehen. Wenn Sie auf der Plattform einer Station gelandet sind (vorsichtig landen, die Kapseln sind nicht allzu stabil) erhalten Sie aus den Tanks der Station wieder einen vollen Tank. Die Kapsel wird mit dem Joystick oder den Pfeiltasten gesteuert. Ein Druck auf <P> unterbricht das Spiel, und das Programm, wartet auf einen Tastendruck. Wenn Sie alle Stationen angeflogen haben, so verlassen Sie den Level durch den rechten Bildschirmrand. Dieser birgt eine kleine Überraschung - genauso wie der obere Rand, wenn sie ihn vor dem Absolvieren aller Stationen zu verlassen versuchen. Durch den linken Bildschirmrand ist es Ihnen möglich, noch einmal in den vorhergehenden Level zurückzukehren. Sollten Sie Level 16 erreichen, ohne vorher eine Kapsel zu verlieren, so verbraucht Ihre Kapsel noch halb soviel Treibstoff.

Die Programme sind als "WEGA.BAS" (Listing 1), "WEGA.MC" und als "WEGA.PRG" (Listing 3) hintereinander abzuspeichern. Die Datei "WEGA.MC" wird durch Listing 2 erzeugt. Die Programme unbedingt vor einem Probelauf abspeichern! Ein kleiner Tippfehler reicht oft aus, die ganze Arbeit zu zerstören.

(Steffen Rau/sr)

für 464-664-6128

```
1 '                                                    [117]
2 '  L I S T I N G   1                                 [1024]
3 '                                                    [117]
10 ' *================================*                [175]
20 ' I                                I                [90]
30 ' I    M I S S I O N    W E G A    I                [2178]
40 ' I         (Ladeprogramm)         I                [1335]
50 ' I                                I                [90]
60 ' *================================*                [175]
70 '                                                   [117]
80 '   Dieses Programm wurde im Maerz/Apri             [3836]
l 1988 von
90 '                                                   [117]
100 '  S t e f f e n  R a u , Obrigheim                [983]
110 ' ======================                           [1579]
120 '                                                  [117]
130 '  fuer den Schneider/Amstrad CPC gesc             [1810]
hrieben.
140 '                                                  [117]
150 '  (Ich danke Jens Koeplinger fuer sei             [4617]
ne freundliche Unterstuetzung
160 '  und seine durchaus hilfreichen Anre             [3045]
gegungen.)
170 '                                                  [117]
180 ' +-----------------+                             [1245]
190 ' :   Titelgrafik   :                              [1061]
200 ' +-----------------+                             [1245]
```

Listing "Mission Wega"

```
210 '                                                  [117]
220 SYMBOL AFTER 256:MEMORY &9FFF:SYMBOL A             [3798]
FTER 32
230 MODE 1:INK 0,0:INK 1,6:INK 2,0:INK 3,2             [2512]
4:BORDER 0
240 a$="Mission":PLOT-10,10,3:MOVE 100,300             [4880]
:GOSUB 660
250 PRINT CHR$(23);CHR$(3);:PLOT-10,10,1:M             [4677]
OVE 106,296:GOSUB 660
260 LOCATE 1,25:PEN 2:PRINT"Wega"                      [2100]
270 FOR f=3 TO 1 STEP-2                                [1466]
280   FOR x=0 TO 62 STEP 2                            [962]
290     FOR y=0 TO 14 STEP 2                           [1487]
300       a=TEST(x,y):IF a=0 THEN 340                  [2430]
310       MOVE 250-f*5+y+x*4,250+y*4+f*3:D             [2565]
RAWR 2,6,f
320       MOVER 0,-6:DRAWR 2,6:MOVER 0,-6:             [1083]
DRAWR 2,6
330       MOVER 0,-6:DRAWR 2,6                         [569]
340     NEXT y                                         [359]
350   NEXT x                                           [356]
360 NEXT f                                             [374]
370 LOCATE 1,25:PRINT"    ":INK 2,13                   [1747]
380 LOCATE 11,15:PRINT"geschrieben 1988 vo             [3145]
n"
390 a$="Steffen Rau":PLOT-10,10,3:MOVE 230             [4123]
,100:GOSUB 660
400 PLOT-10,10,1:MOVE 236,96:GOSUB 660                 [1861]
410 PRINT CHR$(23);CHR$(0);                            [2098]
420 '                                                  [117]
430 ' +-----------------+                             [1245]
440 ' :   Zeichensatz   :                              [843]
450 ' +-----------------+                             [1245]
```

Listing "Mission Wega"


```
460 '                                                  [117]
470 SYMBOL 243,8,8,24,16,32                            [1213]
480 SYMBOL 244,0,0,64,48,28,3                          [1115]
490 SYMBOL 245,0,140,2,33,48,16,8                      [1783]
500 SYMBOL 246,0,3,12,16,32,32,64,64                   [1752]
510 SYMBOL 247,17,17,35,3,134,28,112                   [2092]
520 SYMBOL 248,16,128,140,199,96,112,60,15             [1727]
530 SYMBOL 249,12,2,1,25,68,98,32                      [1872]
540 SYMBOL 250,15,60,112,99,100,200,136,14             [2215]
4
550 SYMBOL 251,226,198,142,28,112,0,192                [2273]
560 SYMBOL 252,70,39,51,48,24,14,3                     [1533]
570 SYMBOL 253,0,136,6,2,50,8,192,96                   [1779]
580 SYMBOL 254,0,7,30,56,48,35,39,7                    [1752]
590 SYMBOL 255,60,90,126,60,24,36,66,195               [2507]
600 '                                                  [117]
610 ' +-------------------+                            [1102]
620 ' :   Proramm laden   :                            [1634]
630 ' +-------------------+                            [1102]
640 '                                                  [117]
650 LOAD"!wega.mc",&A000:RUN"!wega.prg"                [3689]
660 '                                                  [117]
670 ' +-------------------+                            [1102]
680 ' :   Doppelschrift   :                            [1654]
690 ' +-------------------+                            [1102]
700 '                                                  [117]
710 TAG:FOR g=1 TO LEN(a$)                             [353]
720   FOR h=0 TO 7                                     [638]
730     p(h)=PEEK(HIMEM+1+(ASC(MID$(a$,g,1             [2733]
))-32)*8+h)
740   NEXT h                                           [372]
750   SYMBOL 254,p(0),p(0),p(1),p(1),p(2),             [2983]
p(2),p(3),p(3)
760   SYMBOL 255,p(4),p(4),p(5),p(5),p(6),             [2841]
p(6),p(7),p(7)
770   PRINT CHR$(254);:MOVER-16,-16:PRINT              [3979]
CHR$(255);:MOVER 0,16
780 NEXT g:TAGOFF:RETURN                               [1492]
```

```
1 '                                                    [117]
2 '  L I S T I N G   2                                 [1019]
3 '                                                    [117]
10 ' *================================*                [175]
20 ' I                                I                [90]
30 ' I    M I S S I O N    W E G A    I                [2178]
40 ' I    (MCode und Grafik poken)    I                [945]
50 ' I                                I                [90]
60 ' *================================*                [175]
70 '                                                   [117]
80 ' +---------------------------------+               [1843]
90 ' :   MCode poken und abspeichern    :              [3175]
100 '+---------------------------------+               [1843]
110 '                                                  [117]
120 MEMORY &9FFF:adr=&A000                             [955]
130 FOR z=25 TO 50:sum=0                               [1312]
140   FOR g=1 TO 16                                   [805]
150     READ a$:a=VAL("&"+a$):sum=sum+a:PO             [2586]
KE adr,a:adr=adr+1
160   NEXT g                                           [377]
170   READ s:IF s<>sum THEN PRINT CHR$(7);             [5213]
"Fehler in Zeile";z*10;"!!!":END
180 NEXT z                                             [354]
190 SAVE"wega.mc",b,&A000,&1A0                         [752]
200 '                                                  [117]
210 ' +-----------------------+                        [1503]
220 ' :   Daten fuer MCode    :                        [2180]
230 ' +-----------------------+                        [1503]
240 '                                                  [117]
250 DATA 88,CD,9E,A0,0E,05,E5,7E,12,23,13,             [2461]
0D,C2,07,A0,E1, 1704
260 DATA CD,AE,A0,10,EF,C9,CD,9E,A0,0E,05,             [3104]
E5,1A,77,23,13, 1965
270 DATA 0D,C2,1C,A0,E1,CD,AE,A0,10,EF,C9,             [2642]
0E,10,11,BC,A0, 2010
280 DATA C5,D5,2A,9E,A1,ED,5B,9C,A1,CD,1D,             [2045]
BC,79,32,00,A0, 2169
290 DATA D1,C1,22,9C,A1,06,0E,C5,3A,00,A0,             [2913]
4F,1A,FE,00,CA, 1749
300 DATA 56,A0,AE,A1,AE,77,C1,CD,AE,A0,13,             [2290]
10,EA,3A,00,A0, 2093
310 DATA 2A,9C,A1,0F,D2,6B,A0,23,22,9C,A1,             [3893]
32,00,A0,0D,C2, 1654
```

Listing "Mission Wega"

```
320 DATA 45,A0,C9,DD,66,01,DD,6E,00,DD,56,             [1502]
03,DD,5E,02,ED, 1949
330 DATA 53,9C,A1,22,9E,A1,ED,53,A0,A1,22,             [2847]
A2,A1,C9,DD,E5, 2402
340 DATA CD,16,A0,DD,E1,CD,73,A0,CD,01,A0,             [2780]
C3,2B,A0,2A,A2, 2281
350 DATA A1,ED,5B,A0,A1,CD,1D,BC,11,A5,A1,             [3322]
06,0E,C9,D5,11, 2026
360 DATA 00,08,19,D2,BA,A0,11,50,C0,19,D1,             [3307]
C9,00,00,00,00, 1313
370 DATA 00,00,00,00,00,00,00,00,00,FF,00,             [2175]
00,00,00,00,00, 255
380 DATA 00,00,00,00,00,F0,F0,FF,00,00,00,             [2184]
00,00,00,00,00, 735
390 DATA 00,00,F0,F0,F0,FF,00,00,00,F0,F0,             [2756]
F0,00,00,00,F0, 1935
400 DATA F0,00,00,FF,00,00,F0,F0,F0,F0,F0,             [2813]
00,F0,F0,00,00, 2175
410 DATA 00,00,00,00,F0,00,00,F0,F0,00,F0,             [1876]
F0,FF,00,00,00, 1455
420 DATA F0,00,F0,F0,F0,F0,F0,FF,F0,F0,00,             [2840]
00,00,00,FF,00, 2430
430 DATA F0,00,F0,F0,F0,FF,F0,F0,FF,00,00,             [2489]
00,F0,FF,F0,00, 2685
440 DATA F0,F0,F0,FF,F0,F0,FF,00,00,00,FF,             [4257]
00,F0,F0,F0,F0, 2925
450 DATA F0,FF,F0,F0,00,00,00,00,F0,00,F0,             [3409]
00,00,F0,F0,00, 1935
460 DATA F0,F0,FF,00,00,00,00,00,F0,F0,F0,             [4249]
F0,F0,00,F0,F0, 2415
470 DATA 00,00,00,00,00,00,00,F0,F0,F0,00,             [2151]
00,00,F0,F0,00, 1200
480 DATA 00,FF,00,00,00,00,00,00,00,00,00,             [2386]
00,F0,F0,F0,FF, 1230
490 DATA 00,00,00,00,00,00,00,00,00,00,00,             [1841]
F0,F0,FF,00,00, 735
500 DATA 00,00,00,00,00,00,00,00,00,00,00,             [1679]
FF,00,00,00,00, 255
```

```
1 '                                                    [117]
2 '  L I S T I N G   3                                 [1022]
3 '                                                    [117]
10 ' *================================*                [175]
20 ' I                                I                [90]
30 ' I    M I S S I O N    W E G A    I                [2178]
40 ' I        (Hauptprogramm)         I                [2201]
50 ' I                                I                [90]
60 ' *================================*                [175]
70 '                                                   [117]
80 ' +---------------------------------- [2081]
-----+
90 ' :   Initialisierung und Bildschirmauf [2840]
bau  :
100 '+---------------------------------- [2081]
-----+
110 '                                                  [117]
120 WHILE INKEY$<>"":WEND                              [1786]
130 FOR h=1 TO 0 STEP -1                               [1756]
140   PEN h:LOCATE 40,25:PRINT">";                     [1680]
150   FOR g=1 TO 200                                   [850]
160     IF INKEY$<>""THEN 200                          [763]
170   NEXT g                                           [377]
180 NEXT h                                             [372]
190 GOTO 130                                           [371]
200 DIM t(3),sx(10),sy(10),sb(10),o$(3),u$             [1042]
(3)
210 ENV 1,15,-1,10:ENV 2,5,-1,12:ENV 3,15,             [1844]
-1,20
220 FOR g=1 TO 3:a=g*4                                 [1254]
230   o$(g)=CHR$(258-a)+CHR$(257-a)                    [1792]
240   u$(g)=CHR$(256-a)+CHR$(255-a)                    [1672]
250 NEXT g                                             [377]
260 MODE 1:INK 0,0:INK 1,26:INK 2,3:INK 3,             [1718]
13:BORDER 0
270 PRINT CHR$(23);CHR$(0);                            [2098]
280 WINDOW#2,1,40,1,22:l=1:r=40:o=23:u=25:             [3482]
GOSUB 1710
290 PLOT 86,28,0:DRAWR 540,0:PLOTR 2,-2:DR             [2954]
AWR 0,-6
300 PLOTR-2,-2:DRAWR-540,0:PLOTR-2,2:DRAWR             [2932]
 0,6:PLOTR 2,0,3
310 FOR g=1 TO 4:DRAWR 540,0:MOVER-540,-2:             [2651]
NEXT g
```

Listing "Mission Wega"

```
320 PAPER 2:PEN 1:LOCATE 2,24:PRINT STRING [3444]
$(3,255)
330 l=7:r=34:o=9:u=13:GOSUB 1710           [2087]
340 LOCATE 9,10:PRINT"Joystick / Pfeiltast [2765]
en ?"
350 LOCATE 16,12:PRINT"(J / P): ";:PEN 3:P [2674]
RINT"_";CHR$(8);:PEN 1
360 WHILE INKEY$<>"":WEND:a$=""            [2110]
370 WHILE a$<>"J"AND a$<>"P":a$=UPPER$(INK [3776]
EY$):WEND:PRINT a$
380 IF a$="P"THEN RESTORE 410 ELSE RESTORE [4189]
 420
390 FOR g=1 TO 3:READ t(g):NEXT            [2022]
400 FOR g=1 TO 500:NEXT g:PAPER#1,0:CLS#1  [2552]
410 DATA 8,0,1                             [216]
420 DATA 74,72,75                          [331]
430 '                                      [117]
440 ' +-----------+                        [828]
450 ' :   Spiel   :                        [237]
460 ' +-----------+                        [828]
470 '                                      [117]
480 tmin=2:leben=3:level=1                 [2093]
490 landung=0:GOSUB 1490:GOSUB 1840        [2380]
500 IF level=16 AND leben=3 THEN tmin=1    [1737]
510 tank=540:xp=25:yp=195:rx=0:ry=0:CALL & [2980]
A095,xp,yp
520 '                                      [117]
530 ' +++  Tastatur/Joystick  +++          [2315]
540 '                                      [117]
550 v=0:IF INKEY(t(1))>-1 THEN rx=rx-0.2:v [4224]
=1:SOUND 129,0,0,5,2,,10
560 IF INKEY(t(3))>-1 THEN rx=rx+0.2:v=1:S [2515]
OUND 132,0,0,5,2,,10
570 IF INKEY(t(2))>-1 THEN ry=ry+0.2:v=1:S [3525]
OUND 130,0,0,5,2,,10
580 IF INKEY(27)=-1 THEN 610               [1088]
590 BORDER 0,2:WHILE INKEY$<>"":WEND       [2234]
600 WHILE INKEY$="":WEND:BORDER 0          [2642]
610 ry=ry-0.1:IF rx<>0 THEN rx=rx-0.05*SGN [2243]
(rx)
620 IF ABS(rx)<0.001 THEN rx=0             [1238]
630 xp=xp+rx:yp=yp+ry                      [1865]
640 IF yp>199 THEN yp=199:ry=-ry           [2339]
650 IF xp>304 AND landung=anzahl THEN 1250 [901]
660 IF xp>304 THEN xp=304:rx=-rx           [1668]
670 IF xp>-0.5 THEN 720                    [1116]
680 IF level=1 THEN xp=0:rx=0:GOTO 720     [1950]
690 level=level-1:PLOT 86,26,3             [1667]
700 FOR g=1 TO 4:DRAWR 540,0:MOVER-540,-2: [2651]
NEXT g
710 GOTO 490                               [371]
720 CALL &A08E,xp,yp:CALL &BB19            [2322]
730 '                                      [117]
740 ' +++  Test  +++                       [733]
750 '                                      [117]
760 x=ROUND(xp)*2:y=ROUND(yp)*2            [4111]
770 a1=TEST(x,y-28):a2=TEST(x+30,y-28)     [1944]
780 IF a1=3 AND a2=3 THEN IF ry>-1.1 THEN  [3448]
1120 ELSE 890
790 IF(a1=3 AND a2<>3)OR(a1<>3 AND a2=3)TH [2272]
EN 890
800 IF a1=2 OR a2=2 THEN 890               [930]
810 IF TEST(x+4,y-2)=2 OR TEST(x+26,y-2)=2 [1363]
 THEN 890
820 IF tank=0 THEN 610                     [1066]
830 IF v=0 THEN 550                        [954]
840 PLOT 86+tank,26,0:DRAWR 0,-6:tank=tank [2503]
-tmin
850 GOTO 550                               [431]
860 '                                      [117]
870 ' +++  abgestuerzt  +++                [844]
880 '                                      [117]
890 CALL &A016:SOUND 135,0,0,15,3,,31      [1578]
900 PLOT-10,10,3:PRINT CHR$(23)CHR$(1);:TA [2183]
G
910 FOR g=1 TO 3                           [711]
920   MOVE x,y:PRINT o$(g);:MOVE x,y-16:PR [2920]
INT u$(g);
930   FOR t=1 TO 150:NEXT t                [939]
940   MOVE x,y:PRINT o$(g);:MOVE x,y-16:PR [2920]
INT u$(g);
950 NEXT g:TAGOFF:PRINT CHR$(23)CHR$(0);   [2078]
960 leben=leben-1:IF leben=-1 THEN 1030    [1938]
970 LOCATE 2,24:PRINT STRING$(leben,255);S [4101]
PACE$(3-leben)
980 PLOT 86,26,3:FOR g=1 TO 4:DRAWR 540,0: [2983]
MOVER-540,-2:NEXT g
990 WHILE SQ(1)>10:WEND:GOTO 510           [1933]
```

Listing "Mission Wega"

```
1000 '                                     [117]
1010 ' +++  nicht geschafft  +++           [1319]
1020 '                                     [117]
1030 l=1:r=40:o=3:u=9:GOSUB 1710           [2054]
1040 PEN 1:LOCATE 3,5:PRINT"Sie haben es l [4677]
eider nicht geschafft,"
1050 LOCATE 10,7:PRINT"die Forscher zu ret [3433]
ten."
1060 l=10:r=31:o=15:u=19:GOSUB 1710        [1482]
1070 LOCATE 12,16:PRINT"Noch ein Versuch ? [3140]
"
1080 GOTO 1330                             [448]
1090 '                                     [117]
1100 ' +++  gelandet  +++                  [730]
1110 '                                     [117]
1120 landung=landung+1:tank=540:rx=0:ry=1  [2369]
1130 PLOT 86,26,3:FOR g=1 TO 4:DRAWR 540,0 [2983]
:MOVER-540,-2:NEXT g
1140 FOR st=1 TO anzahl                    [1186]
1150 IF sy(st)<>y-28 OR x-sx(st)>sb(st)-30 [1407]
 THEN NEXT st
1160 FOR g=0 TO 2                          [673]
1170   PLOT sx(st),sy(st)-g*2,1:DRAWR sb(s [2264]
t),0
1180 NEXT g                                [377]
1190 SOUND 129,239,0,15,1:SOUND 130,190,0, [3627]
15,1:SOUND 132,159,0,15,1
1200 WHILE SQ(1)>10:WEND                   [1760]
1210 SOUND 2,0,0,5,2,,10:GOTO 550          [1596]
1220 '                                     [117]
1230 ' +++  geschafft  +++                 [1161]
1240 '                                     [117]
1250 level=level+1:IF level=17 THEN 1280   [2797]
1260 PLOT 86,26,3:FOR g=1 TO 4:DRAWR 540,0 [2983]
:MOVER-540,-2:NEXT g
1270 GOTO 490                              [371]
1280 l=2:r=39:o=3:u=9:GOSUB 1710           [1631]
1290 LOCATE 9,5:PEN 1:PRINT"Herzlichen Glu [3414]
eckwunsch !"
1300 LOCATE 4,7:PRINT"Sie haben alle Forsc [5004]
her gerettet !"
1310 l=11:r=30:o=15:u=19:GOSUB 1710        [1495]
1320 LOCATE 13,16:PRINT"Noch ein Spiel ?"  [2234]
1330 LOCATE 16,18:PRINT"(J / N): ";:PEN 3: [3258]
PRINT"_";CHR$(8);:PEN 1
1340 WHILE INKEY$<>"":WEND:a$=""           [2110]
1350 WHILE a$<>"J"AND a$<>"N":a$=UPPER$(IN [4682]
KEY$):WEND:PRINT a$
1360 FOR g=1 TO 500:NEXT g                 [1038]
1370 IF a$="J"THEN LOCATE 2,24:PRINT STRIN [3213]
G$(3,255):GOTO 480
1380 l=7:r=34:o=8:u=16:GOSUB 1710          [1334]
1390 LOCATE 11,12:PRINT"Auf Wiedersehen .. [1862]
."
1400 PEN 0:LOCATE 25,15:PRINT"Taste ..."   [2580]
1410 WHILE INKEY$<>"":WEND:WHILE INKEY$="" [3680]
:WEND
1420 MODE 1:CALL &BC02:CALL &BBBA:PEN 1:PA [2281]
PER 0:END
1430 GOTO 1430                             [341]
1440 '                                     [117]
1450 ' +---------------------------------- [1880]
--+
1460 ' :   Zeile der Landschaft bestimmen  [1647]
  :
1470 ' +---------------------------------- [1880]
--+
1480 '                                     [117]
1490 ON level GOTO 1500,1510,1520,1530,154 [4871]
0,1550,1560,1570,1580,1590,1600,1610,1620,
1630,1640,1650
1500 RESTORE 2130:RETURN                   [1357]
1510 RESTORE 2200:RETURN                   [1986]
1520 RESTORE 2280:RETURN                   [1179]
1530 RESTORE 2350:RETURN                   [1166]
1540 RESTORE 2430:RETURN                   [651]
1550 RESTORE 2520:RETURN                   [1117]
1560 RESTORE 2620:RETURN                   [1359]
1570 RESTORE 2710:RETURN                   [1481]
1580 RESTORE 2810:RETURN                   [1145]
1590 RESTORE 2900:RETURN                   [1300]
1600 RESTORE 3000:RETURN                   [895]
1610 RESTORE 3100:RETURN                   [1051]
1620 RESTORE 3190:RETURN                   [983]
1630 RESTORE 3290:RETURN                   [1642]
1640 RESTORE 3380:RETURN                   [1161]
1650 RESTORE 3480:RETURN                   [2019]
```

Listing "Mission Wega"


```
1660 '                                     [117]
1670 ' +---------------------+             [1395]
1680 ' :   Kasten zeichnen   :             [933]
1690 ' +---------------------+             [1395]
1700 '                                     [117]
1710 WINDOW#1,l,r,o,u:PAPER#1,2:CLS#1      [2145]
1720 l=l*16-16:r=r*16-2:o=(25-o)*16+14:u=( [2147]
25-u)*16
1730 PLOT l,o,3:DRAW r,o:DRAW r,u:DRAW l,u [2115]
:DRAW l,o
1740 PLOT l+4,o-4:DRAW r-4,o-4:DRAW r-4,u+ [3011]
4
1750 DRAW l+4,u+4:DRAW l+4,o-4             [1326]
1760 PLOT l+2,o-2,0:DRAW r-2,o-2:DRAW r-2, [2047]
u+2
1770 DRAW r-2,u+2:DRAW l+2,u+2:DRAW l+2,o- [2528]
2
1780 RETURN                                [555]
1790 '                                     [117]
1800 ' +-------------------------+         [1577]
1810 ' :   Landschaft zeichnen   :         [1748]
1820 ' +-------------------------+         [1577]
1830 '                                     [117]
1840 CLS#2:PEN 1:PAPER 0                   [1424]
1850 LOCATE 17,1:PRINT"Level";level        [1762]
1860 PAPER 2                               [822]
1870 ax=0:READ ay                          [570]
1880 READ sx,sy:IF sx=3.1415 THEN 1940     [2175]
1890 IF sx>0 THEN f=2 ELSE f=0             [1746]
1900 FOR x=ax TO ax+sx STEP 2*SGN(sx)      [2971]
1910   ay=ay+sy/ABS(sx):PLOT x,48,f:DRAW x [1903]
,ay
1920 NEXT x                                [356]
1930 ax=ax+sx:GOTO 1880                    [674]
1940 READ anzahl                           [711]
1950 FOR st=1 TO anzahl                    [1186]
1960   READ sx(st),sy(st),sb(st)           [1944]
1970   FOR g=0 TO 2                        [673]
1980     PLOT sx(st),sy(st)-g*2,3:DRAWR sb [2163]
(st),0
1990   NEXT g                              [377]
2000 NEXT st                               [454]
2010 FOR g=1 TO 100                        [886]
2020   x=INT(RND*640):y=INT(RND*350)+50    [2545]
2030   IF TEST(x,y)=0 THEN PLOT x,y,1      [962]
2040 NEXT g                                [377]
2050 RETURN                                [555]
2060 '                                     [117]
2070 ' +-----------------------------+     [1801]
2080 ' :   Daten fuer Landschaften   :     [2293]
2090 ' +-----------------------------+     [1801]
2100 '                                     [117]
2110 ' +++  1.Bild  +++                    [469]
2120 '                                     [117]
2130 DATA 250, 40,-10, 30,-20, 20,-40, 20, [3551]
-50, 20,-40, 40,-20, 40,-10, 40,0
2140 DATA 40,-10, 20,-20, 10,-20, 30,-10,  [2801]
50,-10, 50,10, 40,20, 30,40, 20,70
2150 DATA 10,150, 20,50, 20,30, 10,10, 10, [2017]
-10, 20,-80, 10,-100
```

Listing "Mission Wega"

```
2160 DATA 3.1415,0, 2, 200,150,70, 374,112 [1417]
,70
2170 '                                     [117]
2180 ' +++  2.Bild  +++                    [637]
2190 '                                     [117]
2200 DATA 248, 20,-10, 20,-30, 20,-150, 10 [1869]
,-20, 10,0, 10,30, 15,150
2210 DATA 5,125, 5,-150, 20,-75, 10,150, 2 [3058]
0,-200, 10,-100, 20,-50
2220 DATA 55,-5, 20,30, 10,40, 2,100, 10,5 [1787]
0, 20,40, 10,50, 10,75
2230 DATA 5,-275, 10,-50, 20,-50, 30,-50,  [3113]
20,-40, 50,0, 50,20
2240 DATA 20,50, 110,100, 3.1415,0, 2, 202 [2366]
,66,50, 425,64,60
2250 '                                     [117]
2260 ' +++  3.Bild  +++                    [810]
2270 '                                     [117]
2280 DATA 100, 100,20, 30,50, 10,35, 10,50 [2802]
, 5,100, 10,50, 15,30, 15,10, 10,5
2290 DATA 100,-10, 10,-10, 10,-50, 10,-100 [2312]
, 1,-125, 40,0, 1,50, 10,175, 20,150
2300 DATA 30,50, 40,20, 10,0, 10,-10, 10,- [2224]
50, 10,-150, 20,-50, 20,-30, 100,-20
2310 DATA 3.1415,0, 4, 20,108,50, 220,304, [3063]
60, 332,78,46, 570,180,50
2320 '                                     [117]
2330 ' +++  4.Bild  +++                    [778]
2340 '                                     [117]
2350 DATA 300, 50,-10, 30,-20, 10,-50, 1,- [2711]
100, 40,0, 1,50, 10,100, 10,40, 20,30
2360 DATA 50,10, 50,-10, 50,-30, 30,-20, 1 [2431]
0,-40, 1,-125, 38,0, 1,100, 10,100
2370 DATA 10,50, 20,40, 30,20, 10,10, 10,- [3142]
20, 10,-40, 20,-50, 10,-60, 10,-50
2380 DATA 20,-30, 10,-10, 50,-10, 50,-4, 3 [2125]
.1415,0, 4, 88,154,44, 186,310,60
2390 DATA 360,128,42, 580,206,50           [1330]
2400 '                                     [117]
2410 ' +++  5.Bild  +++                    [751]
2420 '                                     [117]
2430 DATA 300, 100,10, 50,0, 50,-10, 20,-1 [2767]
8, 10,-18, 10,-28, 4,-25
2440 DATA -10,-30, -50,-40, -50,-10, -40,- [2752]
30, -20,-30, -10,-30
2450 DATA 1,-20, 10,-40, 10,-25, 10,-15, 2 [2645]
0,-10, 50,-2, 100,6, 20,20, 20,50
2460 DATA 20,75, 5,150, 20,90, 30,50, 70,3 [2996]
0, 50,-40, 20,-100, 10,-150
2470 DATA 5,-50, 5,-40, 10,-10, 10,-5, 40, [1845]
-10, 40,0
2480 DATA 3.1415,0, 4, 50,306,70, 120,78,5 [2517]
0, 386,332,36, 570,110,60
2490 '                                     [117]
2500 ' +++  6.Bild  +++                    [761]
2510 '                                     [117]
2520 DATA 60, 75,10, 20,10, 20,20, 20,30,  [2774]
10,30, 10,250, 10,30, 20,30, 20,20
2530 DATA 20,10, 50,10, 5,-10, 5,-100, 48, [2530]
0, 10,90, 5,10, 10,10, 20,10, 75,6
```

Listing "Mission Wega"

Softwareautoren für die Amstrad Computer gesucht.

Haben Sie nicht auch schon einmal daran gedacht, ein gutes Programm, das Sie selbst geschrieben haben, zu veröffentlichen ?
Warum sollten nicht auch andere Leser in den Genuß Ihrer Mini-Dateiverwaltung, Grafikerweiterung, Tips, Tricks, Tools, Utilities, Simulationen, Games usw. kommen?
Wirklich gute Software, die den Anforderungen unserer Leser genügt, wird von uns entsprechend honoriert.
Sie sollten jedoch bei der Einsendung Ihres Programms ein paar Punkte beachten.
Wenn Sie Nachstehendes befolgen, wird Ihre Post zügig und ohne große Rückfragen und Verzögerungen bearbeitet :

Senden Sie uns Ihr Programm mit
(a) allen benötigten Files auf der mit dem Programmnamen bezeichneten Diskette.
(b) den kompletten Ausdrucken/Listings aller Files der Diskette.
(c) einer Beschreibung Ihres Programms und
(d) einer genauen Bedienungsanleitung.

Die Bedienungsanleitung und die Beschreibung sollten als Textdatei mit auf der Programm-Diskette enthalten sein. Wichtig für uns zu wissen wäre noch, mit welcher Konfiguration Sie arbeiten, welchen Drucker Sie benutzen, ob Sie ein zweites Laufwerk angeschlossen haben usw...
Wenn Sie der Meinung sind, ein solches Programm geschrieben zu haben, dann nichts wie einschicken an den

DMV Verlag · Red. PC International · Abtlg. CPC/Joyce/PC · Fuldaer Str. 6 · 3440 Eschwege

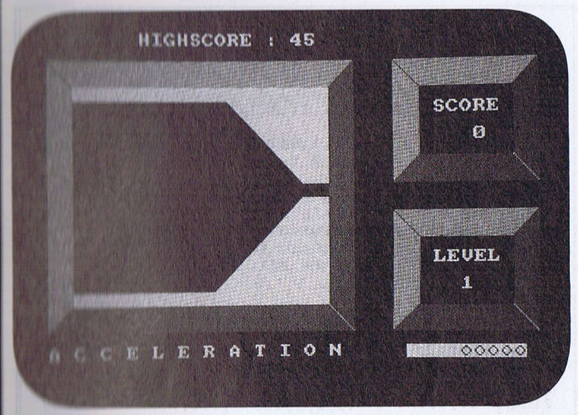
Geben Sie bitte in der Adresse die für Ihren Amstrad Computer (CPC, Joyce, PC) zuständige Abteilung mit an, damit Ihr Programm direkt in die richtigen Hände gelangt.

```
2540 DATA 75,-14, 20,-10, 10,-10, 10,-25,  [2593]
1,-20, -10,-30, -10,-20
2550 DATA -50,-30, -100,-30, -100,-40, -50 [2607]
,-20, -30,-20, -10,-20, -10,-50
2560 DATA 10,-40, 10,-10, 50,0, 10,-10, 20 [2433]
,-30, 10,-20, 70,-4
2570 DATA 10,10, 10,30, 10,50, 30,40, 50,3 [2452]
0, 60,20, 100,10
2580 DATA 3.1415,0, 4, 14,64,40, 280,256,5 [2803]
4, 220,104,46, 316,70,60
2590 '                                     [117]
2600 ' +++  7.Bild  +++                    [605]
2610 '                                     [117]
2620 DATA 300, 70,6, 10,-20, 6,-50, -20,-1 [3567]
50, -4,-50, 50,0, 1,-50, 38,0, 1,150
2630 DATA 10,100, 20,50, 40,20, 70,10, 30, [2758]
0, 40,-20, 20,-20, 10,-30, 1,-50
2640 DATA -20,-50, -60,-30, -30,-50, -6,-2 [2884]
0, 1,-10, 10,-40, 10,-20, 10,-10
2650 DATA 20,-8, 60,4, 20,20, 10,20, 10,40 [3140]
, 10,80, 1,75, 10,40, 10,20, 20,20
2660 DATA 40,0, 20,-20, 10,-20, 10,-40, 10 [2744]
,-80, 1,-100, 10,-20, 100,-20
2670 DATA 3.1415,0, 5, 66,142,44, 110,90,4 [2532]
2, 236,342,60, 322,116,50, 586,124,48
2680 '                                     [117]
2690 ' +++  8.Bild  +++                    [818]
2700 '                                     [117]
2710 DATA 100, 50,0, 10,30, 10,220, 10,50, [3084]
 20,40, 30,30, 40,20, 50,10, 150,0
2720 DATA 50,-10, 20,-10, 10,-10, 2,-10, - [2700]
2,-10, -10,-10, -20,0, -10,10
2730 DATA -100,10, -150,-20, -20,-20, -10, [2671]
-30, -6,-100, 60,0, 10,-10, -10,-20
2740 DATA -80,-10, -10,-10, -10,-100, 10,- [2781]
50, 10,-10, 40,-10, 20,10, 10,50
2750 DATA 20,40, 30,40, 40,30, 50,10, 50,- [2229]
20, 6,-100, 50,0, 10,100, 30,120
2760 DATA 50,50, 50,120, 20,30, 100,50, 3. [2833]
1415,0, 5, 2,104,40, 240,336,70
2770 DATA 130,198,50, 110,70,50, 366,90,56 [1493]
2780 '                                     [117]
2790 ' +++  9.Bild  +++                    [665]
2800 '                                     [117]
2810 DATA 300, 50,-10, 20,-10, 10,-30, 1,- [3121]
50, -40,-100, 1,-100, 40,0, 1,50
2820 DATA 36,100, 1,100, 10,50, 30,40, 50, [2650]
10, 30,-20, 6,-30, 10,-100, 50,0
2830 DATA 10,-10, 1,-5, -10,-10, -80,-10,  [2623]
-20,-30, -10,-40, -1,-50, 10,-30
2840 DATA 30,-20, 50,-10, 40,10, 30,20, 30 [2890]
,30, 10,40, 10,140, 20,180, 10,-100
2850 DATA 1,-200, 36,0, 1,100, 20,80, 50,5 [2542]
0, 100,20,3.1415,0, 5, 36,68,48
2860 DATA 170,330,40, 254,238,50, 242,94,5 [1580]
0, 434,72,42
2870 '                                     [117]
2880 ' +++  10.Bild  +++                   [1157]
2890 '                                     [117]
2900 DATA 320, 70,-10, 20,-10, 10,-40, 1,- [2463]
50, -4,-20, -60,-100, -4,-20, 1,-100
2910 DATA 40,0, 10,130, 4,-80, 40,0, 10,10 [3357]
0, 1,180, 10,20, 150,20, 100,-10
2920 DATA 10,-30, -10,-20, -10,-10, -10,0, [3738]
 -10,10, -120,20, -50,-10, -20,-20
2930 DATA -10,-60, 10,-60, 50,0, 6,-12, -6 [2106]
,-12, -40,0, -10,-20, -8,-150, 10,-20
2940 DATA 50,0, 10,20, 10,100, 100,50, 10, [2375]
50, 120,100, 50,150, 20,10, 20,-10
2950 DATA 1,-200, 34,-0, 1,150, 50,50, 3.1 [2314]
415,0, 5, 28,56,46, 84,72,44
2960 DATA 174,222,50, 184,90,40, 568,142,4 [1791]
0
2970 '                                     [117]
2980 ' +++  11.Bild  +++                   [892]
2990 '                                     [117]
3000 DATA 330, 50,10, 200,-20, 50,-20, 40, [2559]
-40, 20,-60, 10,-80, -10,-60, -20,-40
3010 DATA -30,-30, -10,-5, -10,5, -20,30,  [2832]
-20,90, -30,80, -40,60, -40,10
3020 DATA -40,-20, -40,-30, -10,-30, -10,- [2671]
50, -1,-30, 50,0, 6,-6, 2,-6, -6,-6
3030 DATA -50,-10, -10,-30, 10,-30, 10,-40 [3379]
, 10,-20, 50,0, 40,30, 40,10, 40,-10
3040 DATA 40,-30, 50,-10, 50,0, 50,30, 20, [3211]
200, 10,20, 30,10, 40,-10, 10,-30
3050 DATA 1,-50, 32,0, 1,100, 10,30, 20,40 [3448]
, 50,30, 3.1415,0, 4, 44,184,40
```

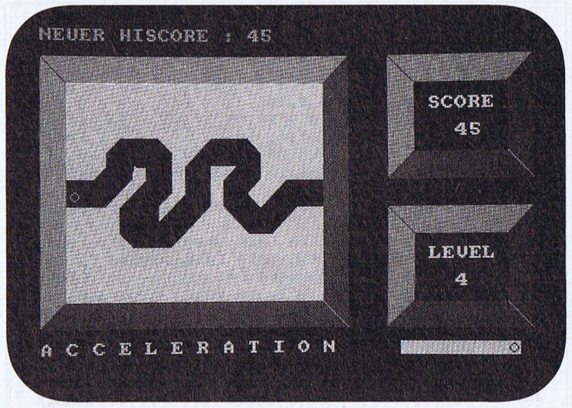
Listing "Mission Wega"

```
3060 DATA 64,90,42, 462,228,50, 528,152,38 [1715]
3070 '                                     [117]
3080 ' +++  12.Bild  +++                   [879]
3090 '                                     [117]
3100 DATA 330, 20,-10, 10,-20, 1,-200, 34, [3714]
0, 1,100, 12,150, 10,20, 20,10, 50,10
3110 DATA 100,10, 150,-10, 20,-10, 10,-20, [4123]
 4,-100, -4,-90, -10,-30, -20,-20
3120 DATA -40,-10, -30,20, -10,120, -10,50 [2330]
, -40,20, -80,10, -50,-20, -20,-40
3130 DATA -4,-40, 10,-20, 70,10, 70,-10, 2 [3184]
0,-170, 20,-20, 50,-10, 70,0, 30,20
3140 DATA 20,50, 10,100, 1,100, 10,30, 30, [2561]
10, 120,20, 3.1415,0, 4, 28,112,38
3150 DATA 360,326,50, 140,188,50, 380,76,6 [1792]
0
3160 '                                     [117]
3170 ' +++  13.Bild  +++                   [1138]
3180 '                                     [117]
3190 DATA 330, 150,10, 300,-10, 50,-30, 30 [1739]
,-50, 10,-50, 1,-80, -10,-50
3200 DATA -10,-30, -20,-20, -30,10, -20,40 [3745]
, -20,50, -10,100, -10,50, -20,40
3210 DATA -50,20, -200,10, -100,-20, -20,- [2521]
40, -10,-80, 10,-20, 60,-4, 10,-40
3220 DATA 20,-4, 20,4, 10,50, 10,30, 10,10 [1681]
, 100,10, 50,-10, 10,-30, 10,-50
3230 DATA -20,-40, -30,-20, -200,-20, -30, [2800]
-20, -20,-40, -10,-40, 10,-50, 50,-20
3240 DATA 100,70, 200,-50, 100,10, 50,20,  [1789]
50,30, 60,100
3250 DATA 3.1415,0, 4, 388,332,50, 28,210, [2668]
40, 92,184,34, 52,54,40
3260 '                                     [117]
3270 ' +++  14.Bild  +++                   [881]
3280 '                                     [117]
3290 DATA 60, 40,0, 10,20, 2,200, 10,70, 3 [2960]
0,50, 50,20, 250,-10, 50,-20, 30,-50
3300 DATA 10,-100, -10,-100, -30,-40, -10, [2377]
-20, -20,-10, -20,20, -30,50, -40,120
3310 DATA -20,40, -10,10, -90,10, -100,-10 [2948]
, -20,-40, -10,-40, 10,-40, 40,0
3320 DATA 20,20, 10,-10, 10,-40, 1,-100, 3 [2409]
4,0, 1,100, 10,30, 30,10, 30,-20
3330 DATA 70,-150, 50,-40, 50,-20, 50,0, 5 [1951]
0,10, 100,50
3340 DATA 3.1415,0, 4, 2,62,34, 84,210,34, [2726]
 160,92,38, 442,96,50
3350 '                                     [117]
3360 ' +++  15.Bild  +++                   [1160]
3370 '                                     [117]
3380 DATA 330, 200,10, 200,-10, 50,-20, 40 [3001]
,-50, 20,-50, 1,-10, -10,-10, -10,10
3390 DATA -50,80, -50,30, -150,10, -100,-2 [2343]
0, -40,-50, -20,-60, -10,-150
3400 DATA -50,-10, -10,-50, 10,-40, 50,-4, [2600]
 30,10, 20,50, 10,180, 40,40, 60,-20
3410 DATA 40,-60, 10,-100, -10,-50, -50,-2 [2355]
0, -20,-40, 10,-50, 50,-10, 40,10
3420 DATA 40,150, 10,150, 10,30, 20,10, 30 [2795]
,-10, 20,-100, 1,-80, -20,-20, -8,-40
3430 DATA 8,-40, 40,0, 12,30, 1,105, 10,30 [2222]
, 50,10, 50,50, 30,100, 100,50
3440 DATA 3.1415,0, 4, 26,100,50, 234,64,4 [3442]
0, 408,60,48, 476,202,36
3450 '                                     [117]
3460 ' +++  16.Bild  +++                   [1411]
3470 '                                     [117]
3480 DATA 330, 250,10, 250,-10, 30,-20, 10 [3028]
,-40, 1,-100, -10,-30, -10,-4, -10,20
3490 DATA -4,120, -20,30, -50,10, -50,-10, [3178]
 -20,-30, -4,-150, -10,-26, -10,0
3500 DATA -20,30, -10,150, -50,10, -10,50, [2514]
 -50,10, -50,-10, -30,-40, -10,-300
3510 DATA -10,300, -40,10, -40,-10, -6,-40 [3092]
, -2,-20, 40,-4, 20,-270, 40,-10
3520 DATA 30,10, 10,40, 20,280, 30,10, 20, [3855]
-80, 50,-10, 14,-150, 20,-30, 50,-10
3530 DATA 40,10, 1,120, 10,50, 30,10, 26,- [2758]
20, 10,-140, 10,-50, 50,-10, 40,20
3540 DATA 20,200, 10,50, 100,50, 3.1415,0, [3118]
 6, 18,224,34, 226,210,40, 84,72,50
3550 DATA 366,104,30, 490,134,46, 538,250, [1670]
60
3560 '                                     [117]
3570 ' +----------+                        [933]
3580 ' :   Ende   :                        [394]
3590 ' +----------+                        [933]
3600 '                                     [117]
```

Listing "Mission Wega"



Am Anfang schon nicht leicht...



...brauchen Sie ab Level 3 gute Nerven.

Acceleration

Ein Computerfreak im All

Wir schreiben das Jahr 3147. Der Planet WEGA-S soll zur menschlichen Kolonie gemacht werden. Forschungstrupps sind auf eine riesige Höhle gestoßen. Ohne zu wissen, was sich am anderen Ende der Höhle befindet, kann WEGA-S nicht zur Kolonie freigegeben werden. Also schickt man einen ahnungslosen Computerfreak mit einem Raumschiff in die Tiefen des Höhlensystems.

Nachdem Sie in Ihrem Raumschiff Platz genommen haben, sollten Sie sich zunächst mit der Steuerung vertraut machen. Das Raumschiff besitzt an jeder Seite eine Steuerdüse, die mittels Joystick aktiviert werden.

Bei Drücken der Feuertaste werden die Düsen aktiviert, ohne daß die Richtung geändert wird. So können Sie bei engen Tunnelgängen testen, ob man überhaupt hindurchpaßt.

Die Punktewertung: Nach erfolgreichem Absolvieren eines Levels (insgesamt sind elf Level zu durchfliegen!) wird die Anzahl der verbleibenden Leben mit der Nummer des Levels multipliziert und zu den erreichten Punkten addiert.

Die Anzahl der verbleibenden Leben können Sie durch die Anzeige im rechten unteren Spielfeld jederzeit ablesen. Hinweis: Wollen Sie die verschiedenen Level einzeln anwählen, so geben Sie ein:

```
2305 le= (Nummer des Levels)
```

(Pascal von Hutten/sr)

Für 464-664-6128

```
1000 ' %%%%%%%%%%%%%%%%%%                 [886]
1010 ' %                  %                [230]
1020 ' %  ACCELERATION    %                [324]
1030 ' %                  %                [230]
1040 ' %   written by     %                [410]
1050 ' %                  %                [230]
1060 ' %  P. v. Hutten    %                [878]
1070 ' %                  %                [230]
1080 ' %  Dezember '87    %                [311]
1090 ' %                  %                [230]
1100 ' %  (ORTEX+S.P.)    %                [844]
1110 ' %                  %                [230]
1120 ' %%%%%%%%%%%%%%%%%%                 [886]
1130 '                                     [117]
1140 '                                     [117]
1150 ' %%%%% Voreinstellungen %%%%%        [2048]
1160 SYMBOL AFTER 32:SYMBOL 255,128,64,32, [3731]
16,8,4,2,1:GOSUB 1780:  ' Vorspann
1170 SYMBOL AFTER 97:SYMBOL 97,0,24,36,66, [12874]
66,36,24,0:SYMBOL 98,66,153,36,66,66,36,15
3,66:SYMBOL 99,54,73,165,145,74,50,70,50:S
YMBOL 100,88,164,82,73,149,170,82,12:SYMBO
L 101,8,52,74,50,68,48,8,0:SYMBOL 102,0,16
,42,40,90,34,0,0
1180 SYMBOL 104,0                          [895]
1190 SYMBOL 103,0,0,4,16,8,36,0,0          [1580]
1200 MODE 1:INK 0,0:INK 1,1:INK 2,2:INK 3, [4315]
13:BORDER 0:PAPER 0:CLS
1210 al=10                                 [213]
```

Listing "Acceleration"

```
1220 ENV 1,5,0,1,10,-1,1:ENV 2,15,-1,10:EN [2467]
T 1,50,1,1,50,-1,1,50,1,1
1230 GOSUB 1250                            [873]
1240 GOTO 1390                             [421]
1250 ' %%%%% Bildschirmaufbau %%%%%        [1913]
1260 SYMBOL 255,128,64,32,16,8,4,2,1       [1874]
1270 xd=20:yd=16:x=3:y=3:GOSUB 1330        [2334]
1280 xd=7:yd=5:x=30:y=3:GOSUB 1330         [1787]
1290 xd=7:yd=5:x=30:y=14:GOSUB 1330        [3327]
1300 PEN 3:PAPER 0:LOCATE 33,6:PRINT"SCORE [5998]
";:LOCATE 33,17:PRINT"LEVEL";:LOCATE 3,24:
PRINT"A C C E L E R A T I O N";
1310 LOCATE 10,1:PRINT"HIGHSCORE :"hs      [1710]
1320 RETURN                                [555]
1330 ' %%%%% Unterprgr:Rahmen %%%%%        [1072]
1340 PEN 1:PAPER 2:LOCATE x,y:PRINT CHR$(2 [4783]
55)+STRING$(xd+2,32)+CHR$(214);
1350 LOCATE x,y+1:PRINT " "+CHR$(255)+STRI [4702]
NG$(xd,32)+CHR$(214)+CHR$(143);
1360 FOR t=y+2 TO yd+y+1:LOCATE x,t:PRINT" [7194]
 ";:LOCATE x+xd+2,t:PRINT CHR$(143);CHR$(
143);:NEXT
1370 LOCATE x,y+yd+2:PRINT CHR$(32);CHR$(2 [8826]
14);STRING$(xd,143);CHR$(24);CHR$(255);CHR
$(24);CHR$(143);:LOCATE x,y+yd+3:PRINT CHR
$(214);STRING$(xd+2,143);CHR$(24);CHR$(255
);CHR$(24);
1380 RETURN                                [555]
1390 ' %%%%% Hauptprogramm %%%%%           [867]
1400 li=5:sc=0:le=1:GOSUB 1530:GOSUB 1580: [3970]
GOSUB 1620:PLOT 1000,1000,3:GOSUB 2299
1410 PLOT 1000,1000,3:u=0:r=0:x=66:y=210:I [2140]
F le>al THEN GOTO 2271
1420 TAG:IF TEST(x,y)<>0 OR TEST(x,y-14)<> [6461]
```

Listing "Acceleration"

```
0 OR TEST(x+15,y-14)<>0 OR TEST(x+15,y)<>0
 THEN GOSUB 1650 ELSE CALL &BD19:MOVE x,y:
PRINT ship$;
1430 j=JOY(0):IF j=0 THEN ship$="a" ELSE s [5261]
hip$="b":SOUND 3,1000,5,15,,,1
1440 CALL &BD19:MOVE x,y:PRINT " ";        [1537]
1450 IF j>=16 THEN f=1:j=j-16              [808]
1460 IF j>=8 THEN j=j-8:IF r<25 THEN r=r+1 [1137]
1470 IF j>=4 THEN j=j-4:IF r>-25 THEN r=r- [1942]
1
1480 IF j>=2 THEN j=j-2:IF u>-25 THEN u=u- [2167]
1
1490 IF j=1 THEN IF u<25 THEN u=u+1        [1138]
1500 x=x+r:y=y+u:IF x+15>=384 THEN 2090     [3152]
1510 IF x<=64 THEN r=0:x=64:u=0             [1457]
1520 GOTO 1420                             [399]
1530 ' %%%%% PRINT lives %%%%%             [1068]
1540 PEN 0:PAPER 3                         [741]
1550 li$=STRING$(9-li," ")+STRING$(li,"a") [1742]
1560 TAGOFF:LOCATE 31,24:PRINT li$;        [1953]
1570 RETURN                                [555]
1580 ' %%%%% PRINT score %%%%%             [626]
1590 PEN 3:PAPER 0                         [617]
1600 TAGOFF:LOCATE 33,8:PRINT USING"####"; [2683]
sc
1610 RETURN                                [555]
1620 ' %%%%% PRINT level %%%%%             [899]
1630 TAGOFF:PEN 3:PAPER 0:LOCATE 34,19:IF  [4903]
le=11 THEN PRINT"ENDE" ELSE PRINT USING"##
";le
1640 RETURN                                [555]
1650 ' %%%%% Zusammenstoss %%%%%           [1358]
1660 tst(1)=TEST(x,y):tst(2)=TEST(x,y-14): [4293]
tst(3)=TEST(x+15,y-14):tst(4)=TEST(x+15,y)
1670 FOR t=1 TO 4:IF tst(t)=3 THEN st=1 EL [3524]
SE IF tst(t)=0 THEN st=st
1680 NEXT                                  [350]
1690 IF st=1 THEN st=0:GOTO 1700 ELSE RETU [1721]
RN
1700 ' %%%%% Explosion %%%%%               [394]
1710 PLOT 50,100,2:SOUND 3,800,15,15,1,,10 [5827]
:FOR u=1 TO 6:MOVE x,y:PRINT CHR$(98+u);:F
OR i=1 TO 50:NEXT:NEXT
1720 ' %%%%% -1 Lebel %%%%%               [497]
1730 li=li-1:IF li=-1 THEN GOTO 1740 ELSE  [3939]
GOSUB 1530:GOTO 1410
1740 ' %%%%% Game over %%%%%              [1001]
1750 TAGOFF:RESTORE 1770:PEN 2:PAPER 0:PRI [10779]
NT CHR$(22);:FOR t=1 TO 20 STEP 2:READ a$:
LOCATE t+3,10:PRINT a$;:IF a$<>"  " AND t<
>1 THEN SOUND 3,1200,15,15,1:FOR i=1 TO 30
0:NEXT
1760 NEXT:PRINT CHR$(22)CHR$(0);:FOR t=1 T [3797]
O 5000:NEXT:GOTO 2130
1770 DATA "G ","G ","A ","M ","E ","  ","O [2249]
 ","V ","E ","R "
1780 ' %%%%% Vorspann %%%%%               [54]
1790 ls$="                Ortex+  Software  [11952]
 Productions   praesentiert   :   ACCELER
ATION   - - -   Autor:  Pascal von Hutten
  -   Dezember '87 und Januar '88.......
.. "
1800 GOTO 1840                             [395]
1810 FOR t=1 TO 170                        [818]
1820 PEN 3:LOCATE 7,10:PRINT MID$(ls$,t,25 [2621]
);:NEXT
1830 LOCATE 13,10:PRINT"> S P A C E <":SPE [3353]
ED INK 15,15:GOSUB 1900:RETURN
1840 SPEED INK 5,5:MODE 1:INK 0,0:INK 1,3, [6648]
6:INK 2,6,3:INK 3,24,12:BORDER 0:PAPER 0:P
EN 2:CLS
1850 RESTORE 1870:FOR t=1 TO 4:READ a$:LOC [5393]
ATE 20-(LEN(a$)/2),t+2:PRINT a$:NEXT
1860 xd=25:yd=1:x=5:y=8:GOSUB 1330:PEN 1:P [2636]
APER 0
1870 DATA SOFTWARE,DER,NEUEN,DIMENSION     [2140]
1880 FOR t=1 TO 2:FOR l=1 TO 7:SOUND 1,120 [5924]
0,4,1:NEXT:NEXT:FOR l=7 TO 0 STEP-1:SOUND
1,1200,50,l:FOR t=1 TO 200:NEXT:NEXT
1890 GOTO 1810                             [377]
1900 ' %%%%% Titel-Sound %%%%%             [452]
1910 ENV 1,15,-1,2                         [627]
1920 '                                     [117]
1930 RESTORE 2070                          [792]
1940 FOR t=1 TO 9                          [578]
1950 READ s                                [418]
1960 ON s GOSUB 2010,2030,2050             [797]
1970 IF INKEY$=" " THEN GOTO 2080          [2287]
1980 NEXT                                  [350]
1990 IF INKEY$=" " THEN GOTO 2080          [2287]
```

Listing "Acceleration"

```
2000 GOTO 1930                             [377]
2010 ' drum 1                              [1049]
2020 SOUND 3,750,0,0,1,,15:RETURN          [2483]
2030 ' drum 2                              [1056]
2040 SOUND 3,400,0,0,1,,7:RETURN           [1886]
2050 ' drum 3                              [1055]
2060 SOUND 3,1,0,0,1,,1:RETURN             [1744]
2070 DATA 2,1,1,2,1,2,2,1,3                [590]
2080 RETURN                                [555]
2090 ' %%%%% naechstes Level %%%%%         [1575]
2100 TAGOFF:PEN 1:LOCATE 25,13:PRINT CHR$( [10769]
143);:LOCATE 25,12:PRINT CHR$(143);:GOSUB
2240:le=le+1:GOSUB 1620:sc=sc+((li+1)*le):
GOSUB 1580:IF le>al THEN 2271
2110 GOSUB 2299                            [884]
2120 GOTO 1410                             [393]
2130 ' %%%%% HighScore ??? %%%%%           [575]
2140 INK 0,9:INK 1,6:INK 2,3:INK 3,26:BORD [4311]
ER 12:PAPER 0:MODE 1:FOR t=1 TO 100:i$=i$+
INKEY$:NEXT:i$=""
2150 ' Rahmen                              [810]
2160 x=1:y=1:xd=36:yd=21:GOSUB 1330        [2206]
2170 x=3:y=3:xd=32:yd=17:GOSUB 1330        [2614]
2180 PAPER 0:IF sc>hs THEN hs=sc:GOTO 2200 [2365]
2190 PEN 3:LOCATE 10,7:PRINT"PECH GEHABT ! [4288]
!":LOCATE 8,9:PRINT"KEIN HIGHSCORE .":GOTO
 2210
2200 PEN 3:LOCATE 10,7:PRINT"GUT GEMACHT ! [3254]
!":LOCATE 8,9:PRINT"NEUER HISCORE ."
2210 LOCATE 10,11:PEN 1:PRINT"SCORE :";:PE [6381]
N 2:PRINT sc;:LOCATE 10,13:PEN 1:PRINT"HIG
H  :";:PEN 2:PRINT hs;
2220 GOSUB 1900                            [895]
2230 GOTO 1170                             [305]
2240 ' %%%% SOUND : LEVEL %%%%%            [1792]
2250 SPEED INK 3,4:BORDER 3,1              [1996]
2260 FOR t=7 TO 0 STEP-1:SOUND 3,200,13,t: [2499]
NEXT
2270 BORDER 0:RETURN                       [907]
2271 ' %%%%% alle Level geschafft %%%%%    [1730]
2272 le=11:GOSUB 2299:xp=200:yp=200:xg=24: [8589]
yg=14:GOSUB 2279:xp=300:yp=230:xg=12:yg=16
:GOSUB 2279:xp=180:yp=250:xg=20:yg=28:GOSU
B 2279
2273 FOR t=1 TO 100:i$=i$+INKEY$:NEXT:i$=" [1640]
"
2274 a$="                          SIE HABEN  [14052]
DAS ENDE DER HOEHLE ERREICHT. IN DER HOEHL
E LEBEN SELTSAME WESEN, DIE ABER FUER DEN
MENSCHEN HARMLOS SIND. WEGA-5 KANN KOLONIA
LISIERT WERDEN !!!          >>>> TASTE <<<<
 "
2275 SOUND 3,1000,1640,5:FOR t=1 TO 180:PE [6216]
N 3:PAPER 0:LOCATE 3,24:PRINT MID$(a$,t,24
);:NEXT
2276 FOR t=1 TO 2000:NEXT:IF INKEY$="" THE [4019]
N GOTO 2275 ELSE GOTO 2130
2278 CALL &BB06                            [393]
2279 FOR a=0 TO 360 STEP 20:DEG:PLOT xp,yp [5509]
,2:DRAW xp+xg*COS(a),yp+yg*SIN(a):NEXT:RET
URN
2280 ' %%%%% alle Level geschafft %%%%%    [1730]
2299 ' %%%%% Level aus Datazeile holen     [3525]
          und aufbauen %%%%%
2300 PEN 3:PAPER 0:l=1                     [751]
2310 IF le=1 THEN RESTORE 2500             [611]
2320 IF le=2 THEN RESTORE 2520             [1547]
2330 IF le=3 THEN RESTORE 2550             [1575]
2340 IF le=4 THEN RESTORE 2690             [1570]
2350 IF le=5 THEN RESTORE 2720             [1330]
2360 IF le=6 THEN RESTORE 2770             [541]
2370 IF le=7 THEN RESTORE 2820             [1151]
2380 IF le=8 THEN RESTORE 2860             [2248]
2390 IF le=9 THEN RESTORE 2890             [1270]
2400 IF le=10 THEN RESTORE 2920            [1208]
2405 IF le=11 THEN RESTORE 2950            [1944]
2410 IF sc>hs THEN PEN 2:LOCATE 3,1:PRINT" [2824]
NEUER HISCORE :"sc"    "
2420 PEN 3                                 [547]
2430 READ x,y:IF x=-1 THEN RETURN          [2886]
2440 IF y=0 THEN s=32 ELSE IF y=1 THEN s=1 [7745]
43 ELSE IF y=2 THEN s=212 ELSE IF y=3 THEN
 s=213 ELSE IF y=4 THEN s=214 ELSE IF y=5
THEN s=215
2450 l$=l$+STRING$(x,s):IF LEN(l$)=20 THEN [2421]
 GOSUB 2470:l$=""
2460 GOTO 2430                             [349]
2470 ' aufbau                              [474]
2480 LOCATE 5,l+4:PRINT l$;:l=l+1:RETURN   [2260]
2490 ' %%%%% LEVEL 1 %%%%%                 [1175]
```

Listing "Acceleration"


```
2500 DATA 20,1,12,0,1,3,7,1,13,0,1,3,6,1,1 [8202]
4,0,1,3,5,1,15,0,1,3,4,1,16,0,1,3,3,1,17,0
,1,3,2,1,20,0,17,0,1,4,2,1,16,0,1,4,3,1,15
,0,1,4,4,1,14,0,1,4,5,1,13,0,1,4,6,1,12,0,
1,4,7,1,11,0,1,4,8,1,20,1
2510 DATA -1,-1                            [418]
2520 ' %%%%% LEVEL 2 %%%%%                [1015]
2530 DATA 20,1,20,1,20,1,20,1,20,1,20,1,20 [5026]
,1,5,0,1,3,8,1,1,2,5,0,20,0,4,1,1,5,10,0,1
,4,4,1,20,1,20,1,20,1,20,1,20,1,20,1
2540 DATA -1,-1                            [418]
2550 ' %%%%% Level 3 %%%%%                [843]
2560 DATA 20,1,20,1,7,1,1,2,4,0,1,3,7,1    [1369]
2570 DATA 6,1,1,2,6,0,1,3,6,1              [1012]
2580 DATA 5,1,1,2,2,0,1,4,2,1,1,5,2,0,1,3, [1801]
5,1
2590 DATA 4,1,1,2,2,0,1,4,4,1,1,5,2,0,1,3, [1034]
4,1
2600 DATA 3,1,1,2,2,0,1,4,6,1,1,5,2,0,1,3, [2591]
3,1
2610 DATA 5,0,1,4,8,1,1,5,5,0              [1001]
2620 DATA 5,0,1,3,8,1,1,2,5,0              [835]
2630 DATA 3,1,1,5,2,0,1,3,6,1,1,2,2,0,1,4, [1313]
3,1
2640 DATA 4,1,1,5,2,0,1,3,4,1,1,2,2,0,1,4, [1180]
4,1
2650 DATA 5,1,1,5,2,0,1,3,2,1,1,2,2,0,1,4, [1598]
5,1
2660 DATA 6,1,1,5,6,0,1,4,6,1              [908]
2670 DATA 7,1,1,5,4,0,1,4,7,1,20,1,20,1    [1070]
2680 DATA -1,-1                            [418]
2690 ' %%%%% Level 4 %%%%%                [1189]
2700 DATA 20,1,20,1,20,1,20,1,3,1,1,2,3,0, [9939]
1,3,2,1,1,2,3,0,1,3,5,1,3,1,5,0,1,1,1,2,5,
0,5,1,2,1,1,2,2,0,1,1,2,0,1,1,2,0,1,1,1,2,
2,0,5,1,4,0,1,4,1,2,2,0,1,1,2,0,1,1,2,0,1,
4,1,1,1,2,3,0,3,0,1,4,1,2,2,0,1,4,1,2,2,0,
1,1,2,0,1,3,1,2,4,0,4,1,3,0,1,2,2,0,1,4,1,
1
2710 DATA 1,5,4,0,1,4,2,1,4,1,5,0,1,4,3,1, [4149]
1,5,2,0,1,4,3,1,4,1,1,5,3,0,1,4,11,1,20,1,
20,1,20,1,20,1,-1,-1
2720 ' %%%%% Level 5 %%%%%                [1036]
2730 DATA 20,1,7,1,1,2,5,0,1,3,6,1,1,1,1,2 [11565]
,3,0,1,3,1,1,7,0,1,3,5,1,1,1,5,0,1,1,2,0,3
,1,1,5,2,0,1,3,4,1,1,1,2,0,1,1,2,0,1,1,5,0
,1,3,1,5,2,0,4,1,1,1,2,0,1,1,2,0,1,1,1,5,5
,0,1,1,2,0,4,1,1,1,2,0,1,1,2,0,5,1,2,0,1,1
,2,0,4,1,3,0,1,1,2,0,1,1,1,2,5,0,1,1,2,0,1
,1
2740 DATA 1,2,2,0,2,0,1,4,1,2,2,0,1,1,5,0, [10701]
1,4,1,1,2,0,1,2,3,0,2,1,1,2,2,0,1,4,1,1,2,
0,5,1,4,0,1,4,1,1,1,1,1,2,2,0,1,4,1,1,1,2,
2,0,5,1,1,5,2,0,1,4,2,1,1,1,2,0,1,4,1,1,1,
2,2,0,1,4,5,1,1,2,1,0,1,4,3,1,1,1,2,0,1,1,
1,2,10,0,1,4,4,1,1,1,5,0,1,4,13,1,1,1,1,5,
3,0
2750 DATA 1,4,14,1,20,1,-1,-1              [988]
2760 DATA -1,-1                            [418]
2770 ' %%%%% Level 6 %%%%%                [1125]
2780 DATA 15,1,1,5,1,0,1,3,2,1,16,1,1,5,1, [12161]
0,1,3,1,1,1,1,1,2,2,0,1,3,1,2,8,0,1,3,2,1,
1,5,1,0,1,3,1,1,14,0,3,1,1,5,1,0,1,1,2,0,1
```

Listing "Acceleration"

```
,3,4,1,1,0,4,1,2,0,4,1,1,5,1,1,1,5,2,0,1,3
,3,1,1,0,1,1,5,0,5,1,2,1,1,5,2,0,1,3,2,1,1
,0,1,1,4,0,1,4,1,2,1,0,1,3,2,1,2,0,1,3,1,5
2790 DATA 2,0,1,3,1,1,1,0,1,1,2,0,2,1,1,2, [10040]
3,0,1,3,1,1,3,0,1,3,1,5,2,0,1,1,1,0,1,1,9,
0,1,1,1,1,1,5,2,0,1,3,2,0,1,1,1,0,1,1,1,5,
4,0,1,4,1,1,2,0,1,1,2,1,1,5,4,0,1,1,1,0,4,
1,1,0,3,1,1,5,1,0,1,1,3,1,1,5,3,0,1,1,7,0,
2,1,1,2,1,0,1,1,4,1,1,5,1,0,1,4,1,1,1,0,8,
1
2800 DATA 3,0,4,0,13,1,1,5,2,0,3,1,1,0,16, [1858]
1,3,1,1,0,16,1
2810 DATA -1,-1                            [418]
2820 ' %%%%% Level 7 %%%%%                [984]
2830 DATA 20,1,20,1,20,1,20,1,20,1,10,1,1, [13027]
2,4,0,5,1,10,1,9,0,1,1,2,0,1,3,7,1,1,5,2,0
,1,3,3,1,1,0,2,1,3,0,1,1,1,2,3,0,1,3,1,2,1
,3,1,5,2,0,3,1,1,0,2,1,1,1,1,5,1,0,1,1,7,0
,1,1,2,0,3,1,1,0,2,1,2,1,4,0,1,1,1,2,2,0,1
,4,1,2,2,0,3,1,1,0,2,1,2,1,1,5,2,0,1,4,1,1
2840 DATA 2,0,1,4,1,2,2,0,1,4,3,1,1,0,2,1, [4292]
7,1,1,5,5,0,4,1,3,0,8,1,1,5,1,0,1,4,1,5,1,
4,4,1,1,0,2,1,20,1,20,1
2850 DATA -1,-1                            [418]
2860 ' %%%%% Level 8 %%%%%                [805]
2870 DATA 20,1,1,1,1,2,16,0,1,3,1,1,1,1,18 [11799]
,0,1,1,1,1,2,0,9,1,1,0,4,1,2,0,1,1,1,1,2,0
,1,1,1,2,14,0,1,1,1,1,2,0,1,1,14,0,1,4,1,1
,1,1,2,0,1,1,2,0,2,1,1,0,11,1,3,0,1,1,14,0
,1,3,1,1,2,0,1,4,1,1,1,5,14,0,1,1,12,1,1,0
,4,1,2,0,1,1,4,1,1,2,14,0,1,1,2,1,1,2,15,0
2880 DATA 1,4,1,1,1,1,1,2,3,0,1,4,1,0,13,1 [2986]
,1,1,3,0,2,1,1,0,13,1,1,1,1,5,18,0,20,1,-1
,-1
2890 ' %%%%% Level 9 %%%%%                [1158]
2900 DATA 20,1,20,1,20,1,20,1,20,1,19,1,1, [11122]
2,18,1,1,2,1,0,4,0,1,3,1,2,4,0,1,3,6,1,1,2
,2,0,1,0,1,4,1,5,4,0,1,4,1,5,2,0,1,3,4,1,1
,2,2,0,1,4,9,1,1,5,8,0,1,4,1,1,10,1,1,5,2,
0,1,4,1,5,2,0,1,4,2,1,20,1,20,1,5,1,1,2,6,
0,8,1,4,1,1,2,7,0
2910 DATA 8,1,3,1,1,2,1,0,1,4,5,1,1,0,8,1, [1592]
-1,-1
2920 ' %%%%% Level 10 %%%%%               [1175]
2930 DATA 20,1,20,1,14,1,1,0,5,1,14,1,5,0, [11338]
1,1,13,1,1,2,1,0,1,3,1,1,1,0,2,1,13,1,3,0,
1,1,3,0,13,1,1,5,1,0,1,4,4,1,4,0,1,3,9,1,1
,0,5,1,9,0,1,3,4,1,1,0,5,1,4,0,1,4,2,1,1,5
,2,0,1,3,3,1,1,0,5,1,8,1,1,5,3,0,1,3,1,2,1
,0,5,1,9,1,1,5,5,0,5,1,20,1,20,1,20,1,20,1
2940 DATA -1,-1                            [418]
2950 ' %%%%% END - LEVEL %%%%%             [883]
2960 DATA 20,1,20,1,2,1,16,0,2,1,2,1,16,0, [7558]
2,1,2,1,16,0,2,1,2,1,16,0,2,1,2,1,16,0,2,1
,18,0,2,1,18,0,2,1,2,1,16,0,2,1,2,1,16,0,2
,1,2,1,16,0,2,1,2,1,16,0,2,1,2,1,16,0,2,1,
20,1,20,1,-1,-1
```

Listing "Acceleration"

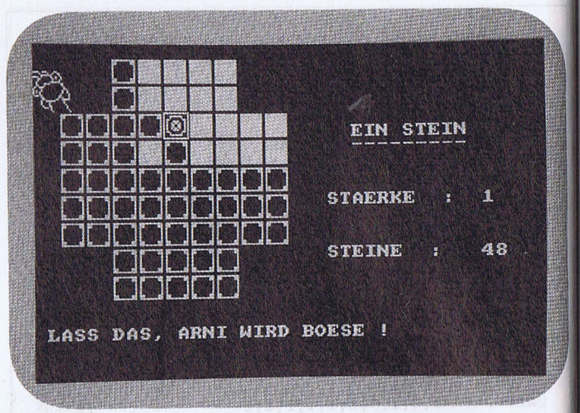
Eine Bitte an unsere Abonnenten

Vermerken Sie bei Schriftverkehr und Zahlungen neben der vollständigen Anschrift stets Ihre Abo-Nummer.

Sie vermeiden damit unnötige Verzögerungen bei der Bearbeitung Ihres Abonnements.

Vielen Dank
Ihre DMV-Versandabteilung

EinStein mit Senf und Arni



EinStein ist eine Version des bekannten Solitaire.

Bei diesem Spiel, das vielleicht unter dem Namen "Solitaire" bekannt sein dürfte, handelt es sich um ein relativ einfaches Strategiespiel.

Zu Beginn befinden sich 64 Steine und ein freies Feld auf dem Spielbrett. Der Spieler nimmt nun einen Stein, springt mit diesem über einen anderen Stein und entfernt den übersprungenen. Gesprungen werden darf nur waagerecht oder senkrecht. Ziel des Spiels ist es, daß am Ende ein einziger Stein übrig bleibt. Die Bedienung sieht nun folgendermaßen aus: Der Spieler steuert entweder mit den Cursortasten oder dem Joystick ein kleines Kreuz. Durch Drücken der <COPY>-Taste oder des Feuerknopfs kann er einen Stein anwählen, dann fährt er auf das freie Feld und drückt nochmals dasselbe. Bei einer Fehlbedienung wird eine Meldung unterhalb des Spielfeldes angezeigt. Falls der Spieler nicht mehr ziehen kann, drückt er <E> für Ende. Es folgt eine Spielbewertung und die Punktezahl, die sich aus Zeit und übriggebliebenen Steinen berechnet. Außerdem wird der Spieler gefragt, ob er weiterspielen möchte oder nicht. Zuletzt noch eine Grundregel: Immer wenn das Programm hält, einfach eine beliebige Taste drücken! Damit das Ganze nicht zu trocken wirkt, begleitet den Spieler eine Witzfigur namens "Senf". Er ist, kurz gesagt, ein Möchtegerntrickser, der während des Spiels herumblödelt, sich über den Spieler lustig macht und andauernd seinen Senf dazugibt. Musikalisch untermalt wird das ganze mit Paulchen Panther.

Zur Eingabe: Tippen Sie das Listing vollständig ab, und speichern Sie es vorsichtshalber auf Kassette oder Diskette. (Im Listing sind POKES enthalten, die, wenn falsch angewandt, zum Absturz führen können.). Das Listing selber ist leicht verständlich und durch die Remarks auch leicht veränderbar (Farben, Sound, ...). Die REM-Zeilen können weggelassen werden.

Viel Spaß!

(A. Huber/sr)

für 464

```
10 REM ******************************      [1365]
20 REM * EINSTEIN MIT SENF UND ARNI *      [1787]
30 REM *      1987 HUBER ALEXANDER  *      [2610]
40 REM ******************************      [1365]
50 CLEAR:SYMBOL AFTER 32:MODE 1:CLS:RANDOM [2073]
IZE TIME
60 LOCATE 14,12:PRINT"BITTE WARTEN!":PRINT [4241]
 CHR$(23)CHR$(0)
70 DIM ton(84),lae(84):ENV 1,10,-1,10:KEY  [1880]
DEF 70,0,0,0,0:POKE &B4E8,255
80 GOSUB 2010                              [817]
90 GOSUB 1260                              [863]
100 INK 0,0:INK 1,26:INK 2,20:INK 3,25     [1345]
110 MODE 1:PAPER 1:PEN 0:BORDER 0:CLS      [2338]
120 xx=116:yy=380:bx=15:by=13:grx=15:gry=1 [3670]
3:s=0:w$=a$
130 GOSUB 1880                             [867]
140 yy=276:w$=b$                           [1096]
150 GOSUB 1880                             [867]
160 yy=172:w$=c$                           [649]
170 GOSUB 1880                             [867]
180 POKE &B939,&F3                         [487]
190 WINDOW 2,39,21,24:PEN 1:PAPER 0:CLS    [2857]

200 LOCATE 4,21:PRINT text$(0);            [1072]
210 LOCATE 6,22:PRINT text$(1);            [1573]
220 LOCATE 1,23:PRINT text$(2);            [1976]
```

Listing "Einstein"

```
230 REM ******************************      [1504]
240 REM *     PAULCHEN PANTHER     *         [1416]
250 REM ******************************      [1504]
260 t=0                                     [273]
270 WHILE INKEY$=""                         [1095]
280 SOUND 1,ton(t),lae(t),15,1              [1740]
290 SOUND 2,ton(t),lae(t),15,1              [1746]
300 SOUND 4,ton(t),lae(t),15,1              [2171]
310 t=t+1:IF t=84 THEN t=0                  [448]
320 WEND                                    [390]
330 PRINT CHR$(2)                           [1059]
340 WINDOW 1,41,1,26                        [869]
350 REM ******************************      [1504]
360 REM *   BILDSCHIRM LOESCHEN    *        [1348]
370 REM ******************************      [1504]
380 MOVE 0,0:lx=0:BORDER 10                 [923]
390 WHILE x<200                             [1167]
400 DRAW 640-lx,y,0:DRAW 640-lx,400-llx:DR  [4084]
AW x,400-llx:DRAW x,y
410 x=x+1:y=y+1:lx=lx+1:llx=llx+1           [2710]
420 WEND                                    [390]
430 REM ******************************      [1504]
440 REM *        HAUPTMENUE          *      [723]
450 REM ******************************      [1504]
460 LOCATE 16,3:PRINT"EIN STEIN"            [1846]
470 LOCATE 16,4:PRINT STRING$(9,"-")        [1359]
480 LOCATE 11,6:PRINT"NATUERLICH MIT SENF"  [1973]
490 LOCATE 9,8:PRINT"UND NICHT ZU VERGESSE  [2555]
N,"
500 LOCATE 10,10:PRINT"MIT ARNI, DEM COMPI  [1935]
 !"
510 LOCATE 19,12:PRINT g$                   [559]
520 IF f1=1 THEN 610                        [467]
530 LOCATE 14,17:PRINT"SPIELSTAERKE!"       [1579]
```

Listing "Einstein"


```
540 LOCATE 15,19:PRINT"1 - LEICHT!"         [1751]
550 LOCATE 15,20:PRINT"2 - MITTEL!"         [2381]
560 LOCATE 15,21:PRINT"3 - SCHWER!"         [2053]
570 ik$=INKEY$:IF ik$="" THEN 570           [2528]

580 IF ASC(ik$)<49 OR ASC(ik$)>51 THEN 570 [1687]

590 ON VAL(ik$) GOTO 2820,2860,2900         [629]

600 GOTO 570                                [379]
610 WINDOW 2,39,16,24:CLS                   [1801]
620 LOCATE 1,1:PRINT TEXT$(3)               [1003]
630 LOCATE 1,5:PRINT STRING$(38,".");       [2158]
640 LOCATE 1,5:INPUT "",name$               [2406]
650 IF LEN(name$)>38 THEN 610               [1744]
660 LOCATE 1,7:PRINT TEXT$(4);              [1942]
670 CALL &BB06                              [393]
680 REM ******************************      [1365]
690 REM *     SPIELFELDAUFBAU        *      [996]
700 REM ******************************      [1365]
710 steine=64:pkt=0:zug=0:t=0:zeit=0:bo=2:  [1769]
flag=0
720 INK 3,2,26:INK 0,23:INK 1,0:INK 2,15    [2204]
730 PAPER 1:PEN 0:BORDER 1                  [1202]
740 MODE 1:CLS                              [1073]
750 LOCATE 29,2:PRINT g$;                   [722]
760 DIM xy(22,21)                           [908]
770 FOR X=7 TO 15 STEP 2                    [960]
780 LOCATE X,2:PRINT d$;                    [1554]
790 xy(x,2)=1                               [459]
800 LOCATE x,4:PRINT d$;                    [1120]
810 xy(x,4)=1                               [533]
820 LOCATE x,16:PRINT d$;                   [1841]
830 xy(x,16)=1                              [496]
840 LOCATE x,18:PRINT d$;                   [971]
850 xy(x,18)=1                              [459]
860 NEXT                                    [350]
870 y=6                                     [373]
880 WHILE y<15                              [1184]
890 FOR x=3 TO 19 STEP 2                    [1326]
900 LOCATE x,y:PRINT d$;                    [612]
910 xy(x,y)=1                               [754]
920 NEXT                                    [350]
930 y=y+2                                   [816]
940 WEND                                    [390]
950 LOCATE 11,10:PRINT e$;                  [762]
960 xy(11,10)=2                             [803]
970 LOCATE 26,7:PRINT"EIN STEIN"            [1756]
980 LOCATE 26,8:PRINT STRING$(9,"-")        [1397]
990 LOCATE 24,12:PRINT"STAERKE  :  1"       [2038]
1000 LOCATE 24,16:PRINT"STEINE  : ";steine  [2395]
1010 PEN 3                                  [547]
1020 GOSUB 2980                             [823]
1030 x=7:y=2                                [486]
1040 LOCATE x,y:PRINT f$                    [795]
1050 REM ******************************     [1365]
1060 REM * TATSTATUR- JOYSTICKABFRAGE *     [2836]
1070 REM ******************************     [1365]
1080 bx=0:by=0                              [400]
1090 ik$=INKEY$:IF ik$="" THEN 1090         [2682]

1100 IF ik$=CHR$(224) OR ik$=CHR$(88) THEN [2677]
 1110
1110 IF ik$=CHR$(240) OR ik$=CHR$(11) THEN [2151]
 by=by-2
1120 IF ik$=CHR$(241) OR ik$=CHR$(10) THEN [1955]
 by=by+2
1130 IF ik$=CHR$(242) OR ik$=CHR$(8) THEN  [1326]
bx=bx-2
1140 IF ik$=CHR$(243) OR ik$=CHR$(9) THEN  [1914]
bx=bx+2
1150 IF ik$=CHR$(69) THEN note$="MANN,  DU [3857]
  VERSAGER !":GOTO 3490
1160 EVERY 200 GOSUB 3440                   [810]
1170 IF bx=0 AND by=0 THEN 1090             [1538]
1180 LOCATE x,y:PRINT f$                    [795]
1190 x=x+bx:y=y+by                          [2539]
1200 IF xy(x,y)=0 THEN x=x-bx:y=y-by        [1690]
1210 LOCATE x,y:PRINT f$                    [795]
1220 GOTO 1080                              [387]
1230 REM ******************************     [1365]
1240 REM *        MUSIKDATAS          *      [1312]
1250 REM ******************************     [1365]
1260 RESTORE 1320                           [735]
1270 POKE &B939,&C9                         [483]
1280 FOR s=0 TO 83                          [879]
1290 READ t,l                               [499]
1300 ton(s)=t:lae(s)=l                      [1164]
```

Listing "Einstein"

```
1310 NEXT                                   [350]
1320 DATA 0,100,402,25,379,50,0,37.5,338,2 [2541]
5,319,50,0,37.5,402,25,379,37.5
1330 DATA 338,12.5,319,37.5,239,12.5,253,3 [3102]
7.5,379,12.5,319,37.5,253,12.5
1340 DATA 268,75,284,12.5,319,12.5,379,12. [3014]
5,426,12.5,379,100
1350 DATA 0,37.5,402,25,379,50,0,37.5,338, [3455]
25,319,50,0,37.5,402,25,379,37.5
1360 DATA 338,12.5,319,37.5,239,12.5,253,3 [3427]
7.5,319,12.5,253,37.5,190,12.5
1370 DATA 201,150,402,25,379,50,0,37.5,338 [2807]
,25,319,50,0,37.5,402,25,379,37.5
1380 DATA 338,12.5,319,37.5,239,12.5,253,3 [3102]
7.5,379,12.5,319,37.5,253,12.5
1390 DATA 268,75,284,12.5,319,12.5,379,12. [3014]
5,426,12.5,379,100
1400 DATA 0,50,190,37.5,213,12.5,253,37.5, [2672]
284,12.5,319,37.5,319,12.5
1410 DATA 268,37.5,284,12.5,268,37.5,284,1 [2663]
2.5,268,37.5,284,12.5
1420 DATA 268,37.5,284,12.5,319,37.5,379,1 [2497]
2.5,426,37.5,379,150
1430 DATA 0,50,319,37.5,379,12.5,426,37.5, [2169]
379,150
1440 REM ******************************     [1365]
1450 REM *     SYMBOLDEFINITION       *     [301]
1460 REM ******************************     [1365]
1470 RESTORE 1570                           [741]
1480 FOR symb=236 TO 255                    [1275]
1490 WHILE sy<8                             [970]
1500 READ sym                               [647]
1510 s(sy)=sym                              [651]
1520 sy=sy+1                                [316]
1530 WEND                                   [390]
1540 SYMBOL symb,s(0),s(1),s(2),s(3),s(4),  [2043]
s(5),s(6),s(7)
1550 sy=0                                   [169]
1560 NEXT                                   [350]
1570 DATA 243,8,7,244,24,16,32,33           [761]
1580 DATA 134,73,240,18,10,8,31,224         [1675]
1590 DATA 0,128,128,64,64,128,128,96        [1345]
1600 DATA 49,78,132,164,168,136,116,6       [1353]
1610 DATA 32,16,24,36,36,66,194,162         [1733]
1620 DATA 3,4,4,4,2,1,0,0                   [579]
1630 DATA 3,252,32,32,48,208,16,8           [367]
1640 DATA 62,16,16,8,4,4,2,1                [581]
1650 DATA 0,127,112,112,64,64,64,64         [1244]
1660 DATA 0,254,14,14,2,2,2,2               [1399]
1670 DATA 64,64,64,64,112,112,127,0         [874]
1680 DATA 2,2,2,2,14,14,254,0               [693]
1690 DATA 0,127,127,127,127,127,127,127     [1440]

1700 DATA 0,254,254,254,254,254,254,254     [1960]
1710 DATA 127,127,127,127,127,127,127,0     [827]
1720 DATA 254,254,254,254,254,254,254,0     [1434]
1730 DATA 0,0,0,7,15,28,26,25               [1475]
1740 DATA 0,0,0,224,240,56,88,152           [1908]
1750 DATA 25,26,28,15,7,0,0,0               [970]
1760 DATA 152,88,56,240,224,0,0,0           [1014]
1770 a$=CHR$(236)+CHR$(237)+CHR$(238)       [1208]

1780 b$=CHR$(239)+CHR$(32)+CHR$(240)        [1653]
1790 c$=CHR$(241)+CHR$(242)+CHR$(243)       [1573]

1800 d$=CHR$(244)+CHR$(245)+CHR$(10)+CHR$(  [3929]
8)+CHR$(8)+CHR$(246)+CHR$(247)
1810 e$=CHR$(248)+CHR$(249)+CHR$(10)+CHR$(  [3206]
8)+CHR$(8)+CHR$(250)+CHR$(251)
1820 f$=CHR$(252)+CHR$(253)+CHR$(10)+CHR$(  [2762]
8)+CHR$(8)+CHR$(254)+CHR$(255)
1830 g$=CHR$(236)+CHR$(237)+CHR$(238)+CHR$  [10773]
(10)+CHR$(8)+CHR$(8)+CHR$(8)+CHR$(239)+CHR
$(32)+CHR$(240)+CHR$(8)+CHR$(8)+CHR$(8)+CH
R$(10)+CHR$(241)+CHR$(242)+CHR$(243)
1840 RETURN                                 [555]
1850 REM ******************************     [1365]
1860 REM *  MULTI-SCRIPT (CPC 2/86)   *     [2070]
1870 REM ******************************     [1365]
1880 FOR a=1 TO LEN(w$)                     [1016]
1890 wz=ASC(MID$(w$,a,1)):IF wz=32 THEN 19  [2302]
60
1900 FOR b=1 TO 8:by(b)=PEEK(42239-1284+(w  [3810]
z-32)*8+b):NEXT
1910 FOR c=7 TO 0 STEP -1                   [992]
1920 FOR d=1 TO 8                           [1206]
1930 IF by(d)<2^c THEN 1950 ELSE by(d)=by(  [2382]
d)-2^c
```

Listing "Einstein"

```
1940 FOR y1=0 TO by STEP 2:PLOT xx+a*8*grx [5155]
-c*grx+d*s,yy-d*gry+y1,0:DRAWR bx,0:NEXT

1950 NEXT d,c                                [370]
1960 NEXT                                    [350]
1970 RETURN                                  [555]
1980 REM ******************************      [1365]
1990 REM *   DATAS FUER KOMMENTARE    *      [1537]
2000 REM ******************************      [1365]
2010 DIM text$(48)                           [1151]
2020 RESTORE 2340                            [754]
2030 POKE &B939,&C9                          [483]
2040 tt=-1                                   [54]
2050 FOR t=0 TO 44                           [810]
2060 tt=tt+1                                 [195]
2070 READ w$                                 [409]
2080 la=LEN(w$)                              [705]
2090 FOR x=la TO 1 STEP -1                   [1204]
2100 ch$=MID$(w$,x,1)                        [1998]
2110 ch=ASC(ch$)-5                           [1332]
2120 txt$=txt$+CHR$(ch)                      [896]
2130 NEXT                                    [350]
2140 text$(tt)=txt$                          [793]
2150 txt$=""                                 [718]
2160 NEXT                                    [350]
2170 FOR t=10 TO 19 STEP 2                   [1092]
2180 text$(t)=text$(t)+CHR$(13)+CHR$(10)+t [1380]
ext$(t+1)
2190 NEXT                                    [350]
2200 a=0                                     [324]
2210 FOR t=10 TO 14                          [988]
2220 text$(t)=text$(t+a)                     [630]
2230 a=a+1                                   [725]
2240 NEXT                                    [350]
2250 t=20                                    [382]
2260 text$(15)=text$(t)+CHR$(13)+CHR$(10)+ [6200]
text$(t+1)+CHR$(13)+CHR$(10)+text$(t+2)+CH
R$(13)+CHR$(10)+text$(t+3)
2270 text$(18)=text$(3):text$(3)=text$(10) [3298]
:text$(10)=text$(18)
2280 text$(18)=text$(4):text$(4)=text$(11) [2657]
:text$(11)=text$(18)
2290 FOR T=24 TO 44                          [1069]
2300 TEXT$(T-8)=TEXT$(T)                     [1073]
2310 NEXT                                    [350]
2320 POKE &B939,&F3                          [487]
2330 RETURN                                  [555]
2340 DATA "&%FI%FIJN\%XN%KSJX%1JYZJQ%TQQFM [2335]
"
2350 DATA "D%MHNS%1LSZMHXFWWJGJZ%JWWN%JS"    [2175]
2360 DATA "D%F\%QQTY1STXWJUYUZFM%JNI%WJNM% [2446]
SNG%MHN"
2370 DATA "&%FIJN\%MHN%SNG%FI%1PHZPHZP"      [1186]
2380 DATA "&%F\%QQTY%1KSJX%FIJN\%STMHX"      [1473]
2390 DATA "&%JXJTG%IWN\%NSWF%1XFI%XXFQ"      [2678]
2400 DATA "&%FIJN\%YRRTP%KSJX%1JYZJQ%IQFG% [2627]
XNG"
2410 DATA "&%MHXYJFGFQQJ%1MHNS%WFL%X,YGNL% [2756]
7%JPWJFYX"
2420 DATA "&LSNQLNJK"                        [141]
2430 DATA "&%SFW%JYZJM%WJGF%YXMJL%ZI%1SSFR [2727]
"
2440 DATA "WJ_YTQL%SJQQT%RJMHQJ\%YNR"        [1995]

2450 DATA "D%SZY%Z_%JYZJM%S,XJ%MHN%GFM"      [1476]

2460 DATA "SJYJGJL%SJRFS%SJSNJI%RZ%GFM%MHN [3070]
"
2470 DATA "&%JYWTXJXJFP%JS,RZ%MHNS%ISZ"      [2999]

2480 DATA "&%JWWN%FIJN\%QFR%FO%YJYKZI%TJI% [2378]
SNJI"
2490 DATA "D%YZFQPJL%KTMMHQNR%RNJG%XFI%JYX [2497]
FM"
2500 DATA "1UZMHYJP%WJRRN%YXSTX"             [869]
2510 DATA "&%KSJX%QFR%JYZJM%WJGF"            [1087]
2520 DATA "&%XJQQF%YLJFQMHX%QJNUX%XFI"       [1544]

2530 DATA "D%QJKKJTQ%JXXNJM%STMHX%JYXFM"     [1732]

2540 DATA "QJNUX%XFI%XXFI%1YWJTMJL%JIFWJL% [2591]
GFM%MHN"
2550 DATA "3YXXNJM%SNJYXSNJ"                 [1743]
2560 DATA "JYQF%JYZL%WJI%&%MHN%SSJP%SJI%1J [3252]
YZJQ"
2570 DATA "&%JNWTJMY[NYFQJW%SJQQT%WJSNJX%Y [3441]
NR%QYWJG"
2580 DATA "&%JQZMHXMHTM%JSNJP%WJGF%SFR%YMH [2254]
ZFWG%LZ_%SJI%WJZK"
```

Listing "Einstein"

```
2590 DATA "DF\1MHXNMHJNWL%XJQQTY%1YSZX%JFY [1793]
HFO%JFJQF"
2600 DATA "D%F\%1ISFX%RF%QMT\%JYZJM%YXMJYX [1600]
"
2610 DATA "&%JM%JM%1WJLSJFKSF%SJS,YLJFWU%L [2460]
Z_%WJI"
2620 DATA "&%KSJX%JGZY%JS%YSJNIWJ[%LZ_%WJI [2894]
"
2630 DATA "&%LSZYMHZJQWJ%RZ%YXNJL%3QM%RZ_% [2470]
QFR%MJQK"
2640 DATA "&%SJGNJWMHXGF%JYXSSFP%WJNG%JYSJ [2860]
NIWJ[%XFI"
2650 DATA "D%TYYT%YNR%YISF\WJ[%ZI%YXNG"     [1947]

2660 DATA "&%IQJK%JYMHJW%XFI%1UNY%WJXXNJM% [1717]
S"
2670 DATA "&%KZF%JX,SJGJL%1UNY%WJXXNJM%S"   [1683]

2680 DATA "&%YZFMHXJLGF%WNR%NJG%JYXFM%LZ_% [1562]
SJI"
2690 DATA "&%ZI%YXNG%SJIWJ%KZF%_YN\%JYXJG% [1862]
WJI"
2700 DATA "&%GF%TNIFW%RN%TSNJM%QFR%YXWJZ_% [2271]
QQJYX"
2710 DATA "&%QWJKUZMYYJG%XN%_YJO%1YFW%WJYZ [3647]
L,S"
2720 DATA "&%KTMSMFG%RN%MHNS%LZ_%WJI%WJGF% [4189]
XN%_YJO"
2730 DATA "%&WJRRFM%W%J%I%YMHJ%1_YN\%S,WJI [2541]
JN\%STMHX"
2740 DATA "D%YPHZLJL%XFQL%XSN%KJNY%Z_%WJIJ [4243]
N\%QFR"
2750 DATA "&%SJWJNQZYFWL%WNI%YXKWFI"        [2126]
2760 DATA "&%YXRRTP%YNJ\%TX%ZI%XXFI%1WJISZ [2436]
\,S"
2770 DATA "&%LZ_%WJISJ_YJF%YMHJF,S"          [1717]
2780 DATA "D%WTYFWJSJLXQQFKZ_%YNR%JNX%SJQJ [3598]
NUX"
2790 REM ******************************      [1365]
2800 REM *       SPIELSTAERKE         *      [835]
2810 REM ******************************      [1365]
2820 CLS:PEN 3:SPEED INK 50,50               [1237]
2830 LOCATE 19,5:PRINT g$                    [1466]
2840 LOCATE 16,12:PRINT text$(8);            [1289]
2850 WHILE INKEY$="":WEND:CLS:GOTO 460       [3053]
2860 CLS:PEN 3:SPEED INK 50,50               [1237]
2870 LOCATE 19,5:PRINT g$                    [1466]
2880 LOCATE 6,12:PRINT text$(7);             [894]
2890 WHILE INKEY$="":WEND:CLS:GOTO 460       [3053]
2900 CLS:PEN 3:SPEED INK 50,50               [1237]
2910 LOCATE 19,5:PRINT g$                    [1466]
2920 LOCATE 6,12:PRINT text$(9);             [1048]
2930 WHILE INKEY$="":WEND                    [1607]
2940 f1=1:CLS:GOTO 460                       [320]
2950 REM ******************************      [1365]
2960 REM *    XOR-TEXT (CPC 1/86)     *      [1305]
2970 REM ******************************      [1365]
2980 RESTORE 3040                            [743]
2990 FOR adr=&BDF4 TO &BE13:READ dat$:dat= [3843]
VAL("&"+dat$)
3000 POKE adr,dat:NEXT                       [525]
3010 POKE &B306,&F4:POKE &B307,&BD           [959]
3020 PRINT CHR$(22);CHR$(0):PRINT CHR$(23) [3842]
;CHR$(1):POKE &B28B,120
3030 RETURN                                  [555]
3040 DATA 21,01,BE,B7,28,03,21,0C            [803]
3050 DATA BE,22,91,B2,C9,2A,8F,B2            [1926]
3060 DATA 7D,AC,A1,AC,0E,FF,18,03            [1260]
3070 DATA 3A,8F,B2,47,EB,C3,E8,BD            [1449]
3080 REM ******************************      [1365]
3090 REM *   SPIELZUGUEBERPRUEFUNG    *      [1499]
3100 REM ******************************      [1365]
3110 IF flag=1 THEN 3140                     [694]
3120 IF xy(x,y)=1 THEN e=x:f=y:flag=1:FOR   [5360]
s=1 TO 15:SOUND 1,s,1,15:NEXT:GOTO 1080
3130 SOUND 1,379,15,15:flag=0:GOSUB 3730:G [3349]
OTO 1080
3140 IF xy(x,y)<>2 THEN 3130                 [1695]
3150 IF x<>e AND y<>f THEN 3130              [1950]
3160 a1=x-e:a2=y-f                           [691]
3170 IF ABS(a1)<>4 THEN 3180 ELSE 3190       [184]
3180 IF ABS(a2)<>4 THEN 3130                 [736]
3190 a1=a1*(-0.5):a2=a2*(-0.5)               [1819]
3200 IF xy(e-a1,f-a2)<>1 THEN 3130           [1328]
3210 REM ******************************      [1365]
3220 REM *     STEIN ELIMINIEREN      *      [655]
3230 REM ******************************      [1365]
3240 PEN 0:PRINT CHR$(23)CHR$(0)             [1996]
```

Listing "Einstein"


```
3250 LOCATE e,f:PRINT e$:xy(e,f)=2           [1759]
3260 LOCATE e-a1,f-a2:PRINT e$:xy(e-a1,f-a [3986]
2)=2
3270 LOCATE x,y:PRINT d$                     [687]
3280 steine=steine-1:IF steine=1 THEN 3820   [899]

3290 zug=zug+1:IF zug=5 THEN GOSUB 3370      [2912]
3300 LOCATE 35,16:PRINT steine;              [1133]
3310 GOSUB 2980                              [823]
3320 LOCATE x,y:PRINT f$:xy(x,y)=1           [1466]
3330 flag=0:GOTO 1080                        [1052]
3340 REM ******************************      [1365]
3350 REM *   KOMMENTARE VON SENF      *      [1257]
3360 REM ******************************      [1365]
3370 WINDOW 1,40,22,25:CLS:PRINT text$(INT [4946]
(RND*26+10)):BORDER bo:bo=bo+1:IF bo=26 TH
EN bo=0
3380 CALL &BB06:zug=0                        [803]
3390 WINDOW 1,40,1,25                        [1024]
3400 RETURN                                  [555]
3410 REM ******************************      [1365]
3420 REM *          ZEIT            *       [727]
3430 REM ******************************      [1365]
3440 zeit=zeit+5                             [1941]
3450 RETURN                                  [555]
3460 REM ******************************      [1365]
3470 REM *       ENDAUSWERTUNG      *       [2115]
3480 REM ******************************      [1365]
3490 PRINT CHR$(23)CHR$(0)                   [1823]
3500 MODE 1                                  [506]
3510 CLS                                     [91]
3520 LOCATE 16,3:PRINT"EIN STEIN"            [1846]
3530 LOCATE 16,4:PRINT STRING$(9,"-")        [1359]
3540 LOCATE 19,7:PRINT g$;                   [1322]
3550 t=0:BORDER 26,0:FOR s=100 TO 400:SOUN [4422]
D 1,s,1,15,1:NEXT
3560 LOCATE 15,12:PRINT"AUSWERTUNG!"         [1492]
```

Listing "Einstein"

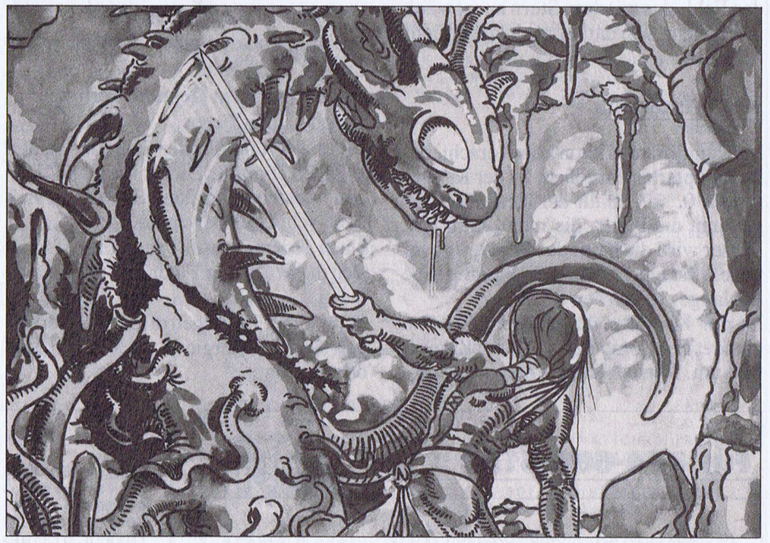
```
3570 LOCATE 15,13:PRINT STRING$(11,"=")      [2021]
3580 pkt=10000-steine*10-zeit                [1665]
3590 LOCATE 14,17:PRINT"PUNKTE :";pkt        [1804]
3600 LOCATE 10,19:PRINT note$;               [2414]
3610 LOCATE 6,22:PRINT"NOCH'N SPIEL MIT SE [2982]
NF (J/N) ?"
3620 ik$=INKEY$                              [822]
3630 SOUND 1,ton(t),lae(t),15,1              [1740]
3640 SOUND 2,ton(t),lae(t),15,1              [1746]
3650 SOUND 4,ton(t),lae(t),15,1              [2171]
3660 t=t+1:IF t=84 THEN t=0                  [448]
3670 IF ik$="J" THEN CLS:ERASE xy:GOTO 710 [1734]

3680 IF ik$="N" THEN CLS:LOCATE 3,12:PRINT [3547]
 text$(6):CALL &BB03:END
3690 GOTO 3620                               [676]
3700 REM ******************************      [1365]
3710 REM *      FALSCHE EINGABE       *      [2069]
3720 REM ******************************      [1365]
3730 PRINT CHR$(23)CHR$(0)                   [1823]
3740 WINDOW 1,40,21,25:CLS                   [1781]
3750 LOCATE 2,2:PRINT text$(5)               [1353]
3760 GOSUB 2980                              [823]
3770 WINDOW 1,40,1,25                        [1024]
3780 CALL &BB06:RETURN                       [1826]
3790 REM ******************************      [1365]
3800 REM *     AUFGABE ERFUELLT       *      [1481]
3810 REM ******************************      [1365]
3820 PRINT CHR$(23)CHR$(0):INK 3,0,26:PAPE [6598]
R 3:FOR s=50 TO 5 STEP-1:SPEED INK s,s+1:N
EXT:BORDER 2,25
3830 FOR s=100 TO 500:SOUND 1,s,1,,1,,10:N [6640]
EXT:FOR s=500 TO 100 STEP -2:SOUND 2,s,2,1
0,,,25:NEXT
3840 FOR s=5 TO 50 STEP 2:SPEED INK s,s+1: [3532]
NEXT:PAPER 1:BORDER 26
3850 note$="HAST'N BIER VERDIENT!":GOTO 34 [2106]
90
```

Listing "Einstein"

AUXILIA

Von Gnomen, Drachen und Druiden



AUXILIA ist ein Strategie- und Actionspiel für zwei Personen und dem erfolgreichen ARCHON angelehnt. Es gilt durch geschicktes Taktieren, entweder alle drei magischen Felder in der Mitte zu besetzen oder alle Figuren des Gegners zu besiegen.

Jeder Spieler kommt abwechselnd an die Reihe und kann eine Figur ziehen. Dazu bewegt er den kleinen Stern auf die Figur, die er bewegen will, drückt Feuer, bewegt den Stern auf das Feld, auf das die Figur ziehen soll und drückt abermals Feuer. Befindet sich auf diesem Feld eine Figur des Gegners, kommt es zum Kampf. Ist die Bewegung unmöglich, gibt der Computer einen Piepston von sich, und man muß eine neue Figur anwählen.

Die Figuren

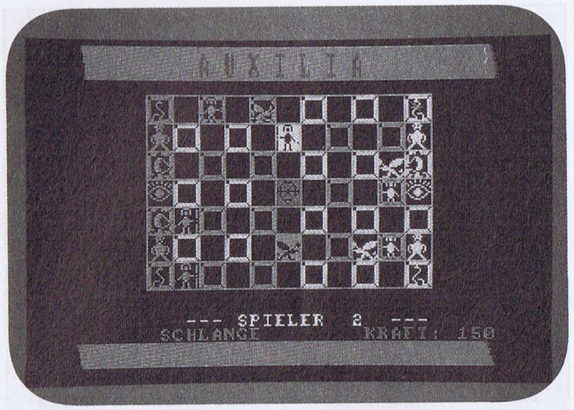
GNOM: kann sich bis zu zwei Felder weit bewegen, keine Figuren überspringen und (pro Zug) nur in eine Richtung gehen.

Schlange: wie GNOM, aber bis zu drei Feldern.

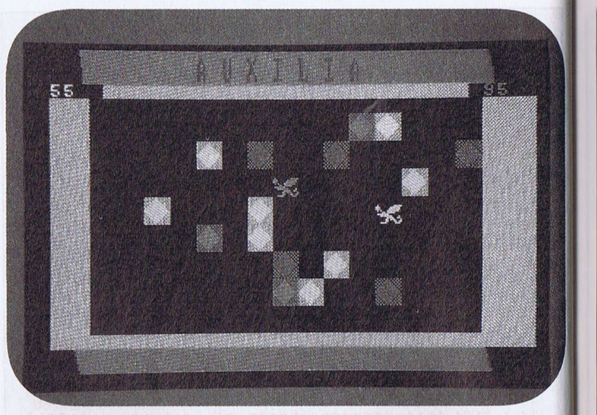
Zwerg: kann sich bis zu drei Felder weit bewegen, keine Figuren überspringen, aber einen "Knick" pro Zug machen (z.B. zwei Felder rechts, eins hoch, ähnlich dem Springer bei Schach).

Hexe: kann sich bis zu vier Felder weit bewegen, andere Figuren überspringen, in jede Richtung gehen (z.B. drei Felder links, vier Felder runter).

Drache: wie Hexe, aber nur zwei Felder.



Das Spielfeld von Auxilia. Wählen Sie ihre Spielfigur...



Magie und Zauberkraft - nur so können Sie Ihren Gegner besiegen.

Druide: kann nur Zaubersprüche einsetzen.

Zaubersprüche

Lähmen - eine Figur des Gegners (die mit dem Stern ausgewählt wurde) kann sich nicht bewegen bis sie geheilt wird.
Heilen - heilt eine (vorher gelähmte) eigene Figur.
Schaden - halbiert die Energie einer gegnerischen Figur.
Helfen - gibt einer eigenen Figur 100 Energieeinheiten zusätzliche Kraft.
Bewegen - teleportiert den Druiden auf das Feld, das mit dem Stern ausgewählt wird (wenn sich darauf noch keine Figur befindet).

Kampfsequenz

Im Kampfmodus kann man verschiedene Kampfarten einstellen. Ist der Kampfmodus 0, so entfällt der Actionteil. Es gewinnt die Figur, die mehr Kraft hat. Dieser wird die Kraft des Gegners abgezogen. Ist eine der beiden Figuren eine Schlange, so sterben beide.
Ist der Kampfmodus größer als 0, so blendet der Computer zu einem Kampffeld um. Je nach Kampfmodus erscheinen noch verschiedenfarbige Blöcke, die Energie dazugeben oder abziehen bzw. den Weg versperren.
Ist der Kampfmodus 4, so ist auch noch die Schuß- und Schrittanzahl begrenzt (werden neben der Energie oben angezeigt). Die Anzahl der Schritte und Schüsse kann durch Einsammeln des ab und zu auftauchenden dunklen Blocks erhöht werden.
Ist die Energie einer Figur auf Null gesunken, so ist diese verloren.
Die einzelnen Figuren haben im Kampf folgende charakteristische Merkmale:
Druide: kann sich gar nicht wehren (sich aber schneller bewegen),
Schlange: beide Spieler sterben,
Zwerg: wirft Axt,
Gnom: kann den Gegner "zertreten",
Drache: speit Feuer,
Hexe: wirft magischen Ball, der an den Wänden abprallt. Dieser kann mit "fangen" gestoppt werden (Hexe kann sich dann schneller bewegen).
Nach jedem Schuß oder Angriff muß sich die Waffe der Figur erst wieder aufladen.
Wird eine Figur "getroffen", so verringert sich die Energie (je nach Feind mehr oder weniger).

Die Steuerung

Wenn man zu Anfang SPACE drückt, können die Tasten umdefiniert werden. Bei Druck auf eine andere Taste werden die zuletzt definierten Tasten übernommen (normalerweise zwei Joysticks und SPACE bzw. die kleine ENTER-Taste zum Fangen).
Kann oder will ein Spieler nicht ziehen, kommt durch Drücken der DEL-Taste der nächste Spieler an die Reihe.
Hat ein Spieler alle drei magischen Felder besetzt, muß er sie noch einen Zug des Gegners lang verteidigen, dann ist das Spiel gewonnen.
Einen kleinen Wermutstropfen dürfen wir Ihnen allerdings nicht vorenthalten. Auxilia manipuliert den Videocontroller beim Bildaufbau, daher ist das Programm nicht auf allen CPC 464 lauffähig. Bei einer Produktionsserie dieser Geräte scheint etwas am Videochip geändert worden zu sein, einige Rechner vertragen diese Manipulation daher nicht. Wir werden dieser Sache jedoch nachgehen. Besitzer eines CPC 664 bzw. 6128 sind hiervon nicht betroffen.

(Andreas Lober/sr)

für 464-664-6128

```
10 'AUXILIA by KRABBE DESIGN                              [1524]
20 DEG:ENV 5,15,-1,1                                      [1220]
30 MODE 1:INK 0,0:INK 1,24:INK 3,6:INK 2,1                [3478]
4:PAPER 0:CLS
40 CLS:FOR i=1 TO 3:LOCATE 14-i,2:PEN i:PR                [7019]
INT">":LOCATE 27+i,2:PRINT"<":NEXT:LOCATE
15,2:PEN 1:PRINT"BITTE WARTEN"
50 DEFINT a,x,y:DIM a(40,25),f(11,7),s(11,                [5660]
7),k(11,7),p(11,7),s$(17,2),ci(80):spieler
=1:lx=1
60 SYMBOL AFTER 240                                       [1392]
70 re=REMAIN(1)                                           [906]
80 'Schriftrolle                                          [1329]
90 GOSUB 3870:FOR i=20 TO 390 STEP 2:PLOT                 [5860]
40+COS(i)*30,i,3:MOVER 2,0:DRAWR 500,0,1:P
LOTR 0,0,3:NEXT
100 LOCATE 14,2:PEN 3:PAPER 1:a$="AUXILIA"                [3685]
:GOSUB 310:ORIGIN 25,0
110 PEN 2:LOCATE 11,5:PRINT"VON KRABBE DES                [4445]
IGN":LOCATE 9,6:PRINT"GESCHRIEBEN MCMLXXXI
I"
120 PLOT 190,200,3:DRAW 194,204:DRAW 224,2                [11119]
04:DRAW 224,196:DRAW 196,196:DRAW 190,200:
PLOT 190,200:DRAW 224,200:PLOT 224,206:DRA
W 220,212:DRAW 198,212:DRAW 194,206:PLOT 2
02,196:DRAW 202,204:PLOT 212,204
```

Listing "AUXILIA"

```
130 DRAW 212,196:PLOT 228,180:DRAW 226,194                [10314]
:PLOT 226,206:DRAW 228,224:DRAW 228,186:PL
OT 230,210:DRAW 372,216:DRAW 390,204:DRAW
370,188:DRAW 230,194:PLOT 230,202:DRAW 384
,202:PLOT 412,158:DRAW 412,246
140 DRAW 158,246:DRAW 158,158:DRAW 410,158                [3859]
:GOSUB 3880:WHILE INKEY$<>"":WEND
150 PEN 0:LOCATE 5,22:PRINT"[SPACE]: TASTE                [6841]
N UMDEFINIEREN":LOCATE 6,23:PRINT"ANDERE T
ASTE FUER DEFINIERTE"
160 a$=INKEY$:IF a$=""THEN 160                            [1368]
170 LOCATE 6,23:PRINT SPACE$(29):LOCATE 5,                [2230]
22:PRINT SPACE$(28)
180 IF a$=" "THEN GOSUB 3650 ELSE IF b(5,0                [6703]
)=0 THEN b(1,0)=72:b(2,0)=73:b(4,0)=75:b(3
,0)=74:b(5,0)=76:b(1,1)=48:b(2,1)=49:b(3,1
)=50:b(4,1)=51:b(5,1)=53:b(6,0)=47:b(6,1)=
6
190 PEN 3:LOCATE 13,22:PRINT"KAMPFMODUS                   [2517]
 0 ":action=0
200 PEN 3:LOCATE 26,22                                    [909]
210 FOR t=1 TO 150:NEXT:IF INKEY(b(5,0))>-                [4803]
1 THEN 270 ELSE IF INKEY(b(1,0))=-1 AND IN
KEY(b(2,0))=-1 THEN 210
220 action=action+1:IF action=6 THEN actio                [2332]
n=0:GOTO 240
230 IF INKEY(b(2,0))>-1 THEN action=action                [4016]
-2:IF action=-1 THEN action=4
240 IF action=5 THEN action=0                             [1012]
250 PRINT action                                          [477]
260 GOTO 200                                              [429]
270 LOCATE 7,22:PEN 3:PRINT"DRUECKE FEUER                 [4209]
UM ANZUFANGEN"
280 anf=1                                                 [179]
290 GOTO 3430                                             [515]
300 'Schrift                                              [567]
310 IF spiel<2 THEN SYMBOL AFTER 32                       [1063]
320 matr=HIMEM+1:IF spiel<2 THEN GOSUB 160                [3297]
0
330 FOR a=1 TO LEN(a$)                                    [1008]
340 FOR b=0 TO 7                                          [956]
350 adr=PEEK(matr+(ASC(MID$(a$,a,1))-32)*8                [2145]
+b)
360 az(b)=adr:NEXT                                        [1324]
370 IF sy2+yp2*2<7 OR sy2+yp2*2>17 THEN yp                [2661]
1=-yp1
380 SYMBOL 254,az(0),az(0),az(1),az(1),az(                [2937]
2),az(2),az(3),az(3)
390 SYMBOL 255,az(4),az(4),az(5),az(5),az(                [3230]
6),az(6),az(7),az(7)
400 PRINT CHR$(254)CHR$(10)CHR$(8)CHR$(255                [3155]
)CHR$(11)" ";:NEXT:RETURN
410 'Spiel                                                [141]
420 FOR i=40 TO 0 STEP-1:OUT &BC00,1:OUT &                [6022]
BD00,i:FOR t=1 TO 2:NEXT:NEXT:IF anf=1 THE
N WINDOW#0,1,40,5,19:PAPER 0:CLS:WINDOW#0,
1,40,1,25
430 IF anf=1 THEN FOR i=1 TO 11:FOR j=1 TO                [2528]
 7:GOSUB 3540:NEXT:NEXT
440 IF anf=0 THEN i=x:j=y:GOSUB 3540:i=fx:                [5404]
j=fy:GOSUB 3540:i=dx:j=dy:GOSUB 3540
450 fx=0:fy=0:f=0:anf=0                                   [779]
460 LOCATE 13,21:PEN spieler:PRINT"--- SPI                [5020]
ELER "spieler" ---":x=1-10*(spieler=2):y=1
470 GOSUB 840                                             [844]
480 LOCATE 1,10:PRINT CHR$(23)CHR$(1):TAG:                [6382]
PLOT 128+x*32+8,496-(128+y*32+8),3:PRINT"*
";:TAGOFF
490 LOCATE 1,10:PRINT CHR$(23)CHR$(1):TAG:                [6382]
PLOT 128+x*32+8,496-(128+y*32+8),3:PRINT"*
";:TAGOFF
500 FOR t=1 TO 100:NEXT                                   [1125]
510 IF tod(3-spieler)=14 THEN 770                         [1216]
520 IF INKEY(b(1,spieler-1))>-1 AND y>1 TH                [2969]
EN y=y-1
530 IF INKEY(b(2,spieler-1))>-1 AND y<7 TH                [2683]
EN y=y+1
540 IF INKEY(b(3,spieler-1))>-1 AND x>1 TH                [1687]
EN x=x-1
550 IF INKEY(b(4,spieler-1))>-1 AND x<11 T                [2591]
HEN x=x+1
560 IF INKEY(79)>-1 THEN 790                              [751]
570 TAG:PLOT 128+x*32+8,496-(128+y*32+8),3                [2454]
:PRINT"*";:TAGOFF
580 IF dt=1 THEN IF yy=7 THEN OPENOUT"!aux                [11480]
ilia.sav":FOR i=1 TO 11:FOR j=1 TO 7:PRINT
#9,s(i,j),k(i,j),f(i,j):NEXT:NEXT:PRINT#9,
tod(1),tod(2),spieler,action:CLS:CLOSEOUT:
SYMBOL AFTER 32:END
590 IF dt=1 THEN IF yy=8 THEN OPENIN"!auxi                [10503]
lia.sav":FOR i=1 TO 11:FOR j=1 TO 7:INPUT#
9,s(i,j),k(i,j),f(i,j):NEXT:NEXT:INPUT#9,t
od(1),tod(2),spieler,action:CLOSEIN:anf=-(
action>0):DT=0:GOTO 410
600 LOCATE 11,22:PEN spieler+1:name$=name$                [4283]
(f(x,y)):k=k(x,y)
610 PRINT name$,"KRAFT:"k"    ":PEN spieler               [2241]
620 IF INKEY(b(5,spieler-1))=-1 THEN 770 E                [14126]
LSE IF f=0 AND dt=0 THEN IF s(x,y)<>spiele
r THEN PRINT CHR$(7);:GOTO 770 ELSE IF f(x
,y)=6 THEN GOSUB 3260:GOTO 770 ELSE fx=x:f
y=y:f=1:SOUND 1,900,10:WHILE INKEY(b(5,spi
eler-1))>-1:WEND:GOTO 770
630 dt2=0                                                 [188]
640 IF f(x,y)=6 THEN 710                                  [1105]
650 IF dt=1 AND k(dx,dy)>20 THEN IF yy=3 T                [8883]
HEN IF s(x,y)=spieler THEN k(x,y)=k(x,y)+1
00:k(dx,dy)=k(dx,dy)-20:dt2=1
660 IF dt=1 AND k(dx,dy)>20 THEN IF yy=4 T                [6145]
HEN IF s(x,y)=3-spieler THEN k(x,y)=k(x,y)
\2:k(dx,dy)=k(dx,dy)-20:dt2=1
670 IF dt=1 AND k(dx,dy)>10 THEN IF yy=5 T                [6885]
HEN IF s(x,y)=0 THEN s(x,y)=spieler:f(x,y)
=6:k(x,y)=k(dx,dy)-10:dt2=1
680 IF dt=1 AND k(dx,dy)>50 THEN IF yy=1 T                [9234]
HEN IF NOT(x=6 AND(y=2 OR y=4 OR y=6))THEN
 IF s(x,y)=3-spieler THEN s(x,y)=s(x,y)+10
:k(dx,dy)=k(dx,dy)-50:dt2=1
690 IF dt=1 AND k(dx,dy)>50 THEN IF yy=2 A                [4961]
ND s(x,y)>9 THEN s(x,y)=s(x,y)-10:k(dx,dy)
=k(dx,dy)-50:dt2=1
700 IF dt=1 THEN IF yy=5 THEN k(dx,dy)=0:s                [3530]
(dx,dy)=0:f(dx,dy)=0
710 IF dt2=0 AND dt=1 THEN PRINT CHR$(7):d                [2290]
t=0:GOTO 770
720 IF dt=1 THEN dt=0:GOTO 790                            [191]
730 IF x=fx AND y=fy THEN PRINT CHR$(7);:F                [5894]
X=0:FY=0:F=0:GOTO 770
740 s=spieler:c$=CHR$(7)                                  [1492]
750 f=0:GOSUB 2680                                        [1086]
760 IF w THEN IF(s(x,y)<>s OR f(x,y)=0)THE                [11386]
N IF s(x,y)=3-s OR s(x,y)=13-s THEN 1610 E
LSE k(x,y)=k(fx,fy):k(fx,fy)=0:f(x,y)=f(fx
,fy):f(fx,fy)=0:s(x,y)=s(fx,fy):s(fx,fy)=0
:GOTO 790 ELSE PRINT C$
770 IF (s(6,2)=spieler AND s(6,4)=spieler                 [10800]
AND s(6,6)=spieler) OR tod(3-spieler)>13 T
HEN GOSUB 3610:WHILE INKEY(b(5,0))=-1:WEND
:PAPER 0:MODE 1:PRINT CHR$(23)+CHR$(0):spi
el=2:tod(3-spieler)=0:spieler=1:INK 3,6:GO
TO 90
780 GOTO 490                                              [371]
790 spieler=3-spieler                                     [1186]
800 GOTO 420                                              [401]
810 'Spielfeld                                            [77]
820 LOCATE 6,22:PRINT STRING$(30,32):ORIGI                [2502]
N-16,-2
830 FOR i=1 TO 150 STEP 2:PLOT 0,200-i,0:D                [3207]
RAWR 640,0:PLOT 0,200+i:DRAWR 640,0:NEXT
840 PRINT CHR$(23)CHR$(0);:ORIGIN-16,-2:FO                [5547]
R i=7 TO 240 STEP 32:PLOT 160,106+i,1:DRAW
R 352,0:NEXT
850 FOR i=0 TO 40:OUT &BC00,1:OUT &BD00,i:                [3970]
FOR t=1 TO 2:NEXT:NEXT
860 FOR i=160 TO 516 STEP 32:PLOT i,112:DR                [3918]
AW i,336:NEXT:IF LEN(s$(0,1))=7 THEN RETUR
N
870 f=1:GOSUB 3870                                        [696]
880 'zeichen                                              [818]
890 FOR i=1 TO 6:s$(i,1)=CHR$(96+i*4)+CHR$                [5516]
(97+i*4)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(98+
i*4)+CHR$(99+i*4):NEXT
900 FOR i=1 TO 6:s$(i,2)=CHR$(150+i*4)+CHR                [6085]
$(149+i*4)+CHR$(10)+CHR$(8)+CHR$(8)+CHR$(1
52+i*4)+CHR$(151+i*4):NEXT
910 s$(0,1)="  "+CHR$(10)+CHR$(8)+CHR$(8)+               [3547]
"  ":s$(0,2)=s$(0,1):s$(0,0)=s$(0,1)
920 xy$=s$(2,2):s$(2,2)=s$(2,1):s$(2,1)=xy                [2739]
$
930 SYMBOL 100,0,0,0,0,0,248,126,127                      [1823]
940 SYMBOL 101,0,0,0,0,0,2,7,7                            [1360]
950 SYMBOL 102,63,15,0,67,103,142,216,112                 [2144]
960 SYMBOL 103,138,176,254,248,128,192,192                [2586]
,120
970 SYMBOL 104,0,3,68,69,69,64,65,3                       [1537]
980 SYMBOL 105,0,192,32,160,160,0,128,192                 [2401]
```

Listing "AUXILIA"

Listing "AUXILIA"

```
990 SYMBOL 106,95,3,67,65,66,66,66,70      [1608]
1000 SYMBOL 107,224,208,200,130,64,64,64,9 [2450]
6
1010 SYMBOL 108,0,0,0,0,1,2,2,1            [1779]
1020 SYMBOL 109,0,0,0,128,192,0,0,0        [1966]
1030 SYMBOL 110,0,0,0,7,8,8,7,0            [1775]
1040 SYMBOL 111,128,64,32,32,64,128,0,0    [1511]
1050 SYMBOL 112,2,3,6,3,2,1,3,15           [2045]
1060 SYMBOL 113,64,192,160,192,64,128,192, [2575]
248
1070 SYMBOL 114,23,23,19,11,3,6,12,24      [1622]
1080 SYMBOL 115,244,244,228,232,96,48,24,1 [2532]
2
1090 SYMBOL 116,0,1,3,0,1,3,7,7            [1379]
1100 SYMBOL 117,192,128,160,0,192,160,144, [2500]
8
1110 SYMBOL 118,7,7,7,3,0,25,30,24         [1471]
1120 SYMBOL 119,4,1,134,152,96,128,32,48   [1957]
1130 SYMBOL 120,0,2,18,72,35,140,17,35     [1732]
1140 SYMBOL 121,0,64,72,18,196,48,137,196  [1818]
1150 SYMBOL 122,199,35,17,12,3,0,0,0       [1990]
1160 SYMBOL 123,227,196,136,48,192,0,0,0   [2375]
1170 SYMBOL 124,1,2,7,8,20,50,81,255       [1555]
1180 SYMBOL 126,128,64,224,16,40,76,138,25 [2449]
5
1190 SYMBOL 125,81,50,20,8,7,2,1,0         [2082]
1200 SYMBOL 127,138,76,40,16,224,64,128,0  [1847]
1210 SYMBOL 33,0,24,36,90,90,36,24,0       [2175]
1220 SYMBOL 34,24,36,66,153,153,66,36,24   [2019]
1230 SYMBOL 35,48,112,240,248,28,14,6,0    [2377]
1240 SYMBOL 36,0,96,112,56,31,15,14,12     [2324]
1250 SYMBOL 37,165,189,86,126,106,60,60,60 [2474]
1260 SYMBOL 38,60,60,60,118,110,90,239,181 [1708]
1270 SYMBOL 39,192,120,239,31,247,223,120, [2690]
192
1280 SYMBOL 40,3,29,254,247,252,247,29,3   [1868]
1290 SYMBOL 128,138,176,254,240,128,224,24 [2136]
,7
1300 SYMBOL 129,95,3,3,1,2,4,8,8           [1484]
1310 SYMBOL 130,0,4,4,4,8,8,7,0            [1872]
1320 SYMBOL 131,23,23,19,11,3,6,6,12       [2494]
1330 SYMBOL 132,7,7,7,3,0,3,12,48          [1788]
1340 FOR i=1 TO 5:s$(i+10,0)=s$(i,0):s$(i+ [3572]
10,1)=s$(i,1):NEXT
1350 FOR i=1 TO 5:s$(i+10,1)=s$(i,1):s$(i+ [6809]
10,2)=s$(i,2):NEXT:RESTORE 1350:FOR i=1 TO
 5:READ p:MID$(s$(i+10,1),p,1)=CHR$(127+i)
:NEXT
1360 FOR i=1 TO 5:READ p:MID$(s$(i+10,2),p [4012]
,1)=CHR$(180+i):NEXT
1370 DATA 7,6,6,6,6,6,7,7,7,7              [576]
1380 GOSUB 2930                            [849]
1390 GOSUB 3880                            [977]
1400 RETURN                                [555]
1410 'feldbelegung                         [914]
1420 FOR i=1 TO 7                          [447]
1430 f(2,i)=1+(i/2-i\2)*2:k(2,i)=150:k(1,i [2534]
)=150:NEXT
1440 f(1,1)=3:f(1,2)=4:f(1,3)=5:f(1,4)=6:f [4078]
(1,5)=5:f(1,6)=4:f(1,7)=3
1450 FOR i=1 TO 7                          [447]
1460 f(10,i)=2-(i/2-i\2)*2:k(10,i)=150:k(1 [3493]
1,i)=150:NEXT
1470 f(11,1)=3:f(11,2)=4:f(11,3)=5:f(11,4) [3955]
=6:f(11,5)=5:f(11,6)=4:f(11,7)=3
1480 FOR i=1 TO 7:s(1,i)=1:s(2,i)=1:s(10,i [4032]
)=2:s(11,i)=2:FOR j=3 TO 9:s(j,i)=0:NEXT:N
EXT
1490 FOR i=1 TO 5 STEP 2:FOR j=1 TO 7 STEP [2831]
 2:p(i,j)=1
1500 NEXT:NEXT                             [1022]
1510 FOR i=7 TO 11 STEP 2:FOR j=1 TO 7 STE [3042]
P 2:p(i,j)=2
1520 NEXT:NEXT                             [1022]
1530 FOR i=2 TO 5 STEP 2:FOR j=2 TO 7 STEP [1964]
 2
1540 p(i,j)=2:NEXT:NEXT                    [627]
1550 FOR i=8 TO 10 STEP 2:FOR j=2 TO 7 STE [2779]
P 2
1560 p(i,j)=1:NEXT:NEXT                    [1597]
1570 RESTORE 1570:FOR i=0 TO 7:READ name$( [8700]
i),weite(i),st(i):NEXT:DATA "           ",0
,0,"DRACHE     ",2,250,"ZWERG     ",3,100,"
SCHLANGE",3,1,"GNOM      ",2,200,"HEXE
",4,190,"DRUIDE     ",0,150,,,,
1580 IF action=0 THEN FOR i=1 TO 7:FOR j=1 [3416]
 TO 11:k(j,i)=st(f(j,i)):NEXT:NEXT
1590 RETURN                                [555]
```

Listing "AUXILIA"

```
1600 SYMBOL AFTER 32:FOR n=HIMEM+1 TO HIME [8002]
M+((127-32)*8)STEP 8:FOR t=n TO n+7:POKE t
,PEEK(t)AND PEEK(t)+255:NEXT:NEXT:RETURN
1610 'kampf                                [102]
1620 re=REMAIN(1):spi=spieler:IF spieler=1 [9358]
 THEN sp(1)=f(fx,fy):sp(2)=f(x,y):en(1)=k(
fx,fy):en(2)=k(x,y)ELSE sp(2)=f(fx,fy):sp(
1)=f(x,y):en(1)=k(x,y):en(2)=k(fx,fy)
1630 en1=en(1):en2=en(2):abz3=0:abz4=0     [1443]
1640 sp2=sp(2):sp1=sp(1):sc(1)=10:sc(2)=10 [3668]
:sh(1)=50:sh(2)=50
1650 spec=-(action>2)                      [715]
1660 x$=CHR$(220)+CHR$(221)+CHR$(8)+CHR$(8 [4076]
)+CHR$(10)+CHR$(223)+CHR$(222)
1670 f(fx,fy)=0:s(fx,fy)=0:k(fx,fy)=0:altx [3079]
=x:alty=y
1680 IF action=0 THEN IF sp(1)=3 OR sp(2)= [2085]
3 THEN en(1)=0:en(2)=0
1690 IF action=0 THEN 1980                 [1197]
1700 WINDOW#0,1,40,5,21:PAPER 0:CLS:WINDOW [2170]
#0,1,40,1,25
1710 PRINT CHR$(22)+CHR$(0)                [1676]
1720 x(1)=3:y(1)=6:x(2)=16:y(2)=6          [1336]
1730 WINDOW#1,3,39,4,22:PAPER#1,2:PEN#1,1: [7493]
FOR i=1 TO 19:PRINT#1,STRING$(37,207);:NEX
T:WINDOW#1,6,35,5,21:PAPER#1,0:CLS#1
1740 ERASE a:DIM a(20,12)                  [571]
1750 IF action<2 THEN 1790                 [1045]
1760 IF spec=1 THEN FOR i=1 TO 5:xx=RND*14 [6579]
+3:yy=RND*7+3:LOCATE xx*2,yy*2:PAPER 3:PEN
 0:PRINT x$:a(xx,yy)=10:NEXT
1770 FOR i=1 TO 10:xx=RND*14+3:yy=RND*7+3: [9962]
LOCATE xx*2,yy*2:PAPER 2:PEN 3:PRINT x$:a(
xx,yy)=2:NEXT:FOR i=1 TO 5:xx=RND*14+3:yy=
RND*7+3:LOCATE xx*2,yy*2:PAPER 1:PRINT x$:
PAPER 0:a(xx,yy)=1:NEXT
1780 IF action=4 THEN EVERY 500,3 GOSUB 38 [2474]
50
1790 a(3,6)=0:a(16,6)=0                    [1204]
1800 LOCATE 2-33*(spi=2),4:PRINT INT(en(sp [6385]
i)):IF action=4 THEN LOCATE 6-22*(spi=2),4
:PRINT sc(spi)-1;sh(spi)
1810 IF en(spi)<1 THEN LOCATE 2-33*(spi=2) [4427]
,4:FOR i=1 TO 1000:NEXT:PRINT CHR$(24)"TOT
"CHR$(24):GOTO 1980
1820 en(spi)=en(spi)-0.5                   [1346]
1830 x1=x(spi):y1=y(spi):l=l(spi):sf1=sf(s [1767]
pi)
1840 en1=en(spi)                           [208]
1850 IF action<2 THEN 1880                 [1185]
1860 EI:DI                                 [444]
1870 EI:DI                                 [444]
1880 GOSUB 2070                            [869]
1890 IF l=4 THEN lx=2 ELSE IF l=8 THEN lx= [2446]
1
1900 PAPER 0:LOCATE x1*2,y1*2:PEN spi:PRIN [4120]
T s$(sp(spi)+fig,lx)
1910 IF sf1 THEN en1=en1-1:en2=en(3-spi):x [5958]
2=x(3-spi):y2=y(3-spi):ON sp(spi)GOSUB 218
0,2350,2520,2540,2560,2670:en(3-spi)=en2
1920 IF sf1=1 THEN IF sp(spi)=2 THEN en2=e [6879]
n(3-spi):GOSUB 2350:en(3-spi)=en2 ELSE IF
sp(spi)=4 THEN en2=en(3-spi):GOSUB 2560:en
(3-spi)=en2
1930 x(spi)=x1:y(spi)=y1:l(spi)=l:sf(spi)= [2075]
sf1
1940 en(spi)=en1                           [636]
1950 d=d+1:IF d=2 THEN d=0:IF fig=10 THEN  [4240]
fig=0 ELSE fig=10
1960 IF sp(spi)=6 OR(sp(spi)=5 AND sf1=0)T [9187]
HEN GOSUB 2070:LOCATE x1*2,y1*2:PEN spi:PR
INT s$(sp(spi)+fig,lx):x(spi)=x1:y(spi)=y1
:l(spi)=l:sf(spi)=sf1
1970 spi=3-spi:GOTO 1800                   [1581]
1980 x=altx:y=alty:IF f(x,y)=sp(1)THEN dra [3263]
n=1 ELSE dran=2
1990 IF en(1)>en(2) THEN en(1)=en(1)-en(2) [5096]
:en(2)=0 ELSE en(2)=en(2)-en(1):en(1)=0
2000 en(1)=INT(en(1)):en(2)=INT(en(2))     [1561]
2010 IF en(1)<1 AND en(2)>=1 THEN s(x,y)=2 [10717]
:f(x,y)=sp2:k(x,y)=en(2):tod(1)=tod(1)+1 E
LSE IF en(2)<1 AND en(1)>=1 THEN s(x,y)=1:
f(x,y)=sp1:k(x,y)=en(1):tod(2)=tod(2)+1
2020 IF en(1)<1 AND en(2)<1 THEN f(x,y)=0: [3475]
s(x,y)=0:k(x,y)=0:tod(1)=tod(1)+1:tod(2)=t
od(2)+1
2030 sf(1)=0:sf(2)=0                       [835]
2040 r=REMAIN(3):IF action>0 THEN WINDOW#0 [4959]
```

Listing "AUXILIA"


```
,1,40,4,22:PAPER 0:CLS:WINDOW#0,1,40,1,25
2050 anf=-(action>0)                       [788]
2060 GOTO 790                              [415]
2070 abz1=(a(x(spi),y(spi))=1):en1=en1-10* [13929]
abz1:abz2=(a(x(spi),y(spi))=2):en1=en1+30*
abz2:abz3=(a(x(spi),y(spi))=9):sh(spi)=sh(
spi)-20*abz3:sc(spi)=sc(spi)-5*abz3:SOUND
spi,-500*abz3,10,5
2080 IF abz1=-1 THEN SOUND spi,en(spi)*3,1 [3373]
0,5 ELSE IF abz2=-1 THEN SOUND spi,en(spi)
*3,10,5,,,30
2090 a(x(spi),y(spi))=0                    [641]
2100 IF sh(spi)=0 THEN 2140 ELSE a1x=x1:a1 [12141]
y=y1:PEN spi:IF INKEY(b(1,spi-1))>-1 THEN
l=1:IF y1>3 THEN IF a(x1,y1-1)<10 THEN IF
NOT(x1=x(3-spi) AND y1-1=y(3-spi) AND sp(s
pi)<>4)THEN LOCATE x1*2,y1*2:PRINT s$(0,1)
:y1=y1-1
2110 IF INKEY(b(2,spi-1))>-1 THEN l=2:IF y [8958]
1<10 THEN IF a(x1,y1+1)<10 THEN IF NOT(x1=
x(3-spi) AND y1+1=y(3-spi) AND sp(spi)<>4)
THEN LOCATE x1*2,y1*2:PRINT s$(0,1);:y1=y1
+1
2120 IF INKEY(b(3,spi-1))>-1 THEN l=4:IF x [8767]
1>3 THEN IF a(x1-1,y1)<10 THEN IF NOT(x1-1
=x(3-spi) AND y1=y(3-spi) AND sp(spi)<>4)T
HEN LOCATE x1*2,y1*2:PRINT s$(0,1):x1=x1-1
2130 IF INKEY(b(4,spi-1))>-1 THEN l=8:IF x [8502]
1<17 THEN IF a(x1+1,y1)<10 THEN IF NOT(x1+
1=x(3-spi) AND y1=y(3-spi) AND sp(spi)<>4)
THEN LOCATE x1*2,y1*2:PRINT s$(0,1):x1=x1+
1
2140 h(spi)=h(spi)+1-sf1                   [1391]
2150 IF INKEY(b(5,spi-1))>-1 THEN IF sf1=0 [9644]
 THEN IF h(spi)>4 THEN IF sc(spi)>1 THEN s
f1=1:sx(spi)=x1:sy(spi)=y1:xp(spi)=1:yp(sp
i)=1:h(spi)=0:sc(spi)=sc(spi)+(action=4)
2160 sh(spi)=sh(spi)+(action=4 AND (a1x<>x [3257]
1 OR a1y<>y1)AND sh(spi)>0)
2170 RETURN                                [555]
2180 PEN 1:FOR i=1 TO 5:SOUND spi,500-i*10 [7746]
,5,i,,,i:NEXT:SOUND spi,570,20,7,,,8:IF l>
1 THEN 2230 ELSE IF y1>3 THEN LOCATE x1*2,
y1*2-1:PRINT"%%":LOCATE x1*2,y1*2-2:PRINT"
%%":CALL &BD19
2190 a(x1,y1-1)=0                          [645]
2200 IF y1>3 THEN  LOCATE x1*2,y1*2-1:PRIN [4562]
T"  ":LOCATE x1*2,y1*2-2:PRINT"  ":CALL &B
D19
2210 IF x1=x2 AND (y1=y2 OR y1-1=y2 OR y1- [3235]
2=y2)THEN en2=en2-50
2220 sf1=0:RETURN                          [607]
2230 IF l>2 THEN 2270 ELSE IF y1<10 THEN L [5600]
OCATE x1*2,y1*2+2:PRINT"&&":LOCATE x1*2,y1
*2+3:PRINT"&&":CALL &BD19
2240 a(x1,y1+2)=0                          [653]
2250 IF x2=x1 AND (y2=y1 OR y2-1=y1 OR y2- [2742]
2=y1)THEN en2=en2-50
2260 IF y1<10 THEN LOCATE x1*2,y1*2+2:PRIN [5741]
T"  ":LOCATE x1*2,y1*2+3:PRINT"  ":sf1=0:R
ETURN ELSE sf1=0:RETURN
2270 IF l>4 THEN 2310 ELSE IF x1>3 THEN LO [5953]
CATE x1*2-2,y1*2:PRINT"''":LOCATE x1*2-2,y
1*2+1:PRINT"''":CALL &BD19
2280 a(x1-1,y1)=0                          [536]
2290 IF (x2=x1 OR x2+1=x1 OR x2+2=x1)AND y [2407]
2=y1 THEN en2=en2-50
2300 IF x1>3 THEN LOCATE x1*2-2,y1*2:PRINT [7970]
"  ":LOCATE x1*2-2,y1*2+1:PRINT"  ":sf1=0:
RETURN ELSE sf1=0:RETURN
2310 IF l>8 THEN RETURN ELSE IF x1<17 THEN [7932]
 LOCATE x1*2+2,y1*2:PRINT"((":LOCATE x1*2+
2,y1*2+1:PRINT"((":CALL &BD19
2320 a(x1+2,y1)=0                          [749]
2330 IF (x2=x1 OR x2-1=x1 OR x2-2=x1)AND y [2312]
2=y1 THEN en2=en2-50
2340 IF x1<17 THEN LOCATE x1*2+2,y1*2:PRIN [7075]
T"  ":LOCATE x1*2+2,y1*2+1:PRINT"  ":LOCATE
 x1*2+2,y1*2:PRINT"  ":sf1=0:RETURN ELSE s
f1=0:RETURN
2350 ende=0:IF sx(spi)=x1 AND sy(spi)=y1 T [9097]
HEN sz1=0:FOR i=7 TO 0 STEP-1:SOUND spi,20
0,2,1:NEXT:lx(spi)=1:IF l=8 THEN sx(spi)=s
x(spi)+1 ELSE IF l=2 THEN sy(spi)=sy(spi)+
1
2360 IF sx1<3 OR sx1>17 OR sy1<3 OR sy1>10 [2198]
 OR ende=1 THEN sf1=0:RETURN
2370 lxx=lx(spi):o=o(spi):sx1=sx(spi):sy1= [2064]
```

Listing "AUXILIA"

```
sy(spi)
2380 pa=a(sx1,sy1):pe=0:IF pa>0 THEN pe=3  [2246]
2390 IF spec=0 THEN 2420                   [650]
2400 IF pa=10 THEN pa=3:pe=0               [761]
2410 IF pa=9 THEN pa=0:pe=3                [1167]
2420 PAPER pa:PEN pe:LOCATE sx1*2,sy1*2:PR [3821]
INT x$:PAPER 0
2430 IF lxx=1 THEN sy1=sy1-1:ym=-1:xm=0    [743]
2440 IF lxx=2 THEN sy1=sy1+1:ym=1:xm=0     [1296]
2450 IF lxx=4 THEN sx1=sx1-1:mx=-1:ym=0    [1465]
2460 IF lxx=8 THEN sx1=sx1+1:mx=1:ym=0     [2438]
2470 IF sx1<3 OR sx1>17 OR sy1<3 OR sy1>10 [2198]
 OR ende=1 THEN sf1=0:RETURN
2480 LOCATE sx1*2,sy1*2:PEN 1:IF o=1 THEN  [3716]
PRINT"#":o=2 ELSE PRINT"$":o=1
2490 o(spi)=o:lx(spi)=lxx:sx(spi)=sx1:sy(s [2347]
pi)=sy1
2500 IF sx1=x2 AND sy1=y2 THEN en2=en2-30: [2737]
lxx=0:ende=1:GOTO 2380
2510 RETURN                                [555]
2520 en1=0:en2=0:FOR i=3000 TO 3300 STEP 1 [4268]
0:FOR j=7 TO 0 STEP-1:SOUND spi,i,1,j,,,1:
NEXT:NEXT:sf1=0
2530 FOR j=1 TO 2:FOR i=0 TO 16 STEP 2:PLO [6389]
T x(j)*32-16+i,304-y(j)*16+i,0:DRAWR i,0:D
RAWR 0,i:DRAWR 2*(-i),0:DRAWR 0,2*(-i):DRA
WR i*2,0:DRAWR 0,i:NEXT i,j:RETURN
2540 sf1=0:SOUND spi,3500,50,7:IF x1=x2 AN [10130]
D y1=y2 THEN en2=0:OUT &BC00,8:OUT &BD00,1
:FOR i=1 TO 400:NEXT:FOR i=3300 TO 3500:SO
UND spi,i,1,7:NEXT:OUT &BC00,8:OUT &BD00,0
2550 RETURN                                [555]
2560 IF sx1=x1 AND sy1=y1 THEN sz(spi)=0:x [2745]
p1=1:yp1=1
2570 sx1=sx(spi):sy1=sy(spi):p=pp(spi):xp1 [3147]
=xp(spi):yp1=yp(spi)
2580 SOUND spi,100+RND*50,5,6:sz(spi)=sz(s [2374]
pi)+1
2590 IF sx1+xp1*2<2 OR sx1+xp1*2>18 THEN x [2770]
p1=-xp1
2600 IF sy1+yp1*2<2 OR sy1+yp1*2>11 THEN y [1588]
p1=-yp1
2610 pa=a(sx1,sy1):pe=0:IF pa>0 THEN pe=3: [6104]
IF pa=10 THEN pa=3:pe=0 ELSE IF pa=9 THEN
pa=0:pe=3
2620 PAPER pa:PEN pe:LOCATE sx1*2,sy1*2:PR [7506]
INT x$:PAPER 0:IF sz(spi)=25 OR INKEY(b(6,
spi-1))=0 THEN sf1=0:RETURN
2630 sy1=sy1+yp1:sx1=sx1+xp1               [613]
2640 PEN spi:LOCATE sx1*2,sy1*2:IF p=1 THE [4921]
N PRINT"!":p=2 ELSE PRINT CHR$(34):p=1
2650 IF sx1=x2 AND sy1=y2 THEN en2=en2/2:s [2987]
z(spi)=25:GOTO 2620
2660 sx(spi)=sx1:sy(spi)=sy1:pp(spi)=p:xp( [3848]
spi)=xp1:yp(spi)=yp1
2670 RETURN                                [555]
2680 'zug gueltig?                         [1525]
2690 w=0                                   [363]
2700 weite=weite(f(fx,fy))                 [1416]
2710 g=f(fx,fy):IF g>1 AND g<5 THEN 2760   [2982]
2720 IF x=fx THEN 2740                     [1291]
2730 FOR i=1 TO weite:IF x+i<>fx AND x-i<> [3132]
fx THEN NEXT:RETURN
2740 IF y=fy THEN IF fx=x THEN RETURN ELSE [3512]
 w=-1:RETURN
2750 FOR i=1 TO weite:IF y+i=fy OR y-i=fy  [5523]
THEN w=-1:RETURN ELSE NEXT:RETURN
2760 IF(ABS(fx-x)<=weite AND fy=y)OR(ABS(f [3783]
y-y)<=weite AND fx=x)THEN w=-1
2770 IF y=fy THEN FOR i=fx+1 TO x-1:IF f(i [6128]
,y)>0 THEN w=0 ELSE NEXT:FOR i=x+1 TO fx-1
:IF f(i,y)>0 THEN w=0 ELSE NEXT
2780 IF x=fx THEN FOR i=fy+1 TO y-1:IF f(x [6387]
,i)>0 THEN w=0 ELSE NEXT:FOR i=y+1 TO fy-1
:IF f(x,i)>0 THEN w=0 ELSE NEXT
2790 IF g>2 OR w THEN RETURN               [887]
2800 IF fx-x=0 OR fy-y=0 THEN RETURN       [2628]
2810 IF ABS(fx-x)+ABS(fy-y)<=weite THEN w= [1523]
-1
2820 zx=x:zy=y:ABSt=ABS(fx-x)+ABS(fy-y)    [3435]
2830 IF ABSt=3 THEN 2870                   [377]
2840 ein=-(x>fx)+(x<fx):zw=-(y>fy)+(y<fy)  [1319]
2850 IF f(fx+ein,fy)>1 AND f(fx,fy+zw)>1 T [2749]
HEN wahr=0
2860 RETURN                                [555]
2870 IF x<fx THEN zx=-zx                   [2086]
2880 IF y<fy THEN zy=-zy                   [1409]
2890 IF zx<0 THEN xp=-1 ELSE xp=1          [1513]
```

Listing "AUXILIA"

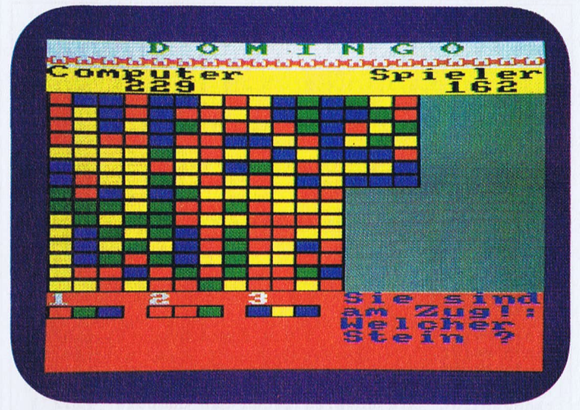
```
2900 IF zy<0 THEN xy=-1 ELSE xp=1                [1436]
2910 IF(f(fx+xp,fy)=1 OR f(fx+xp,fy+yp)=1) [2576]
AND(f(fx,fy+yp)=1 OR f(fx,fy+yp)=1)THEN w=
0
2920 RETURN                                      [555]
2930 'zeichen [2]                                [1207]
2940 SYMBOL 149,4,8,16,0,0,0,0,0                 [1694]
2950 SYMBOL 153,0,0,0,0,0,31,126,254             [1927]
2960 SYMBOL 154,0,0,0,0,0,64,224,224             [2180]
2970 SYMBOL 155,252,240,0,194,230,113,27,1 [2654]
4
2980 SYMBOL 156,81,13,127,31,1,3,3,30            [2300]
2990 SYMBOL 157,0,192,34,162,162,2,130,192 [1887]
3000 SYMBOL 158,0,3,4,5,5,0,1,3                  [1719]
3010 SYMBOL 159,250,192,194,130,66,66,66,9 [1917]
8
3020 SYMBOL 160,7,11,19,65,2,2,2,6               [1473]
3030 SYMBOL 161,0,0,0,0,128,64,64,128            [2021]
3040 SYMBOL 162,0,0,0,1,3,0,0,0                  [1520]
3050 SYMBOL 163,0,0,0,224,16,16,224,0            [1701]
3060 SYMBOL 164,1,2,4,4,2,1,0,0                  [1804]
3070 SYMBOL 165,64,192,96,192,64,128,192,2 [2282]
40
3080 SYMBOL 166,2,3,5,3,2,1,3,31                 [1637]
3090 SYMBOL 167,232,232,200,208,192,96,48, [2280]
24
3100 SYMBOL 168,47,47,39,23,6,12,24,48           [1948]
3110 SYMBOL 169,0,128,192,0,128,192,224,22 [2134]
4
3120 SYMBOL 170,3,1,5,0,3,5,9,16                 [1935]
3130 SYMBOL 171,224,224,224,192,0,152,120, [2259]
24
3140 SYMBOL 172,32,128,97,25,6,1,4,12            [1936]
3150 SYMBOL 173,0,64,72,18,196,49,136,196  [2114]
3160 SYMBOL 174,0,2,18,72,35,12,145,35           [1914]
3170 SYMBOL 175,227,196,136,48,192,0,0,0         [2334]
3180 SYMBOL 176,199,35,17,12,3,0,0,0             [1809]
3190 SYMBOL 181,81,13,127,15,1,7,24,224          [2168]
3200 SYMBOL 182,250,192,192,128,64,32,16,1 [2230]
6
3210 SYMBOL 183,0,32,32,32,16,16,224,0           [1541]
3220 SYMBOL 184,232,232,200,208,192,96,96, [2315]
48
3230 SYMBOL 185,224,224,224,192,0,192,48,1 [2513]
2
3240 RETURN                                      [555]
3250 'druide                                     [1339]
3260 xx=1+(spieler-1)*32:dx=x:dy=y               [2661]
3270 WINDOW#1,xx,xx+7,6,16                       [2008]
3280 PAPER#1,spieler+1:PEN#1,spieler-1:CLS [3438]
#1
3290 PRINT#1:PRINT#1,"LAEHMEN":PRINT#1,"HE [7368]
ILEN":PRINT#1,"HELFEN":PRINT#1,"SCHADEN":P
RINT#1,"BEWEGEN":PRINT#1,"ZURUECK":PRINT#1
,"SAVE":PRINT#1,"LOAD"
3300 dt=1:yy=1                                   [276]
3310 LOCATE#1,8,yy+1:PRINT#1,"*";                [2163]
3320 IF INKEY(b(1,spieler-1))>-1 THEN j=1  [5749]
ELSE IF INKEY(b(2,spieler-1))>-1 THEN j=2
ELSE IF INKEY(b(5,spieler-1))>-1 THEN j=16
 ELSE j=0
3330 FOR i=1 TO 100:NEXT                         [936]
3340 IF j=1 AND yy>1 THEN yy=yy-1:LOCATE#1 [5959]
,8,yy+2:PRINT#1," ";:LOCATE#1,8,yy+1:PRINT
#1,"*";
3350 IF j=2 AND yy<8 THEN LOCATE#1,8,yy+1: [5849]
PRINT#1," ";:yy=yy+1:LOCATE#1,8,yy+1:PRINT
#1,"*";
3360 IF j>15 THEN PAPER#1,0:CLS#1:IF yy=6  [3481]
THEN dt=0:RETURN ELSE RETURN
3370 GOTO 3320                                   [656]
3380 'musik                                      [1087]
3390 tonh=15                                     [694]
3400 xs(1)=30:xs(4)=30:xs(5)=30:xs(7)=1          [2957]
3410 FOR n=0 TO 7:is=-(xs(n)>0):SOUND 4,(2 [4868]
+xu)*is,26,tonh*is,5*is,,xs(n)
3420 NEXT:RETURN                                 [940]
3430 re=REMAIN(0):EVERY 10,1 GOSUB 3750:EN [2655]
V 1,2,-1,0,4,-1,1
3440 ENT 1,4,-10,1                               [1063]
3450 ENV 3,15,-1,2                               [679]
3460 ENV 4,15,-1,1                               [637]
3470 FOR i=1 TO 5:SOUND 1,300-20*i,30+5*(i [5504]
<5)+5*(i<4),5+i*2,1,1:SOUND 1,300,30+5*(i<
5)+5*(i<4),5+i*2,1,1:NEXT:RESTORE 3480
3480 READ typ:IF typ=-1 THEN 3510                [1096]
3490 IF typ=0 THEN READ d:SOUND 1,0,d,0 EL [8277]
```

Listing "AUXILIA"

```
SE IF typ=1 THEN SOUND 1,1,20,15,3,3,31 EL
SE IF typ=2 THEN SOUND 1,4,17,15,4,4,31 EL
SE IF typ=3 THEN SOUND 1,2,30,15,3,3,28 EL
SE SOUND 1,3,30,15,3,3,23
3500 GOTO 3480:DATA 0,50,10,10,2,2,0,3,1,0 [4015]
,10,1,0,30,1,0,10,1,0,10,1,2,2,1,0,10,1,0,
30,3,3,4,3,-1
3510 FOR i=5 TO 1 STEP-1:SOUND 1,300-20*i, [5276]
30+5*(i<5)+5*(i<4),5+i*2,1,1:SOUND 1,300,3
0+5*(i<5)+5*(i<4),5+i*2,1,1:NEXT
3520 re=REMAIN(1):RESTORE:GOSUB 820:GOSUB  [2521]
1420:GOTO 420
3530 'spielfeld aendern                          [1002]
3540 sxd=0:sx=s(i,j):IF sx>9 THEN sx=sx-10 [3186]
:sxd=3-sx
3550 IF i*j=0 THEN 3590                          [836]
3560 PEN sx:PAPER sxd:LOCATE 8+i*2,3+j*2:P [3340]
RINT s$(f(i,j),sx)
3570 IF(j=4 OR j=2 OR j=6)AND i=6 THEN PAP [12825]
ER 3:LOCATE 8+i*2,3+j*2:PRINT CHR$(124)CHR
$(126)CHR$(8)CHR$(8)CHR$(10)CHR$(125)CHR$(
127):IF f(i,j)>0 THEN PAPER s(i,j):PEN 0:L
OCATE 8+i*2,3+j*2:PRINT s$(f(i,j),s(i,j))
3580 PRINT CHR$(23)+CHR$(1):PLOT 130+i*32, [10486]
496-(128+j*32+2),p(i,j):DRAWR 28,0:DRAWR 0
,-28:DRAWR-28,0:DRAWR 0,26:DRAWR 26,0:DRAW
R 0,-24:DRAWR-24,0:DRAWR 0,26
3590 PAPER 0:RETURN                              [1143]
3600 'sieg                                       [286]
3610 m=10:WINDOW#0,1,40,5,23:PAPER 0:CLS:W [11751]
INDOW#0,1,40,1,25:a$="SPIELER"+STR$(spiele
r):GOSUB 3620:a$="HAT":GOSUB 3620:a$="GEWO
NNEN":GOSUB 3620:GOSUB 3780:RETURN
3620 m=m+2:FOR a=1 TO LEN(a$):FOR b=0 TO 7 [11326]
:adr=PEEK(matr+(ASC(MID$(a$,a,1))-32)*8+b)
:az(b)=adr:NEXT:SYMBOL 253,az(0),az(1),0,0
,0,0,0,0:SYMBOL 254,0,0,az(2),az(3),az(4),
0,0,0:SYMBOL 255,0,0,0,0,0,az(5),az(6),az(
7)
3630 LOCATE 19-LEN(a$)\2+a,m:PRINT CHR$(22 [8532]
)CHR$(1)CHR$(15)CHR$(1)CHR$(253)CHR$(8)CHR
$(15)CHR$(2)CHR$(254)CHR$(8)CHR$(15)CHR$(3
)CHR$(255)CHR$(22)CHR$(0):NEXT:RETURN
3640 'tasten                                     [271]
3650 t$(1)="HOCH   ":t$(2)="RUNTER":t$(4)=" [4723]
RECHTS":t$(3)="LINKS ":t$(5)="FEUER ":t$(6
)="FANGEN"
3660 FOR iii=0 TO 79:ci(iii)=iii:NEXT            [2441]
3670 FOR i=0 TO 1                                [468]
3680 FOR ii=1 TO 6                               [583]
3690 LOCATE 13,22:PRINT"SPIELER";i+1;t$(ii [2686]
)
3700 b(ii,i)=-1                                  [637]
3710 FOR iii=0 TO 79                             [640]
3720 a=INKEY(ci(iii)):IF a=0 THEN b(ii,i)= [3342]
iii
3730 NEXT:IF b(ii,i)=-1 THEN 3710                [1481]
3740 WHILE INKEY$<>"":WEND:ci(b(ii,i))=66: [3045]
NEXT:NEXT:RETURN
3750 IF INKEY(b(5,0))>-1 THEN 3520               [1085]
3760 RETURN                                      [555]
3770 'musik 2                                    [1100]
3780 ENV 1,15,-1,3:h=1:h2=1                      [984]
3790 x=0:t(1)=239:t(2)=239:t(3)=253:t(4)=2 [3268]
53
3800 READ t                                      [405]
3810 IF t=-1 THEN RESTORE 3800:h=h+0.5:IF  [6590]
h>2.5 THEN h=1:h2=h2+0.3:GOTO 3800 ELSE GO
TO 3800
3820 SOUND 4,t*h,18,15,1:SOUND 2,(t*h)/1.5 [4049]
,18,15,1:SOUND 6,0,2,0:x=x+1:IF x=5 THEN x
=1
3830 SOUND 1,t(x)*h2,18,15,1:SOUND 1,0,2,0 [5147]
:IF INKEY(b(5,0))>-1 THEN RETURN ELSE 3800
3840 DATA 319,358,379,60,60,319,379,60,358 [2381]
,319,60,60,60,379,-1
3850 xx=RND*14+3:yy=RND*7+3:LOCATE xx*2,yy [4279]
*2:PAPER 0:PEN 3:PRINT x$:a(xx,yy)=9:RETUR
N
3860 'out's                                      [403]
3870 i=30:FOR ii=25 TO 2 STEP-2:OUT &BC00, [7940]
7:OUT &BD00,i:i=i-1:OUT &BC00,6:OUT &BD00,
ii:CALL &BD19:NEXT:RETURN
3880 i=18:FOR ii=2 TO 25 STEP 2:OUT &BC00, [8484]
7:OUT &BD00,i:i=i+1:OUT &BC00,6:OUT &BD00,
ii:CALL &BD19:NEXT:OUT &BC00,6:OUT &BD00,2
5:RETURN
```

Listing "AUXILIA"

DOMINGO



Domingo erfordert schnelle Kombinationsgabe beim Setzen der Steine.

Im Spiel gegen den Computer müssen Sie Ihre Kombinationsgabe unter Beweis stellen. Je mehr richtige Steine Sie auswählen, desto mehr Punkte werden Ihnen gutgeschrieben.

DOMINGO besticht durch sein einfaches, aber fesselndes Spielprinzip. Es gilt, aus drei vorgegebenen Steinkombinationen diejenige auszuwählen, die im Mosaik mit den anderen die meisten Punkte ergeben. Um Punkte zu ergattern, sind die Steine der Farbe nach zu ordnen, vor dem endgültigen Setzen der Steine bietet sich die Möglichkeit, die Kombination umzudrehen. Aus der Kombination rot-grün-blau wird somit blau-grün-rot. Punkte werden waagerecht, senkrecht und diagonal gezählt, man muß also ganz schön aufpassen. Der aktuelle Punktestand wird jederzeit angezeigt, selbst ein relativ großer Rückstand kann durch geschicktes Setzen der Steine noch egalisiert werden.
Domingo wird mit den Cursortasten gesteuert und garantiert viel Spielfreude.
(sr)

für 464-664-6128

```
1000 REM ################################## [2151]
################
1010 REM ##########                        [1006]
      ##########
1020 REM #####             D O M I N G O   [1002]
           #####
1030 REM #####                  1986       [1092]
           #####
1040 REM #####             Stefan Harden   [1465]
           #####
1050 REM ##########                        [1006]
      ##########
1060 REM ################################## [2151]
################
1070 REM                                    [272]
1080 REM #####  Titelbild  #####            [1642]
1090 REM                                    [272]
1100 MODE 0                                 [507]
1110 INK 0,0 : INK 1,24 : INK 2,2 : INK 3, [1034]
25 : INK 4,6
1120 BORDER 12 : PAPER 0                    [1110]
1130 RESTORE 1300 : PRINT CHR$(22);CHR$(1) [1788]
1140 FOR x=4 TO 17 STEP 2                   [631]
1150  FOR i=1 TO 2                          [436]
1160   READ s1,s2,s3,s4,s5,s6,s7,s8 : SYMB [3018]
OL 255,s1,s2,s3,s4,s5,s6,s7,s8
1170   LOCATE x,10 : PEN i : PRINT CHR$(25 [3337]
5); : PRINT " ";
1180  NEXT i                                [375]
1190  SOUND 4,400,3 : SOUND 4,0,3           [1590]
1200 NEXT x                                 [356]
1210 PRINT CHR$(22);CHR$(0)                 [1444]
1220 FOR p=500 TO 100 STEP -10 : SOUND 4,p [2360]
```

Listing "Domingo"

```
,7 : NEXT p
1230 FOR p=0 TO 500 : NEXT p                [1385]
1240 LOCATE 7,13 : PRINT CHR$(164);" 1986" [1921]
1250 FOR x=17 TO 4 STEP -1                  [1319]
1260  LOCATE x,20 : PEN 3 : PRINT MID$("st [2634]
efan harden",x-3,1);
1270 SOUND 4,150,10                         [963]
1280  FOR warte=0 TO 200 : NEXT warte      [2539]
1290 NEXT x                                 [356]
1300 DATA &f0,&48,&44,&44,&44,&48,&f0,&00, [3229]
&08,&B4,&22,&22,&22,&24,&08,&F0
1310 DATA &30,&48,&84,&84,&84,&48,&30,&00, [3121]
&08,&34,&4A,&42,&42,&24,&08,&30
1320 DATA &84,&CC,&EC,&FC,&94,&84,&84,&00, [3202]
&42,&22,&12,&02,&6A,&52,&42,&C6
1330 DATA &7C,&10,&10,&10,&10,&10,&7C,&00, [4099]
&02,&6E,&08,&08,&08,&08,&02,&7E
1340 DATA &84,&C4,&E4,&94,&8C,&84,&84,&00, [2378]
&42,&22,&12,&6A,&42,&42,&42,&C6
1350 DATA &38,&44,&80,&80,&8C,&84,&7C,&00, [3229]
&04,&3A,&46,&40,&42,&42,&02,&7E
1360 DATA &30,&48,&84,&84,&84,&48,&30,&00, [3121]
&08,&34,&4A,&42,&42,&24,&08,&30
1370 REM                                    [272]
1380 REM #####  Vorbereitung  #####         [1987]
1390 REM                                    [272]
1400 DEFINT a-z                             [553]
1410 DIM stein(40,3),feld(20,20),spst(3),c [2165]
omst(3)
1420 RESTORE 1480                           [735]
1430 FOR n=0 TO 39                          [569]
1440  FOR s=1 TO 3                         [1024]
1450   READ stein(n,s)                      [796]
1460  NEXT s                                [365]
1470 NEXT n                                 [366]
1480 DATA 1,1,1,1,1,2,1,1,3,1,1,4,1,2,1,1, [2136]
3,1,1,4,1,1,2,2,1,3,3,1,4,4,1,2,3
1490 DATA 1,2,4,1,3,2,1,3,4,1,4,2,1,4,3     [2251]
1500 DATA 2,2,2,2,2,1,2,2,3,2,2,4,2,1,2,2, [2268]
```

Listing "Domingo"

```
3,2,2,4,2,2,3,3,2,4,4,2,1,3,2,1,4
1510 DATA 2,3,4,2,4,3                                    [591]
1520 DATA 3,3,3,3,3,4,3,1,3,3,2,3,3,4,3,3, [2268]
4,4,3,1,4,3,2,4,3,4,2
1530 DATA 4,4,4,4,1,4,4,2,4,4,3,4                        [1432]
1540 LOCATE 1,24 : PEN 4 : PRINT "  los ge [3161]
ht's durch"
1550 PRINT "    Tastendruck"                             [1802]
1560 CALL &BB18                                          [389]
1570 REM                                                 [272]
1580 REM #####  Spielregeln  #####                       [823]
1590 REM                                                 [272]
1600 MODE 1                                              [506]
1610 INK 0,1 : INK 1,25                                  [1031]
1620 BORDER 0                                            [1008]
1630 PEN 1                                               [549]
1640 PRINT "          D O M I N G O"                     [1227]
1650 PRINT                                               [361]
1660 PRINT "          Spielregeln"                       [2326]
1670 PRINT TAB(11);STRING$(11,208)                       [1553]
1680 PRINT "Sie spielen gegen den Computer [2527]
."
1690 PRINT "Alle beide bekommen nur 3 Stei [3931]
ne."
1700 PRINT "Man muss die Steine so hinlege [4210]
n, dass"
1710 PRINT "man selbst am meisten Punkte b [3810]
ekommt."
1720 PRINT                                               [361]
1730 PRINT "Fuer mindestens 3 gleiche Farb [3510]
en gibt"
1740 PRINT "minimal 3 Punkte. Fuer mindest [3819]
ens 3"
1750 PRINT "gleiche Farben diagonal gibt e [4494]
s"
1760 PRINT "minimal 6 Punkte."                           [1916]
1770 PRINT                                               [361]
1780 PRINT "Los geht's durch Tastendruck ! [4242]
"
1790 CALL &BB18                                          [389]
1800 REM                                                 [272]
1810 REM #####  Modus / Farben / Windows   [2009]
#####
1820 REM                                                 [272]
1830 MODE 0                                              [507]
1840 INK 0,0 : INK 1,24 : INK 2,6 : INK 3, [2145]
2 : INK 4,9 : INK 5,26 : INK 6,13
1850 BORDER 4 : PAPER 6 : CLS                            [1963]
1860 WINDOW #1,1,20,3,4 : PAPER #1,1 : CLS [1772]
 #1
1870 WINDOW #2,1,12,20,25 : PAPER #2,2 : C [2710]
LS #2
1880 WINDOW #3,1,20,1,2 : PAPER #3,5 : CLS [2901]
 #3
1890 WINDOW #4,13,20,20,25 : PAPER #4,2 :  [2971]
CLS #4
1900 REM                                                 [272]
1910 REM #####  Anfang  #####                           [821]
1920 REM                                                 [272]
1930 PEN #3,4 : PRINT #3,"    D O M I N G  [1718]
O"
1940 PEN #3,2 : PRINT #3,STRING$(20,189);                [2446]
1950 GOSUB 2960                                          [907]
1960 FOR z=1 TO 3                                        [1028]
1970  GOSUB 2850 : comst(z)=q : GOSUB 2850 [2727]
 : spst(z)=q
1980 NEXT z                                              [354]
1990 GOSUB 3040                                          [953]
2000 xl=1 : yl=5                                         [716]
2010 GOSUB 2850 : x=xl : y=yl : GOSUB 2730 [3643]
 : y=yl-4 : GOSUB 3140
2020 REM                                                 [272]
2030 REM #####  Spielerzug  #####                       [1647]
2040 REM                                                 [272]
2050 CLS #4 : PEN #4,3 : PRINT #4,"Sie sin [2015]
d";
2060 PRINT #4,"am Zug!:";                                [1169]
2070 PRINT #4,"Welcher ";                                [1718]
2080 PRINT #4,"Stein ? ";                                [1685]
2090 sd$=INKEY$ : IF sd$="" THEN 2090 ELSE [2226]
 sd=VAL(sd$)
2100 IF sd<>1 AND sd<>2 AND sd<>3 THEN 209 [1936]
0
2110 IF xl>18 THEN 2460                                  [1164]
2120 yl=yl+1 : IF yl>19 THEN yl=5 : xl=xl+ [2804]
3
2130 q=spst(sd) : x=xl : y=yl : GOSUB 2730 [2378]
2140 CLS #4 : PRINT #4,"sorum ? " : PRINT  [2224]
```

Listing "Domingo"

```
#4,"j oder n";
2150 sd$=INKEY$ : IF sd$="" THEN 2150                    [1305]
2160 IF sd$="j" OR sd$="J" THEN 2180                     [2380]
2170 GOSUB 2900 : GOSUB 2730                             [1576]
2180 x=xl : y=yl-4 : GOSUB 3140 : GOSUB 28 [3687]
50 : spst(sd)=q : GOSUB 3040
2190 x=xl : y=yl-4 : GOSUB 3190 : sppkt=sp [4064]
pkt+pkt : GOSUB 2960
2200 IF xl>=18 THEN 2450                                 [1031]
2210 REM                                                 [272]
2220 REM #####  Computerzug  #####                       [1773]
2230 REM                                                 [272]
2240 CLS #4 : PEN #4,4                                   [1148]
2250 PRINT #4,"  jetzt "                                 [1203]
2260 PRINT #4,"  ich   "                                 [782]
2270 cpk=0 : ca=0 : cb=0                                 [993]
2280 yl=yl+1 : IF yl>19 THEN yl=5 : xl=xl+ [2804]
3
2290 IF xl>18 THEN 2460                                  [1164]
2300 FOR cz=1 TO 3                                       [449]
2310  FOR cx=0 TO 1                                      [402]
2320   q=comst(cz) : x=xl : y=yl-4 : GOSUB [3881]
 3140 : GOSUB 3190
2330   IF pkt>cpk THEN cpk=pkt : ca=cz : c [3276]
b=cx
2340   GOSUB 2920                                        [851]
2350  NEXT cx                                            [418]
2360 NEXT cz                                             [424]
2370 IF cpk=0 THEN q=comst(INT(RND*3)+1) E [3513]
LSE q=comst(ca)
2380 IF cb=1 THEN GOSUB 2920                             [987]
2390 x=xl : y=yl : GOSUB 2730 : y=yl-4 : G [2984]
OSUB 3140
2400 compkt=compkt+cpk : GOSUB 2960                      [1282]
2410 GOSUB 2850 : comst(ca)=q                            [1506]
2420 GOTO 2040                                           [355]
2430 REM **********************                         [1220]
2440 REM *****  E N D E  *****                           [457]
2450 REM **********************                         [1220]
2460 REM                                                 [272]
2470 IF compkt>sppkt THEN 2580                           [1033]
2480 REM #####  Gewonnen  #####                          [1675]
2490 REM                                                 [272]
2500 MODE 0 : INK 3,6,26 : BORDER 24,1                   [2139]
2510 LOCATE 9,5 : PEN 1 : PRINT "SIE"                    [1876]
2520 LOCATE 8,9 : PEN 2 : PRINT "haben"                  [1651]
2530 LOCATE 3,13 : PEN 3 : PRINT "G E W O  [2100]
N N E N"
2540 FOR p=1000 TO 0 STEP -10 : SOUND 4,p, [1427]
3 : NEXT p
2550 BORDER 1 : GOTO 2640                                [1317]
2560 REM                                                 [272]
2570 REM #####  Verloren  #####                          [2014]
2580 REM                                                 [272]
2590 MODE 0 : INK 1,26 : INK 3,0,26 : BORD [1751]
ER 0
2600 LOCATE 9,5 : PEN 1 : PRINT "SIE"                    [1876]
2610 LOCATE 8,9 : PRINT "haben"                          [1218]
2620 LOCATE 3,13 : PEN 3 : PRINT "V E R L  [2109]
O R E N"
2630 FOR p=0 TO 1000 STEP 10 : SOUND 4,p,3 [3152]
 : NEXT p
2640 INK 4,18 : LOCATE 1,23 : PEN 4 : PRIN [3534]
T " neues Spiel ? j/n"
2650 a$=INKEY$ : IF a$="" THEN 2650                      [1558]
2660 IF a$="J" OR  a$="j" THEN RUN                       [1511]
2670 MODE 1 : INK 1,24 : INK 0,0 : BORDER  [2420]
1 : PEN 1 : PAPER 0 : END
2680 REM ****************************                  [1504]
2690 REM *****  Unterprogramme  *****                    [819]
2700 REM ****************************                  [1504]
2710 REM                                                 [272]
2720 REM #####  Stein setzen  #####                       [1569]
2730 REM                                                 [272]
2740 FOR p=300 TO 0 STEP -50 : SOUND 4,p,5 [3569]
 : NEXT p
2750 LOCATE x,y                                          [672]
2760 FOR s=1 TO 3                                        [1024]
2770  PEN stein(q,s) : PAPER 0 : PRINT CHR [2830]
$(143);
2780 NEXT s                                              [365]
2790 PLOT (x-1)*32,399-((y-1)*16),0                      [1424]
2800 DRAWR 96,0 : DRAWR 0,-15 : DRAWR -96, [2766]
0 : DRAWR 0,15
2810 MOVER 32,0 : DRAWR 0,-15 : MOVER 32,0 [3207]
 : DRAWR 0,15
2820 RETURN                                              [555]
2830 REM                                                 [272]
```

Lisiting "Domingo"


```
2840 REM #####  zufaelliger Stein #####    [2599]
2850 REM                                   [272]
2860 q=INT(RND*39)                         [475]
2870 RETURN                                [555]
2880 REM                                   [272]
2890 REM #####  Stein drehen  #####        [854]
2900 REM                                   [272]
2910 SOUND 4,300,50                        [1007]
2920 stein(q,0)=stein(q,1) : stein(q,1)=st [4138]
ein(q,3) : stein(q,3)=stein(q,0)
2930 stein(q,0)=0 : RETURN                 [1797]
2940 REM                                   [272]
2950 REM #####  Anzeige von Punkten  ##### [2441]
2960 REM                                   [272]
2970 CLS #1                                [373]
2980 PEN #1,0 : PRINT #1,"Computer     Spi [2288]
eler";
2990 LOCATE #1,4,2 : PRINT #1,USING "###"; [2305]
compkt;
3000 LOCATE #1,17,2 : PRINT #1,USING "###" [2487]
;sppkt;
3010 RETURN                                [555]
3020 REM                                   [272]
3030 REM #####  Steine anzeigen  #####     [1633]
3040 REM                                   [272]
3050 CLS #2                                [372]
3060 PEN #2,5 : PRINT #2,"1   2   3"       [1534]
3070 y=21 : x=1                            [487]
3080 FOR i=1 TO 3                          [435]
3090  q=spst(i) : GOSUB 2730 : x=x+4       [2263]
3100 NEXT i                                [375]
3110 RETURN                                [555]
3120 REM                                   [272]
3130 REM #####  Zug festhalten  #####       [2832]
3140 REM                                   [272]
3150 feld(x,y)=stein(q,1) : feld(x+1,y)=st [4480]
ein(q,2) : feld(x+2,y)=stein(q,3)
3160 RETURN                                [555]
```

Listing "Domingo"

```
3170 REM                                   [272]
3180 REM #####  Punkte berechnen  #####    [1068]
3190 REM                                   [272]
3200 pkt=0 : wf=0                          [269]
3210 FOR n=0 TO 2                          [566]
3220 REM waagerecht                        [373]
3230  s1=1 : s2=1                          [799]
3240  IF wf<>0 THEN 3290                  [1070]
3250  IF feld(x+n+s1,y)=feld(x+n,y) THEN s [2887]
1=s1+1 : GOTO 3250
3260  IF feld(x+n-s2,y)=feld(x+n,y) THEN s [2360]
2=s2+1 : GOTO 3260
3270  IF s1+s2-1>2 THEN pkt=pkt+s1+s2-1    [2946]
3280  IF s1-1>1 THEN wf=1                  [897]
3290 REM senkrecht                         [678]
3300  s1=1                                 [331]
3310  IF feld(x+n,y-s1)=feld(x+n,y) THEN s [3167]
1=s1+1 : GOTO 3310
3320  IF s1>2 THEN pkt=pkt+s1              [961]
3330 REM diagonal l.u. - r.o.              [627]
3340  s1=1 : s2=1                          [799]
3350  IF feld(x+n-s1,y+s1)=feld(x+n,y) THE [2672]
N s1=s1+1 : GOTO 3350
3360  IF feld(x+n+s2,y-s2)=feld(x+n,y) THE [3739]
N s2=s2+1 : GOTO 3360
3370  IF s1+s2-1>2 THEN pkt=pkt+(s1+s2-1)* [3057]
2
3380 REM diagonal r.o. - l.u               [1096]
3390  s1=1                                 [331]
3400  IF feld(x+n-s1,y-s1)=feld(x+n,y) THE [3173]
N s1=s1+1 : GOTO 3400
3410  IF s1>2 THEN pkt=pkt+s1*2            [1176]
3420 NEXT n                                [366]
3430 RETURN                                [555]
```

Lisiting "Domingo"

Wer sucht, der findet!

LABYRINTH – Schatzsuche zu Hause

Stellen Sie sich vor, Sie wären Archäologe und hätten gerade die Pyramide des großen Ägypterkönigs Habniewas freigelegt. Da kommt jemand zu Ihnen, und verrät, daß in dieser Pyramide fünf Schätze zu finden sind. Was würden Sie also tun? Richtig, hinein und suchen. Nur leider haben diese Pyramiden im Innern ein weit verzweigtes Netz von Gängen, in denen man sich leicht verlaufen kann. Was ist Ihnen also lieber, nach Ägypten reisen und verhungern, oder LABYRINTH in Ihren Rechner zu Hause laden und auf jeden Fall unbeschadet bleiben?

Im LABYRINTH müssen Sie, wie schon gesagt, fünf Schätze suchen, die Ihnen am rechten Bildschirmrand angezeigt werden. Der Weg besteht aus Puzzleteilen, die Sie richtig aneinanderreihen müssen. Die Schätze können Sie erreichen, indem Sie mit dem Joystick auf das Feld gehen, in das Sie unseren (Sprite)-Professor schicken wollen. Sie können aber auch eine ganze Feldreihe horizontal oder vertikal verschieben, wenn Sie mit dem Joystick an den Rand der Feldreihen in der gewünschten Richtung gehen und die Feuertaste drücken. Haben Sie alle Schätze, müssen Sie wieder zum Ausgangsfeld zurückkehren.

Das Spiel ist für maximal vier Spieler gedacht, die alle nacheinander den Joystick bedienen müssen.
Die Listings geben Sie folgendermaßen ein: Zuerst Listing 1 abtippen und als LISTING1.BAS speichern, dann Listing 2 abtippen und als LISTING2.BAS speichern (wichtig!!). Als nächstes Listing 1 starten mit RUN"LISTING1 , das Programm erzeugt den ersten Teil des Binärcodes. Genauso gehen Sie mit Listing 2 um, dort wird der zweite Teil des Binärcodes erzeugt. Schließlich tippen Sie noch Listing 3 ab und starten es, das Programm erzeugt aus den beiden auf dem Datenträger befindlichen Binärdateien eine einzige.
Kassettenbesitzer sollten die beiden Binärdateien hintereinander auf einer gesonderten Kassette von den Ladern abspeichern lassen.

(Carsten Schaar/jb)

Für 464-664-6128



```
96 '*************************          [894]
97 '  Listing 1 :  LAB1.LAD              [1483]
98 '*************************          [894]
99 '                                     [117]
100 DATA C5,E1,0E,07,CD,CE,BC,AF,4C1     [1265]
101 DATA 01,00,00,CD,32,BC,3E,01,1FB     [1655]
```

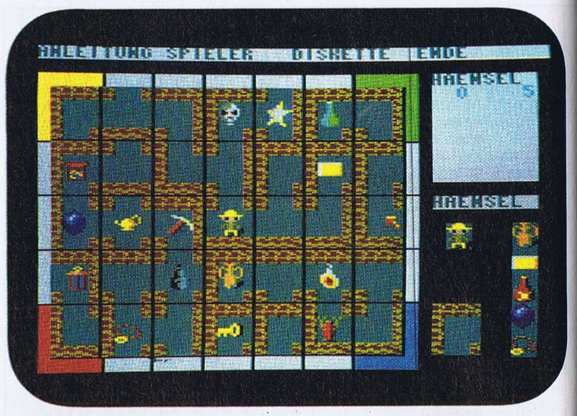
Listing "LABYRINTH"

```
102 DATA 01,00,00,CD,32,BC,3E,00,1FA     [1441]
103 DATA CD,0E,BC,21,00,C0,11,01,28A     [1779]
104 DATA C0,01,FF,3F,36,C0,ED,B0,492     [1276]
105 DATA DD,21,6F,4E,3E,00,F5,DD,3CB     [1809]
106 DATA 4E,00,DD,46,00,CD,32,BC,32C     [1664]
107 DATA F1,DD,23,3C,FE,10,20,EE,449     [1256]
108 DATA DD,46,00,DD,4E,00,CD,38,353     [1488]
109 DATA BC,21,0C,4F,CD,78,45,3E,300     [1291]
110 DATA 03,21,77,4F,F5,CD,82,45,373     [1627]
111 DATA 3E,80,CD,82,45,7E,CD,82,41F     [2017]
```

Listing "LABYRINTH"

```
112 DATA 45,23,3E,9D,CD,82,45,7E,355      [2029]
113 DATA CD,82,45,23,F1,3C,FE,17,3F9      [1178]
114 DATA 20,E2,21,9F,4F,06,03,C5,2DF      [1923]
115 DATA 56,23,46,23,7E,23,5E,23,204      [1469]
116 DATA F5,C5,CD,82,45,3E,80,82,48E      [1943]
117 DATA CD,82,45,3E,20,CD,82,45,386      [1540]
118 DATA 10,F9,C1,F1,3C,BB,20,E8,4BA      [1888]
119 DATA C1,10,DC,21,6D,52,11,1E,2BC      [1759]
120 DATA 53,01,30,00,ED,B0,AF,32,302      [1191]
121 DATA 9D,52,CD,26,46,CD,E7,47,423      [1412]
122 DATA C3,4E,4A,3A,9D,52,6F,26,319      [1768]
123 DATA 0C,CD,7E,48,11,1E,53,19,23A      [1600]
124 DATA 7E,B7,20,0B,3A,9D,52,3C,2C5      [1615]
125 DATA E6,03,32,9D,52,18,E4,3A,340      [2012]
126 DATA 9D,52,3C,21,0D,20,CD,EA,330      [1551]
127 DATA 45,3D,6F,26,0C,CD,7E,48,2B6      [1788]
128 DATA 11,1F,53,19,3E,9F,CD,82,2C8      [1553]
129 DATA 45,3E,0B,CD,82,45,CD,78,367      [1365]
130 DATA 45,23,06,05,3A,9D,52,6F,20B      [2029]
131 DATA 60,CD,7E,48,11,5F,53,19,2CF      [1147]
132 DATA EB,21,0D,25,1A,CD,EA,45,354      [1211]
133 DATA 2C,2C,13,10,F7,CD,C5,48,34C      [1519]
134 DATA DD,21,F3,4D,CD,E3,49,C2,4F9      [658]
135 DATA 2C,4A,DD,21,F7,4D,CD,E3,468      [1871]
136 DATA 49,C2,4E,4A,DD,21,FB,4D,3E9      [1615]
137 DATA CD,E3,49,C2,FA,4B,DD,21,4FE      [830]
138 DATA FF,4D,CD,E3,49,C2,F1,4C,544      [611]
139 DATA DD,21,03,4E,CD,E3,49,C2,40A      [1279]
140 DATA 8D,43,DD,21,07,4E,CD,E3,3D3      [1374]
141 DATA 49,28,67,3A,F2,4D,D6,06,32D      [867]
142 DATA CB,3F,CB,3F,CB,3F,DD,21,41C      [1368]
143 DATA E9,4D,DD,36,00,00,DD,77,39D      [1310]
144 DATA 01,CD,BC,4B,C2,0D,41,F5,3DA      [1135]
145 DATA 6F,26,07,CD,7E,48,29,11,269      [1927]
146 DATA 9E,52,19,7E,32,EB,4D,23,314      [1589]
147 DATA 7E,32,EC,4D,2B,E5,D1,23,3ED      [1749]
148 DATA 23,01,0C,00,ED,B0,CD,1A,2B4      [1382]
149 DATA 43,F1,21,1E,53,11,0C,00,1E3      [1430]
150 DATA DD,21,4E,53,06,04,4F,7E,276      [1960]
151 DATA B7,28,09,DD,7E,01,B9,20,31D      [1992]
152 DATA 03,DD,35,00,DD,23,DD,23,315      [1231]
153 DATA 19,10,EC,79,CD,44,43,C3,3A5      [1046]
154 DATA CA,44,DD,21,0B,4E,CD,E3,415      [1599]
155 DATA 49,28,68,3A,F2,4D,D6,06,32E      [1198]
156 DATA CB,3F,CB,3F,CB,3F,DD,21,41C      [1368]
157 DATA E9,4D,DD,36,00,06,DD,77,3A3      [1248]
158 DATA 01,CD,BC,4B,C2,0D,41,F5,3DA      [1135]
159 DATA 6F,26,07,CD,7E,48,29,11,269      [1927]
160 DATA AB,52,19,7E,32,EC,4D,2B,32A      [1298]
161 DATA 7E,32,EB,4D,23,E5,D1,2B,3EC      [1419]
162 DATA 2B,01,0C,00,ED,B8,1B,CD,2C5      [2337]
163 DATA 1A,43,F1,21,1E,53,11,0C,1FD      [1505]
164 DATA 00,DD,21,4E,53,06,04,4F,1F8      [1079]
165 DATA 7E,B7,28,09,DD,7E,01,B9,37B      [1348]
166 DATA 20,03,DD,34,00,DD,23,DD,311      [1352]
167 DATA 23,19,10,EC,79,CD,44,43,305      [1414]
168 DATA C3,CA,44,DD,21,0F,4E,CD,3F9      [1416]
169 DATA E3,49,28,72,3A,F1,4D,D6,414      [1550]
170 DATA 02,CB,3F,CB,3F,CB,3F,DD,3FD      [1420]
171 DATA 21,E9,4D,DD,77,00,DD,36,3BE      [1589]
172 DATA 01,04,CD,BC,4B,C2,0D,41,2E9      [1280]
173 DATA F5,87,6F,26,00,11,D6,52,34A      [1274]
174 DATA 19,7E,32,EB,4D,23,7E,32,2D4      [1293]
175 DATA EC,4D,2B,E5,11,F2,FF,19,464      [1580]
176 DATA D1,06,04,7E,12,23,13,7E,21F      [1398]
177 DATA 12,2B,E5,11,F2,FF,19,D1,40E      [1159]
178 DATA 10,F1,CD,1A,43,F1,21,1E,35B      [1444]
179 DATA 53,11,0C,00,DD,21,4E,53,20F      [941]
180 DATA 06,04,4F,7E,B7,28,09,DD,29C      [1315]
181 DATA 7E,00,B9,20,03,DD,34,01,26C      [1718]
182 DATA DD,23,DD,23,19,10,EC,79,38E      [1309]
183 DATA CD,69,43,C3,CA,44,DD,21,448      [1295]
184 DATA 13,4E,CD,E3,49,CA,0D,41,372      [1843]
185 DATA 3A,F1,4D,D6,02,CB,3F,CB,425      [1635]
186 DATA 3F,CB,3F,DD,21,E9,4D,DD,45A      [1152]
187 DATA 77,00,DD,36,01,00,CD,BC,314      [1125]
188 DATA 4B,C2,0D,41,F5,87,6F,26,36C      [2475]
189 DATA 00,11,9E,52,19,7E,32,EB,2B5      [1900]
190 DATA 4D,23,7E,32,EC,4D,2B,E5,369      [2052]
191 DATA 11,0E,00,19,D1,06,04,7E,191      [2077]
192 DATA 12,23,13,7E,12,2B,E5,11,1F9      [1605]
193 DATA 0E,00,19,D1,10,F1,CD,1A,2E0      [1735]
194 DATA 43,F1,21,1E,53,11,0C,00,1E3      [1430]
195 DATA DD,21,4E,53,06,04,4F,7E,276      [1960]
196 DATA B7,28,09,DD,7E,00,B9,20,31C      [2294]
197 DATA 03,DD,35,01,DD,23,DD,23,316      [1720]
198 DATA 19,10,EC,79,CD,69,43,C3,3CA      [758]
```

Listing "LABYRINTH"




```
199 DATA CA,44,21,E4,52,7E,12,23,318      [1103]
200 DATA 13,7E,12,11,E4,52,21,E6,2F1      [1714]
201 DATA 52,01,38,00,ED,B0,11,EB,324      [1478]
202 DATA 4D,21,1C,53,1A,77,13,23,1A4      [1312]
203 DATA 1A,77,11,E4,52,21,13,1F,22B      [1460]
204 DATA CD,3E,47,C9,F5,26,0E,6F,3B3      [1656]
205 DATA CD,7E,48,11,9E,52,19,EB,398      [1597]
206 DATA F1,87,87,C6,03,26,01,6F,35E      [1319]
207 DATA 06,07,CD,3E,47,13,13,24,1A9      [1370]
208 DATA 24,24,24,10,F5,CD,C3,46,347      [2041]
209 DATA C9,87,6F,26,00,11,9E,52,2E6      [1420]
210 DATA 19,EB,87,3C,67,2E,03,06,265      [1133]
211 DATA 05,CD,3E,47,E5,21,0E,00,26B      [1326]
212 DATA 19,EB,E1,2C,2C,2C,2C,10,2A5      [915]
213 DATA F0,CD,C3,46,C9,DD,21,E9,576      [1996]
214 DATA 4D,3A,F1,4D,D6,02,CB,3F,3A7      [2153]
215 DATA CB,3F,CB,3F,DD,77,00,3A,3A2      [1735]
216 DATA F2,4D,D6,06,CB,3F,CB,3F,42F      [1497]
217 DATA CB,3F,DD,77,01,CD,BC,4B,433      [1525]
218 DATA C2,0D,41,3A,9D,52,87,6F,32F      [1233]
219 DATA 26,00,11,4E,53,19,E5,FD,2D3      [1375]
220 DATA E1,DD,7E,00,FD,BE,00,28,41F      [2010]
221 DATA 1D,DD,7E,01,FD,BE,01,C2,3F7      [1347]
222 DATA 0D,41,DD,7E,00,3D,FD,BE,3A1      [1289]
223 DATA 00,28,1E,3C,3C,FD,BE,00,279      [1840]
224 DATA CA,21,44,C3,0D,41,DD,7E,39B      [1273]
225 DATA 01,3D,FD,BE,01,28,5A,3C,2B8      [1950]
226 DATA 3C,FD,BE,01,28,70,C3,0D,360      [1582]
227 DATA 41,FD,66,00,FD,6E,01,CD,3DD      [1699]
228 DATA 1E,45,7E,E5,67,2E,03,CD,32B      [1893]
229 DATA 8F,48,FE,02,E1,CA,0D,41,3D0      [1056]
230 DATA E5,23,23,7E,67,2E,03,CD,30E      [1628]
231 DATA 8F,48,B7,E1,CA,0D,41,18,39F      [1267]
232 DATA 5E,DD,66,00,DD,6E,01,CD,3BA      [1988]
233 DATA 1E,45,7E,E5,67,2E,03,CD,32B      [1893]
234 DATA 8F,48,FE,02,E1,CA,0D,41,3D0      [1056]
235 DATA 23,23,E5,7E,67,2E,03,CD,30E      [1199]
236 DATA 8F,48,B7,E1,CA,0D,41,18,39F      [1267]
237 DATA 36,FD,66,00,FD,6E,01,CD,3D2      [1723]
238 DATA 1E,45,7E,FE,06,D2,0D,41,305      [1818]
239 DATA E5,11,0E,00,19,7E,FE,03,29C      [1387]
240 DATA E1,DA,0D,41,18,19,DD,66,37D      [930]
241 DATA 00,DD,6E,01,CD,1E,45,7E,2FA      [666]
242 DATA FE,06,D2,0D,41,11,0E,00,243      [1291]
243 DATA 19,7E,FE,03,DA,0D,41,E5,3A5      [1134]
244 DATA FD,7E,00,87,87,3C,67,FD,429      [1702]
245 DATA 7E,01,87,87,C6,03,6F,D1,396      [1932]
246 DATA CD,3E,47,DD,6E,00,FD,75,40F      [1799]
247 DATA 00,DD,66,01,FD,74,01,CD,383      [1745]
248 DATA C3,46,FD,66,00,FD,6E,01,3D8      [1085]
249 DATA CD,1E,45,23,7E,EB,B7,28,39B      [1752]
250 DATA 19,4F,3A,9D,52,6F,26,05,22B      [844]
251 DATA CD,7E,48,D5,11,5F,53,19,344      [1153]
252 DATA D1,79,06,05,BE,28,3E,23,29C      [1094]
253 DATA 10,FA,3A,9D,52,6F,26,0C,2D4      [1330]
254 DATA CD,7E,48,11,29,53,19,7E,2B7      [1486]
255 DATA B7,20,1E,3A,9D,52,87,6F,314      [781]
256 DATA 26,00,11,67,4E,19,46,23,16E      [1446]
257 DATA 4E,6F,26,00,11,4E,53,19,1AE      [1129]
258 DATA 78,BE,20,05,23,79,BE,28,2DD      [1821]
259 DATA 38,3A,9D,52,3C,E6,03,32,2B8      [1386]
```

Listing "LABYRINTH"


```
260 DATA 9D,52,C3,B3,40,36,19,EB,3DF      [685]
261 DATA 36,00,3A,9D,52,6F,26,0C,200      [1013]
262 DATA CD,7E,48,11,29,53,19,35,26E      [1355]
263 DATA CD,E7,47,C3,B3,40,D5,E5,56B      [686]
264 DATA 26,07,CD,7E,48,D1,5A,16,301      [1822]
265 DATA 00,19,29,11,9E,52,19,D1,22D      [2289]
266 DATA C9,CD,2B,4D,DD,21,43,4E,39D      [1740]
267 DATA CD,68,49,21,53,52,CD,78,389      [1702]
268 DATA 45,3A,9D,52,6F,26,0C,CD,2DC      [2615]
269 DATA 7E,48,11,1F,53,19,CD,78,2A7      [1142]
270 DATA 45,34,CD,E7,47,3A,9D,52,39D      [1866]
271 DATA 3C,21,15,08,CD,EA,45,21,297      [1798]
272 DATA 0E,1E,CD,EA,45,CD,06,BB,3B6      [2067]
273 DATA FE,58,20,F9,CD,37,4D,CD,48D      [1284]
274 DATA E7,47,CD,26,46,C3,B3,40,41D      [1596]
275 DATA 7E,23,FE,FF,C8,CD,82,45,4FA      [1426]
276 DATA 18,F6,E5,F5,2A,EF,4D,FE,54C      [1479]
277 DATA FE,20,03,25,18,1E,FE,20,29A      [748]
278 DATA 30,03,6F,18,17,FE,80,38,287      [1487]
279 DATA 05,D6,80,67,18,0E,D6,20,2DE      [939]
280 DATA CD,B2,45,24,7C,FE,28,38,3C2      [1503]
281 DATA 03,26,00,2C,22,EF,4D,F1,2A4      [1361]
282 DATA E1,C9,E5,D5,C5,F5,E5,6F,672      [2132]
283 DATA 26,00,29,29,29,29,11,A3,17E      [2033]
284 DATA 53,19,EB,E1,CB,24,E5,26,432      [1987]
285 DATA 50,CD,7E,48,C1,48,06,00,2F2      [1090]
286 DATA 09,01,00,C0,09,06,08,1A,0FB      [1455]
287 DATA 77,13,23,1A,77,13,2B,3E,1BA      [1355]
288 DATA 08,84,67,10,F2,F1,C1,D1,478      [1880]
289 DATA E1,C9,E5,D5,C5,F5,E5,6F,672      [2132]
290 DATA 26,00,29,29,29,29,29,11C      [1396]
291 DATA 11,A3,57,19,EB,E1,CB,24,3DF      [1201]
292 DATA E5,26,50,CD,7E,48,C1,48,3F7      [1136]
293 DATA 06,00,09,01,00,C0,09,06,0DF      [1130]
294 DATA 10,C5,E5,EB,01,04,00,ED,397      [1774]
295 DATA B0,EB,E1,CD,26,BC,C1,10,4FC      [1637]
296 DATA F0,F1,C1,D1,E1,C9,3E,3C,597      [1774]
297 DATA 21,80,4E,CD,BA,47,3E,3C,337      [2068]
298 DATA 21,BC,4E,CD,BA,47,3E,14,34B      [1395]
299 DATA 21,F8,4E,CD,BA,47,21,F8,44E      [1187]
300 DATA 4E,11,5F,53,01,14,00,ED,213      [1645]
301 DATA B0,AF,21,80,4E,11,BC,4E,369      [1066]
302 DATA 01,9E,52,F6,00,28,16,FE,323      [1971]
303 DATA 06,28,12,FE,1C,28,0E,FE,28E      [1726]
304 DATA 22,28,0A,F5,7E,23,02,03,1EF      [1214]
305 DATA 1A,13,02,0B,F1,03,03,23,154      [1880]
306 DATA 3C,FE,40,20,DE,CD,95,46,420      [2206]
307 DATA 06,04,21,29,53,11,0C,00,0C4      [1696]
308 DATA 3E,05,77,19,10,FC,11,4E,23E      [2017]
309 DATA 53,21,67,4E,01,08,00,ED,21F      [1152]
310 DATA B0,CD,C3,46,C9,E5,C5,D5,5CE      [1757]
311 DATA F5,06,05,21,03,01,11,9E,1D4      [1280]
312 DATA 52,C5,E5,06,07,CD,3E,47,35B      [1162]
313 DATA 24,24,24,24,13,13,10,F5,1BB      [1851]
314 DATA E1,2C,2C,2C,2C,C1,10,E9,34B      [1353]
315 DATA 21,13,1F,CD,3E,47,F1,D1,367      [1547]
316 DATA C1,E1,C9,E5,C5,D5,F5,AF,68E      [1579]
317 DATA DD,21,4E,53,21,1E,53,E5,316      [1324]
318 DATA 4F,7E,B7,20,25,DD,6E,01,315      [1300]
319 DATA 26,07,CD,7E,48,DD,5E,00,2FB      [2242]
320 DATA 16,00,19,29,11,9F,52,19,173      [1432]
321 DATA 7E,DD,66,00,DD,6E,01,29,336      [2247]
322 DATA 29,24,24,2C,2C,2C,2C,CD,1EE      [1343]
323 DATA EA,45,79,E1,11,0C,00,19,2BF      [1764]
324 DATA DD,23,DD,23,3C,FE,04,20,35E      [883]
325 DATA C6,AF,DD,21,4E,53,21,1E,353      [1447]
326 DATA 53,E5,4F,7E,B7,28,13,79,370      [1104]
327 DATA 3C,DD,66,00,DD,6E,01,29,2F4      [2024]
328 DATA 29,24,24,2C,2C,2C,2C,CD,1EE      [1343]
329 DATA EA,45,79,E1,11,0C,00,19,2BF      [1764]
330 DATA DD,23,DD,23,3C,FE,04,20,35E      [883]
331 DATA D8,F1,D1,C1,E1,C9,F5,C5,6BF      [696]
332 DATA D5,E5,E5,1A,47,67,2E,03,398      [1993]
333 DATA CD,8F,48,4F,E1,E5,3E,1E,415      [931]
334 DATA CD,B2,45,78,FE,03,3E,1E,399      [1390]
335 DATA 38,01,3C,24,CD,B2,45,24,281      [1342]
336 DATA CD,B2,45,24,3E,3B,CD,B2,3E0      [2083]
337 DATA 45,E1,2C,E5,79,B7,3E,1E,3C3      [1376]
338 DATA 28,01,3C,CD,B2,45,2C,CD,322      [1347]
339 DATA B2,45,E1,E5,24,24,24,79,3A2      [1547]
340 DATA FE,02,3E,3B,28,01,3C,CD,2AB      [723]
341 DATA B2,45,2C,CD,B2,45,E1,2C,3F4      [1574]
342 DATA 2C,3E,3D,CD,B2,45,24,78,307      [841]
343 DATA FE,06,3E,3D,30,01,3C,CD,2B9      [1730]
344 DATA B2,45,24,CD,B2,45,24,3E,341      [1055]
345 DATA 3F,CD,B2,45,E1,E5,2C,24,419      [2097]
```

Listing "LABYRINTH"

```
346 DATA 13,1A,CD,EA,45,E1,D1,C1,49C      [2290]
347 DATA F1,C9,47,32,E8,4D,C5,E5,512      [1219]
348 DATA 78,CD,A1,48,16,00,5F,19,2BC      [1177]
349 DATA 7E,F5,3A,E8,4D,93,4F,06,3CA      [1609]
350 DATA 00,E5,D1,23,ED,B0,3A,E8,498      [2257]
351 DATA 4D,3D,5F,16,00,F1,E1,E5,3B6      [1636]
352 DATA 19,77,E1,C1,10,D8,C9,21,404      [1430]
353 DATA 1E,53,06,04,0E,02,C5,7E,1CE      [1367]
354 DATA B7,28,2C,23,79,CD,82,45,33B      [1029]
355 DATA 0C,3E,9F,CD,82,45,CD,78,3C2      [1836]
356 DATA 45,79,CD,82,45,0C,3E,9F,33B      [2186]
357 DATA CD,82,45,7E,23,CD,42,48,38C      [875]
358 DATA 3E,20,CD,82,45,CD,82,45,386      [1437]
359 DATA 7E,23,CD,42,48,18,1D,06,233      [994]
360 DATA 02,79,CD,82,45,3E,9F,CD,3B9      [1327]
361 DATA 82,45,C5,3E,20,06,08,CD,2C5      [1057]
362 DATA 82,45,10,FB,C1,0C,10,E9,398      [1609]
363 DATA 11,0C,00,19,C1,0C,0C,10,11F      [2004]
364 DATA AD,C9,F5,D5,E5,C5,CB,80,635      [1897]
365 DATA 6F,26,00,11,64,00,CD,62,239      [1238]
366 DATA 48,1E,0A,CD,62,48,CB,C0,372      [886]
367 DATA 1E,01,CD,62,48,C1,E1,D1,409      [1668]
368 DATA F1,C9,AF,ED,52,38,03,3C,41F      [1805]
369 DATA 18,F9,19,B7,20,04,CB,40,310      [1087]
370 DATA 28,07,CB,C0,C6,30,C3,82,3F5      [1827]
371 DATA 45,3E,20,C3,82,45,C5,D5,3C7      [1645]
372 DATA 5D,16,00,6A,06,08,29,30,144      [1652]
373 DATA 01,19,10,FA,D1,C1,C9,C5,444      [1303]
374 DATA AF,06,08,CB,14,17,95,30,278      [1871]
375 DATA 01,85,3F,10,F6,CB,14,C1,36B      [1787]
376 DATA C9,D5,E5,F5,ED,5F,6F,67,59A      [961]
377 DATA CD,7E,48,ED,5B,ED,4D,19,42E      [577]
378 DATA 7C,B7,20,01,24,7D,B7,20,2CC      [1525]
379 DATA 01,2C,22,ED,4D,F1,6F,CD,3B6      [1161]
380 DATA 8F,48,E1,D1,C9,FD,21,F1,561      [1311]
381 DATA 4D,21,00,C0,FD,4E,01,CB,345      [2001]
382 DATA 39,30,03,21,00,E0,16,50,1D3      [1748]
383 DATA AF,06,09,1F,CB,19,30,01,1F2      [1502]
384 DATA 82,10,F8,47,09,16,00,FD,2ED      [1674]
385 DATA 5E,00,19,E5,DD,21,47,4E,2EF      [2010]
386 DATA 11,7F,53,06,08,C5,E5,06,2A1      [1366]
387 DATA 02,7E,12,DD,4E,00,A1,DD,33B      [616]
388 DATA 4E,01,B1,77,23,13,DD,23,2AD      [1898]
389 DATA DD,23,10,ED,E1,CD,26,BC,48D      [1754]
390 DATA C1,10,E2,CD,24,BB,FD,7E,4DA      [1976]
391 DATA 01,CB,44,28,04,B7,28,01,21C      [2305]
392 DATA 3D,CB,4C,28,05,FE,30,28,2D7      [1116]
393 DATA 01,3C,FD,77,01,FD,7E,00,32D      [1801]
394 DATA CB,54,28,04,B7,28,01,3D,268      [1536]
395 DATA CB,5C,28,05,FE,4E,28,01,2C9      [1353]
396 DATA 3C,FD,77,00,CD,19,BD,E1,434      [1372]
397 DATA 11,7F,53,06,08,C5,1A,77,247      [1088]
398 DATA 23,13,1A,77,13,2B,CD,26,1F8      [2061]
399 DATA BC,C1,10,F1,CD,09,BB,D2,4E1      [490]
400 DATA C9,48,FE,58,C8,C3,C9,48,503      [981]
401 DATA DD,6E,02,26,50,CD,7E,48,356      [1289]
402 DATA 11,00,C0,19,DD,5E,00,CB,2F0      [1512]
403 DATA 23,16,00,19,DD,7E,03,DD,28D      [1339]
404 DATA 46,02,90,3C,47,AF,C5,E5,3B4      [1832]
405 DATA 16,00,B7,28,0B,14,14,14,13C      [1304]
406 DATA 3E,01,B8,20,03,14,14,14,156      [1097]
407 DATA DD,7E,01,DD,46,00,90,3C,34B      [1054]
408 DATA 47,AF,D5,C5,E5,B7,28,07,45B      [1463]
409 DATA 3E,01,14,B8,20,01,14,CB,20B      [1676]
410 DATA 22,CB,22,CB,22,CB,22,5A,343      [659]
411 DATA 16,00,21,B3,53,19,EB,06,247      [1957]
412 DATA 08,E1,E5,1A,77,23,13,1A,2AF      [1477]
413 DATA 77,13,2B,3E,08,84,67,10,1F6      [1777]
414 DATA F2,E1,23,23,C1,D1,3E,01,3EA      [2021]
415 DATA 10,C8,E1,11,50,00,19,C1,2F4      [2024]
416 DATA 10,A4,C9,FD,21,F1,4D,C5,49E      [1944]
417 DATA 4F,FD,7E,00,DD,46,00,CB,3B8      [1455]
418 DATA 20,B8,38,23,DD,46,01,04,25B      [1786]
419 DATA CB,20,B8,30,1A,FD,7E,01,369      [1416]
420 DATA DD,46,02,CB,20,B8,38,0F,30F      [984]
421 DATA DD,46,03,04,CB,20,B8,30,2FD      [1232]
422 DATA 06,3E,FF,B7,79,C1,C9,AF,4AC      [1342]
423 DATA B7,79,C1,C9,C5,CD,68,49,4FD      [885]
424 DATA DD,23,DD,23,DD,23,DD,23,400      [1953]
425 DATA C1,10,F1,C9,CD,2B,4D,DD,4AD      [1428]
426 DATA 21,17,4E,CD,68,49,21,AB,2D0      [1850]
427 DATA 4F,CD,78,45,CD,C5,48,DD,490      [1579]
428 DATA 21,17,4E,CD,E3,49,20,F4,393      [2158]
429 DATA CD,37,4D,C3,0D,41,CD,2B,35A      [1286]
430 DATA 4D,DD,21,37,4E,CD,68,49,34E      [1353]
431 DATA AF,DD,21,3B,4E,21,1E,53,2C8      [1758]
```

Listing "LABYRINTH"

```
432 DATA 0E,03,F5,DD,E5,E5,DD,71,4FB      [2558]
433 DATA 02,DD,71,06,0C,0C,DD,71,2BC      [2354]
434 DATA 03,DD,71,07,CD,68,49,DD,3B3      [1704]
435 DATA 23,DD,23,DD,23,DD,23,CD,3F0      [997]
436 DATA 68,49,79,0C,0C,3D,CD,82,2CE      [911]
437 DATA 45,3E,87,CD,82,45,E1,7E,3FD      [1574]
438 DATA B7,3E,20,28,02,3E,40,CD,28A      [1615]
439 DATA 82,45,3E,8B,CD,82,45,23,347      [1200]
440 DATA CD,78,45,23,23,DD,E1,F1,47F      [2224]
441 DATA 3C,FE,04,20,B5,21,38,52,2BE      [1694]
442 DATA CD,78,45,CD,C5,48,DD,21,462      [2247]
443 DATA 37,4E,CD,E3,49,20,1D,21,2DC      [1625]
444 DATA 1E,53,11,0C,00,06,04,7E,116      [1637]
445 DATA B7,20,05,19,10,F9,18,E3,2F9      [1477]
446 DATA CD,37,4D,CD,E7,47,CD,C3,4DC      [1102]
447 DATA 46,C3,B3,40,AF,DD,21,3B,3E4      [1292]
448 DATA 4E,0E,03,DD,71,02,DD,71,2FD      [765]
449 DATA 06,0C,0C,DD,71,03,DD,71,2BD      [1285]
450 DATA 07,0C,0C,CD,E3,49,20,39,271      [1332]
451 DATA DD,23,DD,23,DD,23,DD,23,400      [1953]
452 DATA CD,E3,49,20,0F,DD,2B,DD,40D      [2574]
453 DATA 2B,DD,2B,DD,2B,3C,FE,04,379      [1666]
454 DATA 20,D1,18,9F,F5,87,87,C6,471      [1043]
455 DATA 04,CD,82,45,3E,8B,CD,82,3B0      [1637]
456 DATA 45,F1,6F,26,0C,CD,7E,48,36A      [1649]
457 DATA 11,1F,53,19,CD,68,4D,18,236      [1483]
458 DATA 82,F5,87,87,C6,04,CD,82,49E      [1220]
459 DATA 45,3E,87,CD,82,45,C1,68,3C7      [1134]
460 DATA 26,0C,CD,7E,48,11,1E,53,247      [1288]
461 DATA 19,7E,B7,28,0A,36,00,3E,1F4      [1626]
462 DATA 20,CD,82,45,C3,B3,4A,E5,459      [1809]
463 DATA 48,68,26,00,29,11,4E,53,1B1      [1488]
464 DATA 19,E5,E5,FD,E1,DD,E1,FD,67C      [1699]
465 DATA 6E,01,26,07,CD,7E,48,FD,32C      [1102]
466 DATA 5E,00,16,00,19,29,11,9F,166      [2218]
467 DATA 52,19,7E,69,26,05,CD,7E,2C8      [1380]
468 DATA 48,11,5F,53,19,06,05,BE,1ED      [1488]
469 DATA 28,08,23,10,FA,CD,BC,4B,331      [1138]
470 DATA 28,13,DD,21,67,4E,AF,CD,36A      [1711]
471 DATA BC,4B,28,09,DD,23,DD,23,338      [1455]
472 DATA 3C,FE,04,20,F2,DD,6E,00,39B      [2040]
473 DATA FD,75,00,DD,66,01,FD,74,427      [1397]
474 DATA 01,E1,36,FF,3E,40,CD,82,3E4      [2206]
475 DATA 45,C3,B3,4A,E5,C5,D5,47,4CB      [1471]
476 DATA 21,4E,53,11,1E,53,AF,F5,2E8      [1440]
477 DATA 7E,DD,BE,00,20,0C,23,7E,2E6      [1834]
478 DATA 2B,DD,BE,01,20,04,1A,B7,2BC      [1810]
479 DATA 20,17,23,23,E5,21,0C,00,18F      [1149]
480 DATA 19,EB,E1,F1,3C,FE,04,20,434      [1308]
481 DATA DE,3E,00,B7,78,D1,C1,E1,4BE      [887]
482 DATA C9,F1,3E,FF,B7,78,D1,C1,5B8      [951]
483 DATA E1,C9,3A,9B,BC,FE,DF,C2,5DA      [1468]
484 DATA 0D,41,01,7E,FA,3E,01,ED,2F3      [1685]
485 DATA 79,CD,2B,4D,DD,21,27,4E,331      [1484]
486 DATA 06,04,CD,1C,4A,21,0F,52,1BF      [1291]
487 DATA CD,78,45,3E,94,CD,82,45,3F0      [1920]
488 DATA 3E,06,CD,82,45,21,56,53,2A2      [2200]
489 DATA CD,78,45,01,08,00,11,73,217      [958]
490 DATA 53,21,56,53,ED,B0,CD,C5,44C      [1085]
491 DATA 48,3E,01,01,7E,FA,ED,79,366      [1712]
492 DATA DD,21,27,4E,CD,E3,49,20,38C      [839]
493 DATA 12,CD,7D,BC,CD,92,BC,01,434      [1990]
494 DATA 7E,FA,AF,ED,79,CD,37,4D,4DE      [2553]
495 DATA C3,0D,41,DD,21,2B,4E,CD,355      [1948]
496 DATA E3,49,28,13,3E,94,CD,82,388      [1825]
497 DATA 45,3E,06,CD,82,45,21,56,294      [1483]
498 DATA 53,CD,68,4D,C3,1B,4C,21,320      [1294]
499 DATA 01,01,CD,75,BB,DD,21,33,330      [842]
500 DATA 4E,CD,E3,49,28,2A,CD,43,3A9      [1548]
501 DATA 4D,CB,6F,28,A9,CB,77,20,3BA      [1705]
502 DATA A5,06,0C,21,73,53,CD,8C,2F7      [1829]
503 DATA BC,30,AE,21,9D,52,11,D7,392      [1682]
504 DATA 00,01,00,00,3E,02,CD,98,1A6      [1535]
505 DATA BC,30,9E,CD,8F,BC,18,99,453      [1641]
506 DATA DD,21,2F,4E,CD,E3,49,CA,43E      [2065]
507 DATA 36,4C,CD,43,4D,CB,6F,CA,3E3      [1562]
508 DATA 36,4C,06,0C,21,73,53,CD,248      [1315]
509 DATA 77,BC,D2,49,4C,21,9D,52,3AA      [1831]
510 DATA CD,83,BC,D2,49,4C,CD,7A,4BA      [1100]
511 DATA BC,D2,49,4C,CD,37,4D,CD,441      [1586]
512 DATA E7,47,CD,95,46,CD,C3,46,4AC      [670]
513 DATA 01,7E,FA,AF,ED,79,C3,B3,504      [1522]
514 DATA 40,CD,2B,4D,DD,21,1B,4E,2EC      [1496]
515 DATA 06,03,CD,1C,4A,21,F0,51,29E      [2000]
516 DATA CD,78,45,CD,C5,48,DD,21,462      [2247]
517 DATA 1B.4E,CD,E3,49,20,06,CD,355      [964]
```

Listing "LABYRINTH"

```
518 DATA 37,4D,C3,0D,41,DD,21,23,2B6      [836]
519 DATA 4E,CD,E3,49,28,01,C9,DD,416      [1998]
520 DATA 21,1F,4E,CD,E3,49,28,DB,38A      [1828]
521 DATA C3,07,40,01,00,40,21,00,16C      [1508]
522 DATA C0,11,23,5E,ED,B0,C9,01,3B9      [981]
523 DATA 00,40,21,23,5E,11,00,C0,1B3      [1462]
524 DATA ED,B0,C9,F3,01,7F,FB,3E,512      [1636]
525 DATA 04,ED,79,0D,ED,78,87,30,393      [1734]
526 DATA FB,0C,AF,ED,79,3E,FF,3D,496      [1197]
527 DATA 00,20,FC,ED,78,FB,C9,FE,543      [2477]
528 DATA 61,D8,FE,7B,D0,D6,20,C9,541      [1337]
529 DATA CD,09,BB,38,FB,06,08,3E,310      [1655]
530 DATA 20,CD,82,45,10,FB,06,08,2CD      [2146]
531 DATA 3E,FE,CD,82,45,10,FB,0E,3E9      [1736]
532 DATA 00,3E,3C,CD,82,45,3E,FE,34A      [2052]
533 DATA CD,82,45,CD,06,BB,CD,5F,44E      [1329]
534 DATA 4D,FE,0D,28,3C,FE,7F,20,359      [2626]
535 DATA 17,79,B7,28,E4,3E,20,CD,37E      [1476]
536 DATA 82,45,3E,FE,CD,82,45,CD,464      [1801]
537 DATA 82,45,0D,36,FF,2B,18,D1,31D      [1595]
538 DATA FE,30,38,CD,FE,5B,30,C9,485      [1521]
539 DATA FE,3A,38,04,FE,41,38,C1,3AC      [1470]
540 DATA D5,5F,3E,08,B9,7B,D1,28,3A7      [1292]
541 DATA B8,CD,82,45,77,23,0C,18,30A      [1166]
542 DATA B0,AF,B9,28,AC,3E,20,CD,417      [1484]
543 DATA 82,45,79,FE,08,28,06,36,2AA      [932]
544 DATA 20,23,3C,18,F6,36,FF,C9,38B      [2115]
545 DATA 00,00,00,00,00,00,00,00,000      [1005]
546 DATA 00,00,00,00,08,00,00,0A,012      [1342]
547 DATA 10,00,00,14,1B,00,00,1E,05D      [977]
548 DATA 21,00,00,01,1C,03,16,1D,074      [1471]
549 DATA 1D,07,12,00,00,07,12,05,054      [935]
550 DATA 18,02,02,05,18,17,17,00,067      [1692]
551 DATA 25,00,14,18,27,00,0A,19,09B      [1435]
552 DATA 26,03,05,19,26,07,09,10,08D      [1328]
553 DATA 1F,00,0C,11,1E,03,07,11,075      [1885]
554 DATA 16,09,0B,19,1E,09,0B,05,07A      [1401]
555 DATA 18,00,12,06,08,03,05,0A,04A      [917]
556 DATA 17,03,05,06,21,0C,18,00,06A      [1148]
557 DATA 0C,55,08,00,30,55,20,00,10E      [1274]
558 DATA 30,FF,00,00,30,55,08,55,211      [1546]
559 DATA 20,55,20,55,20,00,24,FF,22D      [1223]
560 DATA 00,AA,10,FF,00,AA,10,00,273      [610]
561 DATA 00,06,00,06,04,00,04,1A,02E      [1319]
562 DATA 00,18,0B,06,12,0C,02,03,04C      [1376]
563 DATA 09,0D,0A,0F,14,10,01,00,054      [1972]
564 DATA 00,00,00,00,00,01,01,01,003      [1284]
565 DATA 01,01,01,01,01,02,02,02,00B      [1185]
566 DATA 02,02,03,03,03,03,03,03,016      [1116]
567 DATA 03,03,04,04,04,04,04,04,01E      [1207]
568 DATA 04,04,05,05,05,05,05,05,026      [1711]
569 DATA 05,05,06,06,06,06,06,07,02F      [1293]
570 DATA 07,07,07,07,07,07,07,08,039      [1810]
571 DATA 08,08,08,08,05,06,07,08,03A      [1002]
572 DATA 09,0A,0B,0C,0D,0E,0F,10,064      [1747]
573 DATA 11,12,13,14,15,16,17,18,0A4      [1539]
574 DATA 00,00,00,00,00,00,00,00,000      [1005]
575 DATA 00,00,00,00,00,00,00,00,000      [1005]
576 DATA 00,00,00,00,00,00,00,00,000      [1005]
577 DATA 00,00,00,00,00,00,00,00,000      [1005]
578 DATA 00,00,00,00,00,00,00,00,000      [1005]
579 DATA 05,06,07,08,09,0A,0B,0C,044      [1333]
580 DATA 0D,0E,0F,10,11,12,13,14,084      [1511]
581 DATA 15,16,17,18,80,00,41,4E,169      [1520]
582 DATA 4C,45,49,54,55,4E,47,20,238      [1343]
583 DATA 53,50,49,45,4C,45,52,20,234      [1529]
584 DATA 20,20,44,49,53,4B,45,54,204      [1496]
585 DATA 54,45,20,20,45,4E,44,45,1F5      [1152]
586 DATA 20,20,20,20,20,20,80,02,142      [1136]
587 DATA 3A,3A,3A,3A,3A,2A,20,20,18C      [1457]
588 DATA 2B,20,20,20,2B,20,20,20,116      [1168]
589 DATA 2B,20,20,20,2B,20,20,20,116      [1168]
590 DATA 2B,3B,3B,3B,3B,3B,80,17,1E9      [1516]
591 DATA 3D,3D,3D,3D,3D,2A,20,20,19B      [1991]
592 DATA 2B,20,20,20,2B,20,20,20,116      [1168]
593 DATA 2B,20,20,20,2B,20,20,20,116      [1168]
594 DATA 2B,3C,3C,3C,3C,3C,FF,3A,290      [2418]
595 DATA 3B,3A,3B,3A,3B,3A,3B,2C,1C6      [1711]
596 DATA 2C,20,20,20,20,2D,2D,20,126      [1777]
597 DATA 20,20,20,20,20,2D,2D,20,11A      [1119]
598 DATA 20,20,20,20,20,2D,2D,3D,137      [1033]
599 DATA 3C,3D,3C,3D,3C,3D,3C,1F,1C6      [1617]
600 DATA 08,0B,0C,20,02,0D,0F,25,082      [1008]
601 DATA 02,0D,17,8C,01,53,50,49,19F      [2004]
602 DATA 45,4C,41,4E,4C,45,49,54,24E      [1827]
603 DATA 55,4E,47,81,03,4A,45,44,241      [1156]
```

Listing "LABYRINTH"


```
604 DATA 45,52,20,20,44,45,52,20,1D2      [1331]
605 DATA 20,4D,41,58,49,4D,41,4C,229      [1463]
606 DATA 20,56,49,45,52,20,53,50,219      [1582]
607 DATA 49,45,4C,45,52,20,48,41,21A      [2051]
608 DATA 54,81,04,44,49,45,20,41,20C      [2098]
609 DATA 55,46,47,41,42,45,2F,20,1F9      [1005]
610 DATA 49,4E,20,45,49,4E,45,52,22A      [1776]
611 DATA 20,41,4C,54,45,4E,20,50,204      [1760]
612 MEMORY &7FFF                          [150]
613 adr=&8000                             [628]
614 FOR zeile=100 TO 611                  [1653]
615 p=0                                   [257]
616 FOR i=1 TO 8                          [462]
617 READ w$:w=VAL("&"+w$)                 [1166]
618 POKE adr,w:adr=adr+1                  [1068]
619 p=p+w                                 [296]
620 NEXT                                  [350]
621 READ pr$:pr=VAL("&"+pr$)              [823]
622 IF p<>pr THEN PRINT"Fehler in Zeile";z [1818]
eile:f=1
623 NEXT                                  [350]
624 IF f=1 THEN STOP                      [910]
625 SAVE"LAB1.BIN",b,&8000,&1000          [2008]
626 PRINT"Fertig!!!"                      [1187]
```

```
96 '*************************          [894]
97 '  Listing 2 :  LAB2.LAD              [1801]
98 '*************************          [894]
99 '                                     [117]
100 DATA 59,52,41,4D,49,44,45,81,28C     [1936]
101 DATA 05,46,55,45,4E,46,20,20,1B9     [1062]
102 DATA 53,43,48,41,45,54,5A,45,257     [937]
103 DATA 2F,20,20,44,49,45,20,20,181     [2189]
104 DATA 41,4E,20,20,44,45,52,20,1CA     [1573]
105 DATA 53,45,49,54,45,81,06,41,242     [685]
106 DATA 4E,47,45,5A,45,49,47,54,25D     [1658]
107 DATA 20,57,45,52,44,45,4E,2F,214     [1398]
108 DATA 20,45,49,4E,5A,55,53,41,23F     [1329]
109 DATA 4D,4D,45,4C,4E,2E,81,07,22F     [1184]
110 DATA 44,45,52,20,20,53,50,49,207     [1740]
111 DATA 45,4C,45,52,20,20,4B,41,1F4     [1921]
112 DATA 4E,4E,20,20,45,4E,54,57,21A     [1238]
113 DATA 45,44,45,52,20,20,53,45,1F8     [1376]
114 DATA 49,4E,45,4E,81,08,53,54,25A     [1463]
115 DATA 45,49,4E,20,20,20,41,55,1D2     [1343]
116 DATA 46,20,20,45,49,4E,20,20,1A2     [1455]
117 DATA 42,45,4E,41,43,48,42,41,224     [1793]
118 DATA 52,54,45,53,20,20,46,45,209     [1496]
119 DATA 4C,44,81,09,53,45,54,5A,260     [1412]
120 DATA 45,4E,2F,20,57,45,4E,4E,21A     [1160]
121 DATA 20,44,41,48,49,4E,20,44,1E8     [2436]
122 DATA 45,52,20,57,45,47,20,46,200     [1468]
123 DATA 52,45,49,20,49,53,54,2F,21F     [873]
124 DATA 81,0A,4F,44,45,52,20,20,1F5     [1999]
125 DATA 20,45,52,20,20,20,56,45,1B2     [1443]
126 DATA 52,53,43,48,49,45,42,54,254     [1115]
127 DATA 20,20,20,45,49,4E,45,20,1A1     [1211]
128 DATA 20,47,41,4E,5A,45,81,0B,221     [1531]
129 DATA 53,54,45,49,4E,52,45,49,263     [1199]
130 DATA 48,45,2F,20,20,57,4F,42,1E4     [1475]
131 DATA 45,49,20,20,44,41,53,20,1C6     [1912]
132 DATA 20,46,52,45,49,45,20,20,1CB     [626]
133 DATA 46,45,4C,44,81,0C,44,55,241     [1215]
134 DATA 52,43,48,20,20,44,45,4E,1F4     [1550]
135 DATA 20,20,53,54,45,49,4E,20,1E3     [1788]
136 DATA 20,45,52,53,45,54,5A,54,251     [1541]
137 DATA 20,57,49,52,44,2F,20,44,1E9     [1834]
138 DATA 45,52,81,0D,4E,45,42,45,23F     [863]
139 DATA 4E,20,44,45,4D,20,53,50,207     [1842]
140 DATA 49,45,4C,46,45,4C,44,20,215     [1174]
141 DATA 41,4E,47,45,5A,45,49,47,24A     [1334]
142 DATA 54,20,57,49,52,44,2E,81,259     [1377]
143 DATA 0E,53,49,45,47,45,52,20,1ED     [1807]
144 DATA 49,53,54,2F,20,57,45,52,22D     [1938]
145 DATA 20,53,45,49,4E,45,20,46,1FA     [1190]
146 DATA 55,45,4E,46,20,53,43,48,22C     [1770]
147 DATA 41,45,54,5A,45,81,0F,41,24A     [1357]
148 DATA 4C,53,20,20,20,20,45,52,1B6     [1906]
149 DATA 53,54,45,52,20,20,20,45,1E3     [1500]
150 DATA 49,4E,47,45,53,41,4D,4D,251     [1375]
151 DATA 45,4C,54,20,48,41,54,20,202     [1890]
152 DATA 55,4E,44,81,10,5A,55,20,247     [1518]
```

Listing "LABYRINTH"

```
153 DATA 20,53,45,49,4E,45,4D,20,201     [1149]
154 DATA 48,45,49,4D,41,54,46,45,243     [1164]
155 DATA 4C,44,20,5A,55,52,55,45,24B     [1325]
156 DATA 43,4B,47,45,4B,45,48,52,244     [859]
157 DATA 54,81,11,49,53,54,2E,84,288     [2074]
158 DATA 13,43,4F,50,59,52,49,47,230     [2054]
159 DATA 48,54,20,31,39,38,37,20,1B5     [1170]
160 DATA 20,43,41,52,53,54,45,4E,230     [841]
161 DATA 20,53,43,48,41,41,52,FF,2D1     [1592]
162 DATA 9C,01,45,4E,44,4D,45,4E,254     [810]
163 DATA 55,45,9A,04,4E,45,55,45,265     [582]
164 DATA 20,20,50,41,52,54,49,45,205     [1258]
165 DATA 9E,08,45,4E,44,45,FF,91,352     [1423]
166 DATA 01,44,49,53,4B,45,54,54,219     [1777]
167 DATA 45,4E,4D,45,4E,55,45,93,2A0     [1832]
168 DATA 04,53,50,49,45,4C,53,54,228     [1490]
169 DATA 41,4E,44,92,0A,4C,4F,41,24B     [1065]
170 DATA 44,9A,0A,53,41,56,45,FF,316     [1726]
171 DATA 89,01,53,50,49,45,4C,45,24C     [1398]
172 DATA 52,4D,45,4E,55,45,96,04,266     [914]
173 DATA 3A,96,08,3B,96,0C,3C,96,287     [1398]
174 DATA 10,3D,FF,90,10,48,41,54,2C9     [1873]
175 DATA 93,12,47,45,57,4F,4E,4E,273     [1497]
176 DATA 45,4E,2E,9B,16,46,45,55,252     [2205]
177 DATA 45,52,88,0E,FF,00,53,50,2CF     [1996]
178 DATA 49,45,4C,45,52,31,FF,00,2A1     [1567]
179 DATA 00,00,53,50,49,45,4C,45,1C2     [1584]
180 DATA 52,32,FF,00,00,00,53,50,226     [1662]
181 DATA 49,45,4C,45,52,33,FF,00,2A3     [1813]
182 DATA 00,00,53,50,49,45,4C,45,1C2     [1584]
183 DATA 52,34,FF,00,00,00,00,00,185     [1732]
184 DATA 00,00,00,00,00,00,00,00,000     [1005]
185 DATA 00,00,02,00,00,00,00,00,002     [1409]
186 DATA 00,00,00,00,00,00,00,00,000     [1005]
187 DATA 00,00,00,00,00,00,00,00,000     [1005]
188 DATA 00,00,00,00,00,00,00,00,000     [1005]
189 DATA 00,00,00,00,00,00,00,00,000     [1005]
190 DATA 00,00,00,00,00,00,06,00,006     [849]
191 DATA 00,00,00,00,00,00,00,00,000     [1005]
192 DATA 00,00,08,00,00,00,00,00,008     [1470]
193 DATA 00,00,00,00,00,00,00,00,000     [1005]
194 DATA 00,00,00,00,00,00,00,00,000     [1005]
195 DATA 00,00,00,00,00,00,00,00,000     [1005]
196 DATA 00,00,00,00,00,00,00,00,000     [1005]
197 DATA 00,00,00,00,00,00,00,00,000     [1005]
198 DATA 00,00,00,00,00,00,00,00,000     [1005]
199 DATA 00,00,00,00,00,00,00,00,000     [1005]
200 DATA 00,00,00,00,00,00,00,00,000     [1005]
201 DATA 00,00,00,00,00,00,00,00,000     [1005]
202 DATA 00,00,00,00,00,00,00,00,000     [1005]
203 DATA 00,00,00,00,00,00,00,00,000     [1005]
204 DATA 00,00,00,00,00,00,00,00,000     [1005]
205 DATA 00,00,00,00,00,00,00,00,000     [1005]
206 DATA 00,00,00,00,00,00,53,50,0A3     [1316]
207 DATA 49,45,4C,20,20,20,FF,00,239     [1567]
208 DATA 00,00,00,00,00,00,00,00,000     [1005]
209 DATA 00,00,00,00,00,00,00,00,000     [1005]
210 DATA 00,00,00,20,20,20,20,0A0        [909]
211 DATA 20,20,20,2E,4C,41,42,00,15D     [1815]
212 DATA 00,00,00,00,00,00,00,00,000     [1005]
213 DATA 00,00,00,00,00,00,00,00,000     [1005]
214 DATA 00,00,00,00,00,00,00,00,000     [1005]
215 DATA 00,00,00,00,00,00,00,00,000     [1005]
216 DATA 00,00,00,00,00,00,00,00,000     [1005]
217 DATA 00,00,00,00,00,00,00,00,000     [1005]
218 DATA 00,00,00,30,30,30,30,20,0E0     [1676]
219 DATA 00,20,00,20,00,20,00,20,080     [1135]
220 DATA 00,20,00,30,30,30,30,00,0E0     [1979]
221 DATA 00,00,00,00,00,00,00,00,000     [1005]
222 DATA 00,00,00,30,30,30,30,00,0C0     [1470]
223 DATA 10,00,10,00,10,00,10,00,040     [1656]
224 DATA 10,00,10,20,00,20,00,20,080     [1670]
225 DATA 00,20,00,20,00,20,00,20,080     [1135]
226 DATA 00,20,00,00,00,00,00,00,020     [1555]
227 DATA 00,00,00,00,00,00,00,00,000     [1005]
228 DATA 00,00,00,00,10,00,10,00,020     [1265]
229 DATA 10,00,10,00,10,00,10,00,040     [1656]
230 DATA 10,00,10,20,00,20,00,20,080     [1670]
231 DATA 00,20,00,20,00,20,00,30,090     [1099]
232 DATA 30,30,30,00,00,00,00,00,090     [1712]
233 DATA 00,00,00,00,00,00,00,30,030     [1222]
234 DATA 30,30,30,00,10,00,10,00,0B0     [1518]
235 DATA 10,00,10,00,10,00,10,30,070     [1515]
236 DATA 30,30,30,80,00,80,00,80,210     [1493]
237 DATA 00,80,00,80,00,80,00,80,200     [1332]
238 DATA 00,80,00,00,40,00,40,00,100     [1105]
239 DATA 40,00,40,00,40,00,40,00,100     [1470]
```

Listing "LABYRINTH"

```
240 DATA 40,00,40,C0,C0,00,00,00,200      [1493]
241 DATA 00,00,00,00,00,00,00,00,000      [1005]
242 DATA 00,00,00,00,00,00,00,00,000      [1005]
243 DATA 00,00,00,00,00,00,00,00,000      [1005]
244 DATA 00,C0,C0,00,00,00,00,00,180      [1551]
245 DATA 00,00,00,00,00,00,00,40,040      [1124]
246 DATA 00,00,00,00,00,00,00,00,000      [1005]
247 DATA 00,00,00,00,00,00,00,40,040      [1124]
248 DATA 00,80,00,51,00,A2,A2,A2,2B7      [1312]
249 DATA A2,A2,A2,A2,A2,A2,A2,51,4BF      [1066]
250 DATA 00,00,00,00,A2,51,A2,00,195      [1758]
251 DATA A2,00,A2,00,A2,00,A2,00,288      [1327]
252 DATA A2,00,00,51,00,A2,A2,00,237      [1330]
253 DATA A2,51,00,A2,00,A2,A2,F3,3CC      [1805]
254 DATA A2,00,00,51,00,A2,A2,00,237      [1330]
255 DATA A2,51,00,00,A2,A2,A2,51,32A      [2084]
256 DATA 00,00,00,A2,A2,A2,A2,A2,32A      [1756]
257 DATA A2,F3,A2,00,A2,00,A2,00,37B      [1507]
258 DATA A2,00,00,F3,A2,A2,00,F3,3CC      [1064]
259 DATA 00,00,A2,00,A2,A2,A2,51,2D9      [1655]
260 DATA 00,00,00,51,00,A2,A2,A2,237      [1569]
261 DATA 00,F3,00,A2,A2,A2,A2,51,3CC      [1084]
262 DATA 00,00,00,F3,A2,A2,A2,00,2D9      [898]
263 DATA A2,00,A2,51,00,51,00,51,237      [1489]
264 DATA 00,00,00,51,00,A2,A2,A2,237      [1569]
265 DATA A2,51,00,A2,A2,A2,A2,51,3CC      [1002]
266 DATA 00,00,00,51,00,A2,A2,A2,237      [1569]
267 DATA A2,51,A2,00,A2,A2,A2,51,3CC      [1111]
268 DATA 00,00,00,0C,0C,0C,0C,0C,03C      [1632]
269 DATA 0C,0C,0C,0C,0C,0C,0C,0C,060      [1792]
270 DATA 0C,0C,0C,F0,F0,F0,F0,F0,4D4      [950]
271 DATA F0,F0,F0,F0,F0,F0,F0,F0,780      [2130]
272 DATA F0,F0,F0,CC,CC,CC,CC,CC,6CC      [2658]
273 DATA CC,CC,CC,CC,CC,CC,CC,CC,660      [1252]
274 DATA CC,CC,CC,30,30,30,30,30,354      [652]
275 DATA 30,30,30,30,30,30,30,30,180      [1618]
276 DATA 30,30,30,3C,29,03,3C,03,137      [1975]
277 DATA 16,03,29,3C,3C,03,29,03,0E9      [1307]
278 DATA 03,3C,29,CF,CF,CF,CF,CF,473      [1808]
279 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
280 DATA CF,CF,CF,00,04,00,04,00,275      [1951]
281 DATA 20,00,20,A0,AA,88,AA,55,311      [1261]
282 DATA 00,54,00,45,00,8A,8A,8A,237      [2054]
283 DATA 8A,CF,8A,8A,8A,8A,8A,8A,495      [1047]
284 DATA 8A,00,00,CF,00,8A,8A,8A,2F7      [885]
285 DATA 8A,CF,00,8A,8A,8A,8A,CF,450      [1578]
286 DATA 00,00,00,45,00,8A,8A,8A,1E3      [1677]
287 DATA 00,8A,00,8A,00,8A,8A,45,26D      [1271]
288 DATA 00,00,00,CF,00,8A,8A,8A,26D      [1830]
289 DATA 8A,8A,8A,8A,8A,8A,8A,CF,495      [1994]
290 DATA 00,00,00,CF,8A,8A,8A,00,8A,26D   [1585]
291 DATA 00,CF,00,8A,00,8A,00,CF,2B2      [1031]
292 DATA 8A,00,00,CF,8A,8A,00,8A,2F7      [724]
293 DATA 00,CF,00,8A,00,8A,00,8A,26D      [419]
294 DATA 00,00,00,45,00,8A,8A,8A,1E3      [1677]
295 DATA 00,8A,8A,8A,8A,8A,8A,45,381      [1797]
296 DATA 00,00,00,8A,8A,8A,8A,8A,2B2      [637]
297 DATA 8A,CF,8A,8A,8A,8A,8A,8A,495      [1047]
298 DATA 8A,00,00,45,00,45,00,45,159      [1263]
299 DATA 00,45,00,45,00,45,00,45,114      [1639]
300 DATA 00,00,00,CF,8A,00,8A,00,1E3      [1341]
301 DATA 8A,00,8A,00,8A,8A,8A,45,2F7      [950]
302 DATA 00,00,00,8A,8A,8A,8A,8A,2B2      [637]
303 DATA 8A,CF,00,8A,8A,8A,8A,8A,40B      [1597]
304 DATA 8A,00,00,8A,00,8A,00,8A,228      [1133]
305 DATA 00,8A,00,8A,00,8A,00,CF,26D      [1447]
306 DATA 8A,00,00,8A,8A,CF,8A,CF,3C6      [1557]
307 DATA 8A,8A,8A,8A,8A,8A,8A,8A,450      [1990]
308 DATA 8A,00,00,8A,8A,8A,8A,CF,381      [1095]
309 DATA 8A,CF,8A,CF,8A,8A,8A,8A,4DA      [1604]
310 DATA 8A,00,00,45,00,8A,8A,8A,26D      [1829]
311 DATA 8A,8A,8A,8A,8A,8A,8A,45,40B      [2042]
312 DATA 00,00,00,CF,00,8A,8A,8A,26D      [1830]
313 DATA 8A,CF,00,8A,00,8A,00,8A,2F7      [1937]
314 DATA 00,00,00,45,00,8A,8A,8A,1E3      [1677]
315 DATA 8A,8A,8A,8A,8A,8A,8A,45,40B      [2042]
316 DATA 8A,00,00,CF,00,8A,8A,8A,2F7      [885]
317 DATA 8A,CF,00,8A,8A,8A,8A,8A,40B      [1597]
318 DATA 8A,00,00,45,00,8A,8A,8A,26D      [1829]
319 DATA 00,45,00,00,8A,8A,8A,45,228      [1387]
320 DATA 00,00,00,CF,8A,45,00,45,1E3      [1471]
321 DATA 00,45,00,45,00,45,00,45,114      [1639]
322 DATA 00,00,00,8A,8A,8A,8A,8A,2B2      [637]
323 DATA 8A,8A,8A,8A,8A,8A,8A,45,40B      [2042]
324 DATA 00,00,00,8A,8A,8A,8A,8A,2B2      [637]
325 DATA 8A,8A,8A,8A,8A,45,00,45,33C      [1850]
326 DATA 00,00,00,8A,8A,8A,8A,8A,2B2      [637]
```

Listing "LABYRINTH"

```
327 DATA 8A,8A,8A,CF,8A,CF,8A,8A,4DA      [2753]
328 DATA 8A,00,00,8A,8A,8A,8A,8A,33C      [992]
329 DATA 8A,45,00,8A,8A,8A,8A,8A,381      [705]
330 DATA 8A,00,00,8A,8A,8A,8A,8A,33C      [992]
331 DATA 8A,45,00,45,00,45,00,45,19E      [1047]
332 DATA 00,00,00,CF,8A,00,8A,00,1E3      [1341]
333 DATA 8A,45,00,8A,00,8A,00,CF,2B2      [633]
334 DATA 8A,00,00,16,68,03,68,29,19C      [1471]
335 DATA 42,16,68,3C,68,16,42,03,1BF      [1380]
336 DATA 68,3C,68,CF,CA,CF,CA,CF,50D      [1756]
337 DATA CA,CF,CA,CF,CA,CF,CA,CF,664      [1973]
338 DATA CA,CF,CA,3C,29,29,16,03,30A      [1777]
339 DATA 03,3C,03,3C,3C,03,3C,3C,135      [1321]
340 DATA 03,C0,C0,CF,CF,CF,CF,CF,58E      [2540]
341 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
342 DATA CF,C0,C0,3C,42,03,68,3C,374      [1510]
343 DATA 68,03,68,03,42,29,42,16,199      [2106]
344 DATA 68,C0,C0,CF,CF,CF,CF,CF,5F3      [1041]
345 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
346 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
347 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
348 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
349 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
350 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
351 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
352 DATA CF,CF,CF,CF,8E,4D,CF,CF,5B5      [1140]
353 DATA 0C,0C,CF,CF,0C,0C,CF,0C,2A9      [900]
354 DATA 0C,0C,0C,9F,95,6A,6F,9F,2D0      [635]
355 DATA 95,6A,6F,CF,3F,3F,CF,CF,459      [824]
356 DATA 9E,6D,CF,CF,3C,3C,CF,9E,48E      [1902]
357 DATA 9E,6D,6D,9E,9E,6D,6D,CF,45D      [1672]
358 DATA 9E,6D,CA,CF,0C,0C,C0,CF,44B      [2999]
359 DATA 4D,8E,C0,CF,4D,84,C5,8B,48B      [1041]
360 DATA 42,8B,47,CF,DA,E5,CF,CF,540      [1586]
361 DATA F0,F0,CF,CF,F0,F0,CF,F0,71D      [3106]
362 DATA F0,F0,F0,9F,95,6A,6F,9F,57C      [2483]
363 DATA 95,6A,6F,CF,3F,3F,CF,CF,459      [824]
364 DATA CB,C7,CF,CF,C3,C3,CF,CB,650      [1172]
365 DATA CB,C7,C7,CB,CB,C7,C7,CF,64C      [2002]
366 DATA CB,C7,CA,CF,F0,F0,C0,CF,69A      [2557]
367 DATA E5,D0,C0,CF,E0,DA,C5,8B,64E      [1359]
368 DATA 47,8B,47,CF,DE,ED,CF,CF,551      [2257]
369 DATA FC,FC,CF,CF,FC,FC,CF,FC,759      [854]
370 DATA FC,FC,FC,9F,95,6A,6F,9F,5A0      [786]
371 DATA 95,6A,6F,CF,3F,3F,CF,CF,459      [824]
372 DATA DF,EF,CF,CF,FF,FF,CF,DF,718      [2028]
373 DATA DF,EF,EF,DF,DF,EF,EF,CF,728      [1201]
374 DATA DF,EF,CA,CF,CC,CC,C0,CF,68E      [1821]
375 DATA CD,C4,C0,CF,C8,CE,C5,8B,606      [1679]
376 DATA 47,8B,47,CF,9A,65,CF,CF,485      [2259]
377 DATA 30,30,CF,CF,30,30,CF,30,35D      [1472]
378 DATA 30,30,30,9F,95,6A,6F,9F,33C      [1777]
379 DATA 95,6A,6F,CF,3F,3F,CF,CF,459      [824]
380 DATA 8B,47,CF,CF,03,03,CF,8B,3D0      [1888]
381 DATA 8B,47,47,8B,8B,47,47,CF,38C      [2080]
382 DATA 8B,47,CA,CF,30,30,C0,CF,45A      [1723]
383 DATA 65,90,C0,CF,60,9A,C5,8B,4CE      [1142]
384 DATA 47,8B,47,6D,6D,9E,9E,6D,39C      [896]
385 DATA 6D,9E,9E,9E,CF,CF,6D,9E,4F0      [1962]
386 DATA 9A,47,6D,9E,9A,47,6D,9E,3D8      [2211]
387 DATA 30,03,6D,9E,30,03,6D,CF,2AD      [1680]
388 DATA 38,16,CF,CF,30,03,CF,CF,3BD      [2270]
389 DATA 30,03,CF,CF,30,03,CF,CF,3A2      [2279]
390 DATA 30,03,CF,CF,9A,47,CF,CF,450      [1980]
391 DATA 9A,47,CF,CF,21,03,CF,CF,441      [1530]
392 DATA CF,CF,CF,CF,CF,4D,CF,CF,5F6      [1635]
393 DATA CF,4D,CF,CF,8C,4C,CF,CE,52F      [2315]
394 DATA 8C,4C,CD,CC,08,4C,CD,0C,39E      [2131]
395 DATA 00,04,0C,08,00,00,4D,8C,0F1      [1702]
396 DATA 00,04,CD,8C,00,4C,CD,CC,342      [1391]
397 DATA 08,4C,CD,8C,00,4C,CD,8C,352      [1245]
398 DATA 0C,4C,CF,0C,CC,4D,CF,4D,368      [2398]
399 DATA CC,0C,CF,CF,CF,8E,CF,CF,571      [1652]
400 DATA CF,CF,CF,CF,8A,4D,CF,CF,5B1      [2151]
401 DATA DB,E5,CF,CF,DB,E5,CF,CF,6BC      [984]
402 DATA DB,E5,CF,CF,DB,E5,CF,CF,6BC      [984]
403 DATA DB,E5,CF,CF,DB,E5,CF,CF,6BC      [984]
404 DATA F3,F0,CF,CF,F3,F0,CF,CF,702      [1143]
405 DATA F2,F0,CF,DB,F2,F0,E5,DB,72E      [1059]
406 DATA F2,F0,E5,DB,F2,F0,E5,DB,744      [2312]
407 DATA F2,F0,E5,CF,F3,F0,CF,CF,717      [1492]
408 DATA CF,CF,CF,CF,8E,6D,CF,CF,5D5      [1638]
409 DATA 9A,47,CF,CF,9A,47,CF,CF,4FE      [982]
410 DATA 9A,47,CF,CF,9A,47,CF,CF,4FE      [982]
411 DATA 30,03,CF,CF,30,03,CF,CF,3A2      [2279]
412 DATA 30,21,CF,9A,30,21,47,9A,2EC      [1420]
413 DATA 18,34,47,9A,24,09,47,9A,23B      [863]
```

Listing "LABYRINTH"

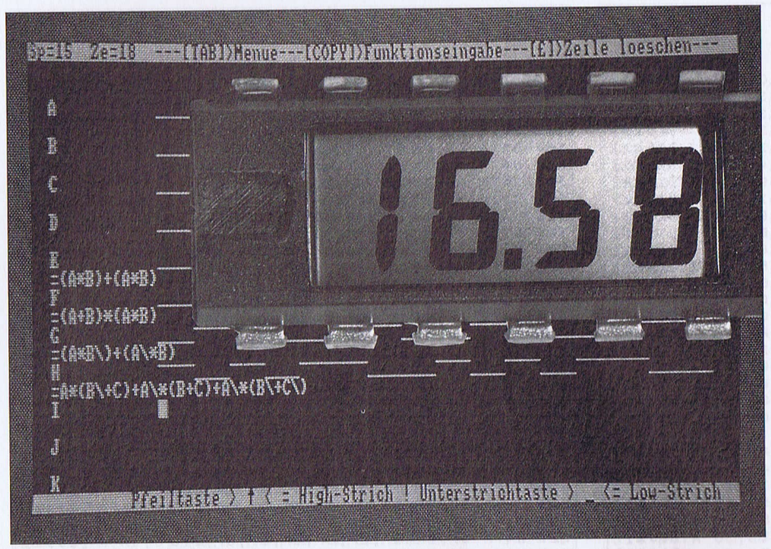

```
414 DATA 24,09,47,9A,24,09,47,9A,21C      [1805]
415 DATA 18,34,47,CF,30,03,CF,CF,333      [1283]
416 DATA CF,CF,CF,CF,CA,CF,CF,CF,673      [2179]
417 DATA CA,CF,CF,CF,8F,CF,CF,CF,633      [2323]
418 DATA CA,CF,CF,CF,85,E7,CF,CF,641      [2541]
419 DATA 85,E7,CF,CF,85,E7,CF,CF,614      [1295]
420 DATA C0,4F,CF,CA,85,F3,CF,CA,5B9      [1549]
421 DATA 0F,5B,CF,CA,4A,5B,CF,CA,441      [1158]
422 DATA 85,A7,CF,CA,85,5B,CF,CA,53E      [1799]
423 DATA 85,F3,CF,CF,C0,E7,CF,CF,65B      [1654]
424 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
425 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
426 DATA CF,CF,CF,03,03,03,03,12,28B      [1453]
427 DATA 30,30,21,8B,30,30,47,8B,23E      [1554]
428 DATA 03,03,47,8B,03,03,47,8B,1B0      [1648]
429 DATA 24,CE,47,8B,4D,18,47,8B,2FB      [2604]
430 DATA 9A,8C,47,8B,4D,9A,47,8B,3B1      [1787]
431 DATA 24,CC,47,8B,03,03,47,CF,2DE      [1914]
432 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
433 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
434 DATA 9A,CF,47,CF,30,CF,03,CF,450      [1068]
435 DATA 9A,21,47,CF,CF,21,47,CF,3D7      [1052]
436 DATA CF,30,CF,CF,CF,10,CF,CF,51A      [1653]
437 DATA 8A,10,65,CF,8A,4D,30,CF,3A4      [2166]
438 DATA 04,CF,65,CF,04,CF,CF,8A,433      [1944]
439 DATA 4D,CF,CF,8E,CF,CF,CF,CF,5B5      [2125]
440 DATA CF,CF,CF,CF,DF,CF,CF,CF,688      [1936]
441 DATA FE,EF,CF,DF,EC,FD,CF,DE,731      [651]
442 DATA 88,DC,CF,EC,44,FD,EF,EC,63B      [2352]
443 DATA 44,FD,EF,EE,DC,FF,EF,FE,6E6      [1087]
444 DATA DC,FF,EF,FF,FD,FF,EF,FF,7B3      [1148]
445 DATA FF,FF,EF,FF,FF,FF,EF,DF,7B8      [1386]
446 DATA FF,FF,CF,DF,FF,FF,CF,CF,748      [2644]
447 DATA FF,EF,CF,CF,DF,CF,CF,CF,6D8      [734]
448 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
449 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
450 DATA CF,CF,CF,CF,12,47,CF,9A,4FE      [2500]
451 DATA CF,9A,CF,47,CF,CF,47,65,4C9      [1528]
452 DATA CF,CF,65,8B,CF,CF,8B,9A,551      [2438]
453 DATA CF,CF,9A,8B,CF,47,8B,8E,4F2      [2482]
454 DATA 18,9A,9A,1C,2C,CF,47,1C,2C6      [1689]
455 DATA 2C,CF,CF,8E,4D,CF,CF,CF,512      [3162]
456 DATA CF,CF,CF,CF,30,30,CF,CF,53A      [1434]
457 DATA 8B,47,CF,CF,8B,47,CF,65,476      [1916]
458 DATA 8B,47,9A,21,03,03,12,21,1C6      [1484]
459 DATA 03,03,12,21,03,03,12,65,0B6      [1452]
460 DATA 8B,47,9A,CF,8B,47,CF,CF,4AB      [1325]
461 DATA 8B,47,CF,CF,8B,47,CF,CF,4E0      [2110]
462 DATA 8B,47,CF,CF,8B,47,CF,CF,4E0      [2110]
463 DATA 8B,47,CF,CF,30,30,CF,CF,46E      [1462]
464 DATA CF,CF,CF,CF,DE,ED,CF,CF,6A5      [2606]
465 DATA 7C,BC,CF,DE,BC,7C,ED,9E,5A8      [2775]
466 DATA BC,54,6D,7C,28,14,BC,FC,3ED      [1249]
467 DATA 3C,3C,FC,3C,0C,0C,3C,AC,2B0      [1930]
468 DATA AC,5C,5C,2C,1C,2C,1C,AC,2A0      [1908]
469 DATA 0C,0C,5C,2C,0C,0C,1C,FC,1D0      [1225]
470 DATA 1C,2C,FC,3C,2C,1C,3C,FC,300      [2340]
471 DATA EC,DC,FC,3C,6C,9C,3C,CF,513      [867]
472 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
473 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
474 DATA CF,CF,CF,CF,CF,8E,C5,CF,62D      [1726]
475 DATA CF,26,62,CF,CF,4D,48,0C,396      [1422]
476 DATA 0C,4D,48,0C,0C,67,48,33,19B      [1960]
477 DATA 33,67,48,4D,4D,26,62,4D,251      [1999]
478 DATA 4D,8E,C5,CF,CF,CF,CF,CF,5AB      [2093]
479 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
480 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
481 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
482 DATA CF,CF,4D,CF,CF,8E,9B,CF,581      [1691]
483 DATA 8E,67,9B,4D,0C,33,9B,4D,304      [1645]
484 DATA 0C,19,9B,8E,0C,19,67,CF,2A9      [1446]
485 DATA 0C,33,CF,CF,8E,67,CF,CF,470      [1373]
486 DATA 0C,33,CF,CF,CF,CF,CF,CF,519      [950]
487 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
488 DATA CF,CF,CF,CF,8F,CF,CF,CF,638      [2258]
489 DATA 0F,45,CF,8F,1E,00,CF,8F,32E      [2699]
490 DATA 3C,00,CF,0F,28,00,45,4A,1D1      [1769]
491 DATA 00,40,45,4A,80,C0,45,4A,29E      [856]
492 DATA 20,60,45,0F,00,00,45,8F,1A8      [899]
493 DATA 40,0A,CF,8F,0A,0A,CF,CF,35A      [1906]
494 DATA 0A,45,CF,CF,CF,CF,CF,CF,529      [2239]
495 DATA 0A,45,CF,CF,8F,CF,CF,CF,4E9      [2291]
496 DATA CF,CF,CF,CF,33,36,CF,CF,543      [1212]
497 DATA 9B,6D,CF,9B,9B,6D,67,67,448      [2283]
498 DATA 33,39,9B,67,9B,6D,9B,67,378      [1430]
499 DATA 9B,6D,9B,67,33,6C,9B,67,3AB      [1375]
```

Listing "LABYRINTH"

```
500 DATA 33,36,9B,9B,33,36,67,CF,33E      [1943]
501 DATA 33,36,CF,CF,33,36,CA,CF,409      [1969]
502 DATA 33,36,C0,CF,33,36,C0,CF,3F0      [1567]
503 DATA 33,9C,C5,CF,9B,C8,CF,CF,564      [2245]
504 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
505 DATA CF,4D,CF,CF,8E,8E,CF,CF,574      [1507]
506 DATA CF,4D,CF,CF,33,32,65,9B,41F      [1544]
507 DATA 33,33,30,CF,64,46,47,CF,325      [1795]
508 DATA 64,46,47,CF,64,46,47,CF,380      [1501]
509 DATA 64,46,47,CF,64,46,47,CF,380      [1501]
510 DATA 64,46,47,CF,64,46,47,CF,380      [1501]
511 DATA 64,46,47,CF,30,03,47,CF,309      [2379]
512 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
513 DATA CF,CF,CF,CF,CF,CF,CF,8A,633      [1727]
514 DATA 45,CF,CF,CF,00,4D,CF,CF,49D      [1893]
515 DATA CF,04,CF,CF,CF,20,4D,CF,47C      [1668]
516 DATA 9A,03,4D,CF,9A,47,04,CF,36D      [2133]
517 DATA 21,47,8E,CF,21,CF,8A,9A,3D9      [729]
518 DATA 03,CF,8A,9A,47,CF,CF,21,3FC      [921]
519 DATA 47,CF,CF,8B,CF,CF,CF,CF,5AC      [1966]
520 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
521 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
522 DATA CF,CF,CF,00,00,00,00,04,271      [1880]
523 DATA 0C,0C,19,04,0C,0C,19,04,06A      [1151]
524 DATA 0C,0C,19,04,0C,0C,19,04,06A      [1151]
525 DATA 0C,0C,19,04,0C,0C,19,11,077      [952]
526 DATA 33,33,33,CF,CF,CF,CF,CF,4A4      [1481]
527 DATA CF,CF,CF,CF,CF,CF,CF,CF,678      [1691]
528 DATA CF,CF,CF,CF,CF,CF,CF,9E,647      [1669]
529 DATA 08,28,45,2C,00,00,6D,68,176      [1327]
530 DATA 40,94,6D,2C,00,00,CF,8E,2CA      [869]
531 DATA 00,00,CF,9E,48,40,45,9E,2D8      [1249]
532 DATA 08,00,45,9E,08,00,45,2C,164      [1259]
533 DATA 40,80,6D,2C,00,00,45,9E,23C      [1097]
534 DATA 80,C0,45,9E,08,00,45,2C,29C      [1374]
535 DATA 14,00,6D,9E,9E,6D,CF,CF,3C8      [1403]
536 DATA CF,CF,CF,CF,8A,CF,CF,CF,633      [2279]
537 DATA CF,45,CF,CF,CF,45,CF,CF,564      [741]
538 DATA 8A,CF,CF,CF,8A,45,CF,CF,564      [1243]
539 DATA 8A,45,CF,CF,04,45,CF,8A,40F      [1715]
540 DATA 19,08,CF,8E,32,22,CF,8E,32F      [782]
541 DATA 23,24,45,8E,21,13,45,8E,221      [1721]
542 DATA 23,13,4D,CF,18,31,4D,CF,2B7      [1526]
543 DATA 11,26,45,CF,8E,0C,CF,CF,383      [1491]
544 DATA CF,CF,CF,CO,CO,CO,CO,CO,62D      [2293]
545 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
546 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
547 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
548 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
549 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
550 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
551 DATA CO,CO,CO,CO,CO,CO,CO,CO,600      [1468]
552 DATA CO,CO,CO,00,00,00,00,00,240      [1159]
553 MEMORY &7FFF                          [150]
554 adr=&8000                             [628]
555 FOR zeile=100 TO 552                  [1680]
556 p=0                                   [257]
557 FOR i=1 TO 8                          [462]
558 READ w$:w=VAL("&"+w$)                 [1166]
559 POKE adr,w:adr=adr+1                  [1068]
560 p=p+w                                 [296]
561 NEXT                                  [350]
562 READ pr$:pr=VAL("&"+pr$)              [823]
563 IF p<>pr THEN PRINT"Fehler in Zeile";z [1818]
eile:f=1
564 NEXT                                  [350]
565 IF f=1 THEN STOP                      [910]
566 SAVE"LAB2.BIN",b,&8000,&E23           [1263]
567 PRINT"Fertig!!!"                      [1187]
```

```
96 '*************************           [894]
97 '  Listing 3 :  LAB .WAN               [882]
98 '*************************           [894]
99 '                                      [117]
100 MEMORY &3FFF                          [758]
101 LOAD"LAB1.BIN",&4000                  [1806]
102 LOAD"LAB2.BIN",&5000                  [1690]
103 SAVE"LAB.",b,&4000,&1E23,&4000        [843]
```

Listing "LABYRINTH"



Manche mögen's digital!

Logische Zustände dargestellt mit IMPULS.

Technisch Interessierte, die sich außer Ihrem Computer noch einem anderen Hobby, nämlich der Elektronik, zugewandt haben, kommen um die Digitaltechnik nicht herum. Diese kann in manchen Fällen sehr knifflig sein. Vor allem die Verknüpfungen des Herrn Boole bringen den Benutzer beim "entknüpfen" manchmal an den Rand der Verzweiflung. Damit dieses für Lern- und Anwendungswillige nicht zur Dauersituation wird, stellen wir Ihnen folgendes helfende Programm vor.

Das Programm namens "IMPULS" errechnet den Ausgangszustand aus einer eingegebenen Funktion und stellt diese grafisch dar.
Da der Zeichensatz des CPC nicht über die genormten Verknüpfungszeichen verfügt, wurden aus Gründen der Vereinfachung folgende Zeichen als Verknüpfungssymbole gewählt:

*** = logisch UND (AND)**
\+ = logisch ODER (OR)
& = logisch UND - NICHT (NAND)
% = logisch ODER - NICHT (NOR)
< = EXOR
> = EXNOR
\ = NEGATION (NOT)

Bei der Eingabe einer Funktion können natürlich auch Klammern (sogar ineinander verschachtelte) verwendet werden.
Beim Start des Programmes wird ein Grundgerüst zur Verfügung gestellt. Bevor das Programm startet, werden Sie nach der Anzahl der darzustellenden Impulse gefragt, diese kann zwischen einem und insgesamt 36 liegen. Sie können jederzeit überschrieben werden, jedoch wird dann das Ergebnis der Funktion aus der geänderten Zeile errechnet, was allerdings nicht unbedingt von Nachteil ist.
Der komfortable Editor ermöglicht es, bei einem auftretenden Syntax- Fehler in der Formel, diese mit dem Copy- Cursor zu übernehmen und auszubessern. Für falsche Formeln kann allerdings keine Garantie übernommen werden.

Grundgedanken zum Programm

Damit denjenigen Lesern, die sich noch nicht ausführlich mit der Thematik der Digitaltechnik vertraut gemacht haben, das Programm nicht unnütz erscheint, hier ein paar Grundgedanken dazu.

Jeder von Ihnen weiß, daß des Computers 'Herz', der Mikroprozessor, alles das, was man von ihm verlangt, nur als Nullen und Einsen versteht. Diese beiden Zahlen charakterisieren eigentlich immer zwei Zustände, die man darstellen kann:

Zustand	0	1
Beispiel	Strom aus	Strom an
oder	Licht aus	Licht an

Nun gibt es eine Reihe von integrierten Schaltungen (Insider sprechen von ICs, dies hat jedoch keinen Bezug zur deutschen Bundesbahn, sondern ist die englische Beschreibung Integrated Circuit), die ebenfalls mit diesen zwei Zuständen arbeiten, nämlich die Digital- ICs. Diese kleinen Wunderdinger 'errechnen' aus den angelegten Eingangszuständen A, B, C, D... usw. den Ausgangszustand Q. Darauf jetzt genauer eingehen zu wollen, hieße, dieses Heft auf den fünffachen Umfang zu bringen, wovon allein dieser Artikel 4/5 verbrauchen würde, und das kann man niemanden zumuten. Es gibt jedoch gute Elektronik-Lehrbücher, die das Thema für jedermann verständlich machen.

Nehmen wir uns stattdessen ein kleines Beispiel vor. Stellen Sie sich vor, Sie haben eine Wohnzimmerlampe und zwei Ausschalter gekauft und möchten nun, daß das Licht nur dann angeht, wenn beide Schalter auf EIN stehen. (Eigentlich unsinnig, wer möchte schon im Dunkeln einen zweiten Schalter suchen, aber als Beispiel gut genug.) Nun müssen wir uns die Funktion suchen. Sie sieht folgendermaßen aus:

Schalter 1 eingeschaltet UND Schalter 2 eingeschaltet = Lampe leuchtet.

Schreiben wir nun alle Möglichkeiten der Funktion untereinander:

SCHALTER 1	SCHALTER 2	LAMPE
AUS	AUS	AUS
AUS	EIN	AUS
EIN	AUS	AUS
EIN	EIN	EIN

Wie man unschwer erkennen kann, leuchtet die Lampe nur dann, wenn Schalter 1 und Schalter 2 auf EIN geschaltet sind, also genau das was wir brauchen. Setzen wir nun statt EIN und AUS die logischen Zustände 1 und 0 ein:

SCHALTER 1	SCHALTER 2	LAMPE
0	0	0
0	1	0
1	0	0
1	1	1

Damit haben wir unsere erste digitale Funktion geschrieben. Was hat das aber jetzt mit dem Programm IMPULS zu tun?
Nun, starten Sie das Programm doch einmal. Zuerst werden Sie nach der Anzahl der Impulse gefragt. Hier geben Sie die Zahl 2 ein. Der Grund: Wir haben zwei Schalter, die den Ausgangszustand ergeben sollen, und wie oben zu sehen, sind das insgesamt 4 Schaltmöglichkeiten, denn 2 Schalter hoch 2 Schaltzustände ergeben 4 Schaltmöglichkeiten.
Diese werden in eine Prioritätsreihe gesetzt, wobei Schalter 2 die niedrige Priorität bekommt, Schalter 1 die höhere, d.h. Schalter 2 wird bei jeder Zustandsmöglichkeit als Erster geschaltet.
Nach dem wir diese Eingabe mit <RETURN> abgeschlossen haben, erscheint die Funktionsgrafik mit den beiden Schaltern A und B. Zu verwirrend, meinen Sie? Überhaupt nicht. Die Grafik von Bild 1 ist in vier gleiche Abschnitte unterteilt (aha, vier Abschnitte = 4 Schaltmöglichkeiten). Ist die Impulslinie unten, haben wir den 0- Zustand, ist Sie oben, den 1- Zustand. Prüfen wir nun unsere Funktionsgleichung nach. Wir drücken die Copy-Taste und befinden uns nun im Eingabe-Modus. In der oberen Zeile stehen noch einmal alle Zeichen, die wir für die Eingabe benutzen können.
Geben wir also jetzt ein:

A*B , also A (Schalter 2) AND (UND) B (Schalter 2).

Nach einem nochmaligen <RETURN> sagt der Rechner uns, daß er rechnet, und nach kurzer Wartezeit erscheint unter C der Ausgangszustand (unsere Lampe). Jetzt nehmen wir uns die einzelnen Abschnitte einmal vor:

Abschnitt 1: A (Schalter 2) ist auf 0, B (Schalter 1) ist auf 0, C ist auf 0 (Lampe leuchtet nicht).

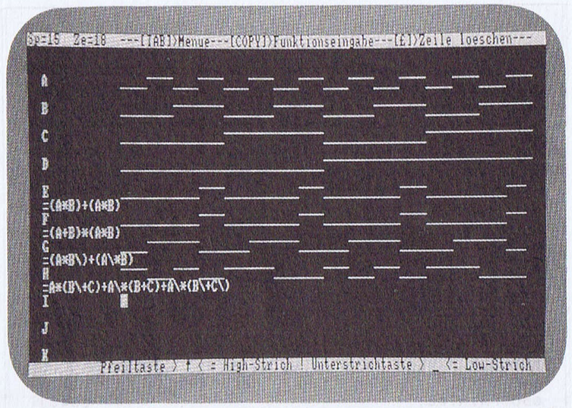


Abb. 1: Boolesche Algebra grafisch dargestellt.

Abschnitt 2: A ist auf 1, B auf 0, C ist wieder auf 0 (immer noch dunkel).

Abschnitt 3: A ist auf 0, B auf 1, C ist immer noch auf 0 (es wird noch nicht heller).

Aber jetzt!
Abschnitt 4: A ist 1, B ist 1, C ist 1, Herz, was willst du mehr, die Lampe leuchtet.

Wenn man dieses etwas arg einfache Beispiel als Grundlage für eigene Formeln nimmt, so dürfte es nach kurzer Einarbeitung selbst für Nichtfachleute recht einfach sein, die ODER-, EXOR-, EXNOR-, NOT- Funktionen darzustellen, da diese ja auch als BASIC-Befehle auf den CPC- Rechnern bestehen. Dazu ein kleiner Tip: Unsere Lampe soll leuchten, wenn Schalter 1 ODER Schalter 2 auf EIN steht. Na, haben Sie schon eine Idee? Immerhin ist diese Funktion schon um einiges besser als die erste, ODER?

(P. Kolb/JB)

für 464-664-6128

```
10 '----------------------------                [1412]
20 '---> impuls.dig <--- 11.87                  [1955]
30 '---> von P.Kolb <---                        [1151]
40 '--->             <---                       [1660]
50 '---> 8670 Hof/S.<---                        [791]
60 '----------------------------                [1412]
70 OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOUT      [2303]
80 MODE 1:INK 1,0:PAPER 0:PEN 3                 [1663]
90 GOSUB 2180                                   [855]
100 GOSUB 2520                                  [867]
110 GOSUB 2580                                  [903]
120 '---> Textteil                              [694]
130 a$="":a$=" ":b$="":i$=SPACE$(20)            [1896]
140 b$=CHR$(242)+CHR$(243)+CHR$(240)+CHR$(      [6382]
241)+CHR$(127)+CHR$(16)+CHR$(13)+CHR$(9)+C
HR$(224):MID$(i$,1,LEN(b$))=b$:b$=""
150 sp=1:GOSUB 170:GOTO 310                     [1349]
160 LOCATE sp,ze:PRINT MID$(c$(q),sp,1);:R      [2525]
ETURN
170 PRINT CHR$(24);:LOCATE sp,ze:PRINT MID      [6699]
$(c$(q),sp,1);:PRINT CHR$(24);:RETURN
180 IF sp=1 THEN c$(q)=MID$(c$(q),sp+1,80-      [3572]
(sp))+" ":GOTO 460
190 c$(q)=MID$(c$(q),1,sp-1)+MID$(c$(q),sp      [2867]
+1,80-(sp))+" ":GOTO 460
200 sp=sp-1:IF sp<1 THEN sp=1                   [2227]
210 c$(q)=MID$(c$(q),1,sp-1)+MID$(c$(q),sp      [2608]
+1,81-(sp+1))+" ":GOTO 460
220 GOSUB 160:'------ C  links -----            [1112]
230 sp=sp-1                                     [513]
```

Listing "IMPULS"

```
240 IF sp<1 THEN sp=1                           [1075]
250 GOSUB 170                                   [939]
260 GOTO 310                                    [510]
270 GOSUB 160:'------ C  rechts ----            [1808]
280 sp=sp+1                                     [549]
290 IF sp>79 THEN sp=80                         [554]
300 GOSUB 170                                   [939]
310 LOCATE#1,1,1:PRINT#1,USING"S> ## Z> ##      [4169]
";sp,q:MID$(a$,1,1)=CHR$(128)
320 MID$(a$,1,1)=INKEY$:IF MID$(a$,1,1)=CH      [3031]
R$(128) THEN 310
330 hq=MAX(hq,q)                                [162]
340 ON INSTR(i$,a$) GOTO 220,270,410,500,2      [3249]
00,180,390,2670,1770
350 '---------------       L- R-  O-  U- 1     [2600]
27- 16- 13-  M-  F-Eing.
360 IF q=zeilen THEN CLS#1:PRINT#1," E N D      [6486]
 E ":FOR s=1 TO 1000:NEXT:GOSUB 2640:GOTO
310
370 IF MID$(a$,1,1)="^"THEN MID$(a$,1,1)=C      [2274]
HR$(255)
380 MID$(c$(q),sp,1)=MID$(a$,1,1):GOTO 270      [1508]
390 GOSUB 160:sp=1:GOTO 500:'(13)               [1865]
400 '--- routine cursor oben                    [2860]
410 IF q>=2 THEN 430 ELSE CLS#1                 [993]
420 CLS#1:PRINT#1," A N F A N G ":PRINT CH      [4399]
R$(7);:FOR s=1 TO 1000:NEXT:GOSUB 2640:GOT
O 310
430 GOSUB 160                                   [933]
440  q=q-1:ze=ze-1                              [390]
450 IF ze=0 THEN PRINT CHR$(11):ze=1            [2018]
460 LOCATE#0,1,ze:PRINT#0,c$(q);                [2036]
470 GOSUB 170                                   [939]
480 GOTO 310                                    [510]
```

Listing "IMPULS"

```
490 '--- routine cursor unten                    [2627]
500 IF q<zeilen THEN 520                         [159]
510 CLS#1:PRINT#1," E N D E     ":PRINT CH       [3782]
R$(7);:FOR s=1 TO 1000:NEXT:GOSUB 2640:GOT
O 310
520 GOSUB 160                                    [933]
530  q=q+1:ze=ze+1                               [965]
540 IF ze=24 THEN PRINT#0,CHR$(10):ze=23         [2038]
550 LOCATE#0,1,ze:PRINT#0,c$(q);                 [2036]
560 GOSUB 170                                    [939]
570 GOTO 310                                     [510]
580 '---> Drucken                                [886]
590 PRINT#2,"Bitte Drucker einschalten !";       [5145]
:PRINT#8,CHR$(27);"7";:WIDTH 82:CLS#2
600 PRINT#8,CHR$(27);"7";                        [1573]
610 FOR s1=1 TO hq                               [643]
620 FOR s2=1 TO 80                               [965]
630 IF MID$(c$(s1),s2,1)=CHR$(255) THEN PR       [5415]
INT#8,"-";ELSE PRINT#8,MID$(c$(s1),s2,1);
640 NEXT                                         [350]
650 PRINT#8,                                     [546]
660 NEXT                                         [350]
670 PRINT#8," * = AND ! & = NAND ! + = OR        [3756]
! % = NOR ! < = EXOR ! > = EXNOR ! \ = NOT
"
680 GOTO 110                                     [423]
690 '---> Teil-Funktions-String auswerten        [1944]
700 s1=1:b$=" "                                  [847]
710 d0=2+(ASC(MID$(tf$,s1,1))-65)*2:'--->z       [3607]
eile A
720 IF d0>=65 THEN d0=(ASC(MID$(tf$,s1,1))       [2587]
-97):d0=d0+65
730 IF d0<0 THEN erfl=1:RETURN                   [1068]
740 MID$(h$(0),fze(0),zeichen-1)=MID$(c$(d       [3474]
0),fze(0),zeichen-1):d3=0
750 s1=s1+1                                      [781]
760 IF MID$(tf$,s1,1)="\"THEN s1=s1+1:d3=0       [2546]
:GOSUB 1280:'--> Neg.
770 IF MID$(tf$,s1,1)="'" THEN 950 :'--->        [2451]
Nur A\
780 '                                            [117]
790 MID$(b$,1,1)=MID$(tf$,s1,1):s1=s1+1:'-       [2766]
--> Verknuepfung
800 IF INSTR("*&+%<>",b$)=0 THEN erfl=1:RE       [1728]
TURN
810 '                                            [117]
820 d1=2+(ASC(MID$(tf$,s1,1))-65)*2:'--->        [2639]
zeile B
830 IF d1>=65 THEN d1=(ASC(MID$(tf$,s1,1))       [4557]
-97):d1=d1+65
840 IF d1<0 THEN erfl=1:RETURN                   [1835]
850 MID$(h$(1),fze(0),zeichen-1)=MID$(c$(d       [3096]
1),fze(0),zeichen-1):d3=1
860 s1=s1+1                                      [781]
870 IF MID$(tf$,s1,1)="\"THEN s1=s1+1:d3=1       [3255]
:GOSUB 1280:'--> Neg.
880 s1=s1-1                                      [729]
890 ON INSTR("*&+%<>",b$) GOSUB 990,990,10       [3630]
60,1060,1130,1130
900 MID$(tf$,s1,1)=CHR$((fze(zf)/2)-2+66)        [1594]
910 '                                            [117]
920 MID$(h$(0),1,80)=SPACE$(80):MID$(h$(1)       [3188]
,1,80)=SPACE$(80)
930 IF MID$(tf$,s1+1,1)<>"'" THEN 710            [1787]
940 RETURN                                       [555]
950 '---> Sonderfall (nur A\)                    [1371]
960 MID$(c$(fze(zf)),fze(0),zeichen-1)=MID       [3684]
$(h$(d3),fze(0),zeichen-1)
970 RETURN                                       [555]
980 '---> Gosubs auswerten                       [1658]
990 '---> Und ( Und nicht )                      [1153]
1000 FOR s2=fze(0) TO fze(0)+zeichen-1           [1747]
1010 IF MID$(h$(1),s2,1)="_" OR MID$(h$(0)       [5242]
,s2,1)="_" THEN MID$(c$(fze(zf)),s2,1)="_"
:GOTO 1030
1020 IF MID$(h$(1),s2,1)=CHR$(255) AND MID       [6328]
$(h$(0),s2,1)=CHR$(255)THEN MID$(c$(fze(zf
)),s2,1)=CHR$(255)
1030 NEXT                                        [350]
1040 IF b$="&" THEN d3=fze(zf):GOSUB 1220        [2323]
1050 RETURN                                      [555]
1060 '---> Oder ( Oder nicht )                   [1113]
1070 FOR s2=fze(0) TO fze(0)+zeichen-1           [1747]
1080 IF MID$(h$(0),s2,1)="_" AND MID$(h$(1       [5512]
),s2,1)="_" THEN MID$(c$(fze(zf)),s2,1)="_
":GOTO 1100
1090 IF MID$(h$(0),s2,1)=CHR$(255) OR MID$       [4596]
(h$(1),s2,1)=CHR$(255)THEN MID$(c$(fze(zf)
),s2,1)=CHR$(255)
1100 NEXT                                        [350]
1110 IF b$="%" THEN d3=fze(zf):GOSUB 1220        [1982]
1120 RETURN                                      [555]
1130 '---> Exor (Exnor)                          [703]
1140 FOR s2=fze(0) TO fze(0)+zeichen-1           [1747]
1150 IF MID$(h$(0),s2,1)="_" AND MID$(h$(1       [5438]
),s2,1)="_" THEN MID$(c$(fze(zf)),s2,1)="_
":GOTO 1190
1160 IF MID$(h$(0),s2,1)=CHR$(255) AND MID       [4589]
$(h$(1),s2,1)=CHR$(255) THEN MID$(c$(fze(z
f)),s2,1)="_":GOTO 1190
1170 IF MID$(h$(0),s2,1)="_" AND MID$(h$(1       [6975]
),s2,1)=CHR$(255) THEN MID$(c$(fze(zf)),s2
,1)=CHR$(255):GOTO 1190
1180 IF MID$(h$(0),s2,1)=CHR$(255) AND MID       [5098]
$(h$(1),s2,1)="_" THEN MID$(c$(fze(zf)),s2
,1)=CHR$(255)
1190 NEXT                                        [350]
1200 IF b$=">" THEN d3=fze(zf):GOSUB 1220        [1280]
1210 RETURN                                      [555]
1220 '---> Neg. ( & > )                          [903]
1230 FOR s2=fze(0) TO fze(0)+zeichen-1           [1747]
1240 IF MID$(c$(d3),s2,1)="_" THEN MID$(c$       [3417]
(d3),s2,1)=CHR$(255):GOTO 1260
1250 IF MID$(c$(d3),s2,1)=CHR$(255) THEN M       [2960]
ID$(c$(d3),s2,1)="_"
1260 NEXT                                        [350]
1270 RETURN                                      [555]
1280 '---> Neg. ( \ )                            [915]
1290 FOR s2=fze(0) TO fze(0)+zeichen-1           [1747]
1300 IF MID$(h$(d3),s2,1)="_" THEN MID$(h$       [3315]
(d3),s2,1)=CHR$(255):GOTO 1320
1310 IF MID$(h$(d3),s2,1)=CHR$(255) THEN M       [2861]
ID$(h$(d3),s2,1)="_"
1320 NEXT                                        [350]
1330 RETURN                                      [555]
1340 '                                           [117]
1350 '---> Leerzeichen aus Funkt.-String e       [2930]
ntfernen
1360 dz=0                                        [53]
1370 FOR s=1 TO R                                [865]
1380 IF MID$(f$(zf),s,1)<>" "THEN dz=dz+1:       [2011]
GOTO 1400
1390 MID$(f$(zf),s,R-s)=MID$(f$(zf),s+1,R-       [2889]
s):s=s-1:dz=dz-1
1400 NEXT                                        [350]
1410 MID$(f$(zf),dz+1,R-dz)=SPACE$(R-dz):R       [1684]
=dz
1420 RETURN                                      [555]
1430 '---> Klammer suchen                        [1189]
1440 kL=0:kR=0                                   [301]
1450 FOR s=1 TO R                                [865]
1460 IF MID$(f$(zf),s,1)=")"THEN kR=s:GOTO       [2510]
1490
1470 NEXT                                        [350]
1480 erfl=1:RETURN                               [1020]
1490 FOR s=kR TO 1 STEP-1                        [1265]
1500 IF MID$(f$(zf),s,1)="("THEN kL=s:GOTO       [2737]
1530
1510 NEXT                                        [350]
1520 erfl=1:RETURN                               [1020]
1530 MID$(f$(zf),kL,1)=" ":MID$(f$(zf),kR,       [2358]
1)=" "
1540 RETURN                                      [555]
1550 '---> Funktions-String auswerten            [3384]
1560 MID$(tf$,1,80)=SPACE$(80)                   [1266]
1570 hz=96:R=LEN(f$(zf))                         [1170]
1580 IF INSTR(f$(zf),"(")=0 THEN 1700            [1835]
1590 GOSUB 1430                                  [853]
1600 IF erfl=1 THEN 2110                         [542]
1610 MID$(tf$,1,kR-kL+1)=MID$(f$(zf),kL+1,       [2589]
kR-(kL+1))
1620 MID$(f$(zf),kL,(kR-kL)+1)=SPACE$((kR-       [2040]
kL)+1)
1630 MID$(tf$,kR-kL,1)="'"                       [1008]
1640 GOSUB 690 :'--->Ergebnis steht in A-Z       [3090]
 bzw. in fze(zf)
1650 IF erfl=1 THEN 2110                         [542]
1660 hz=hz+1:MID$(f$(zf),kL,1)=CHR$(hz)          [2486]
1670 MID$(c$(hz-32),fze(0),zeichen-1)=MID$       [4602]
(c$(fze(zf)),fze(0),zeichen-1)
1680 GOSUB 1350                                  [972]
1690 GOTO 1580                                   [319]
1700 '--->keine Klammer                          [1293]
1710 kL=INSTR(f$(zf),"'")                        [1058]
```

Listing "IMPULS"


```
1720 MID$(tf$,1,kL)=MID$(f$(zf),1,kL)            [2025]
1730 GOSUB 690                                   [970]
1740 IF erfl=1 THEN 2110                         [542]
1750 f$(zf)="":f$(zf)=f$(0)+"'"                  [1730]
1760 GOTO 110                                    [423]
1770 '---> Funktion eingeben                     [2280]
1780 CLS#1:PRINT#1," * = AND ! & = NAND !        [4785]
+ = OR ! % = NOR ! < = EXOR ! > = EXNOR !
\ = NOT";
1790 ze=0                                        [676]
1800 FOR s1=1 TO 52:ze=ze+1                      [990]
1810 IF RIGHT$(c$(s1),80-fze(0)-1)=SPACE$(       [2802]
80-fze(0)-1) AND s1 MOD 2=0 THEN 1860
1820 NEXT                                        [350]
1830 CLS#2,:PRINT#2,CHR$(24);" KEINE EINGA       [3540]
BE MEHR MOEGLICH ";CHR$(24);CHR$(7);
1840 GOTO 110                                    [423]
1850 '                                           [117]
1860 zf=zf+1:fze(zf)=s1:hq=s1+1:kL=0             [2582]
1870 CLS#2:PRINT#2,"EINGABE ---> ";              [1906]
1880 f$(zf)="":f$(0)="":LOCATE#2,15,1:INPU       [2559]
T#2," ",f$(0)
1890 IF f$(0)="" OR LEN(f$(0))>zeichen THE       [3437]
N f$(zf)="":zf=zf-1:GOTO 110
1900 kL=0                                        [180]
1910 FOR s=1 TO LEN(f$(0))                       [1365]
1920 IF MID$(f$(0),s,1)="(" THEN kL=kL+1         [1819]
1930 IF MID$(f$(0),s,1)=")" THEN kL=kL-1         [1481]
1940 NEXT                                        [350]
1950 IF kl<>0 THEN zf=zf-1:kl=0:GOTO 2110        [1005]
1960 '                                           [117]
1970 FOR s=1 TO LEN(f$(0))                       [1365]
1980 IF ASC(MID$(f$(0),s,1))>96 THEN MID$(       [3347]
f$(0),s,1)=UPPER$(MID$(f$(0),s,1))
1990 IF MID$(f$(0),s,1)<>" "THEN f$(zf)=f$       [3177]
(zf)+MID$(f$(0),s,1)
2000 NEXT                                        [350]
2010 '                                           [117]
2020 IF LEN(f$(zf))<2 AND f$(zf)< CHR$((fz       [8180]
e(zf)/2)-2+66)THEN PRINT#2,CHR$(24);SPACE$
(10);CHR$((fze(zf)/2)-2+66);"=";f$(zf);CHR
$(24):FOR s=1 TO 2000:NEXT:GOTO 110
2030 '                                           [117]
2040 f$(zf)=f$(zf)+"'"                           [1228]
2050 MID$(c$(fze(zf)),3,1)=CHR$((fze(zf)/2       [2960]
)-2+66)
2060 MID$(c$(fze(zf)+1),1,80)=SPACE$(80)         [1957]
2070 MID$(c$(fze(zf)+1),3,1+LEN(f$(0)))="=       [2296]
"+f$(0)
2080 '                                           [117]
2090 CLS#2,:PRINT#2,"Bitte warten ";f$(0);       [3741]
2100 GOTO 1550                                   [389]
2110 '---> Fehler in Eingabe                     [1112]
2120 CLS#0,:LOCATE#0,5,5:PRINT#0,CHR$(24);       [4934]
" Falsche Eingabe ";CHR$(24)
2130 LOCATE#0,1,10:PRINT#0,"Letzte Eingabe       [2652]
 ";f$(0):f$(0)="":f$(zf)=""
2140 LOCATE#0,15,12:INPUT#0," ";f$(0)            [1714]
2150 MID$(c$(fze(zf)+1),1,80)=SPACE$(80)         [1957]
2160 erfl=0                                      [211]
2170 GOTO 1890                                   [361]
2180 '---> Initialisierung                       [1310]
2190 name$="IMPULS.DIG":sp=4                     [1295]
2200 FOR s=1 TO LEN(name$):sp=sp+2               [1287]
2210 LOCATE sp,s+3:PRINT MID$(name$,s,1)         [2313]
2220 NEXT                                        [350]
2230 zeilen=64:dz=INT(zeilen/2)+zeilen:tf$       [4397]
=SPACE$(80):b$=" ":erfl=0
2240 DIM c$(dz),f$(26),h$(1),fze(26):'fze(       [4179]
0)=spaltenanfang von Graphik
2250 FOR d1=0 TO dz                              [1602]
2260 c$(d1)=SPACE$(80)                           [1070]
2270 NEXT                                        [350]
2280 h$(0)=SPACE$(80):h$(1)=SPACE$(80)           [1360]
2290 SYMBOL 255,&FF,&0,&0,&0,&0,&0,&0            [1910]
2300 '                                           [117]
2310 PRINT:INPUT"Wieviel Impulse";wahl:zei       [4892]
chen=72:ziffer=65
2320 IF wahl=0 OR wahl >36 THEN PRINT CHR$       [2146]
(7):GOTO 2310
2330 '                                           [117]
2340 d2=INT(zeichen/wahl/2)                      [723]
2350 zeichen=d2*wahl*2:fze(0)=80-zeichen-1       [2343]
2360 a$=SPACE$(80):d0=0:d1=1:hq=0                [1467]
2370 IF d1+d2>zeichen THEN d0=zeichen-d1+1       [5653]
 ELSE d0=d2
2380 MID$(a$,d1,d0)=STRING$(d0,95)               [1184]
```

Listing "IMPULS"

```
2390 d1=d1+d2:IF d1+d2>zeichen THEN d0=zei       [3454]
chen-d1+1:ELSE d0=d2
2400 IF d1>zeichen THEN 2430                     [1824]
2410 MID$(a$,d1,d0)=STRING$(d0,255)              [1413]
2420 IF d1+d2<zeichen THEN d1=d1+d2 :GOTO        [2542]
2370
2430 hq=hq+2:MID$(c$(hq),3,1)=CHR$(ziffer)       [4958]
:ziffer=ziffer+1
2440 MID$(c$(hq),80-zeichen-1,zeichen)=MID       [2184]
$(a$,1,zeichen)
2450 MID$(a$,1,80)=SPACE$(80)                    [1266]
2460 d2=d2*2:IF d2<zeichen THEN d1=1:GOTO        [2495]
2370
2470 FOR s=hq+2 TO zeilen STEP 2                 [2524]
2480 IF ziffer>90 THEN 2510                      [1196]
2490 MID$(c$(s),3,1)=CHR$(ziffer):ziffer=z       [3486]
iffer+1
2500 NEXT                                        [350]
2510 RETURN                                      [555]
2520 '---> Window-Ink                            [341]
2530 MODE 2                                      [513]
2540 WINDOW#0,1,80,2,24:WINDOW#1,1,80,1,1:       [2711]
WINDOW#2,1,80,25,25
2550 INK 0,26:INK 1,0:BORDER 9                   [1624]
2560 PAPER#1,1:PEN#1,0:PAPER#2,1:PEN#2,0:C       [4115]
LS#1,:CLS#2,
2570 RETURN                                      [555]
2580 '---> ausgabe auf screen                    [2340]
2590 IF hq<24 THEN d0=23:q=1:ze=1 ELSE hq=       [3324]
hq+1:q=q+1:d0=hq:q=d0:ze=23
2600 IF hq>zeilen THEN hq=zeilen:q=zeilen:       [2466]
d0=q
2610 FOR s=2 TO d0                               [911]
2620 LOCATE#0,1,s:PRINT#0, c$(s);                [1584]
2630 NEXT                                        [350]
2640 CLS#1,:LOCATE#1,25,1:PRINT#1,"Funktio       [5615]
nseingabe > Copy-Taste   Menue >TAB<";
2650 CLS#2,:PRINT#2,"Pfeiltaste > ^ < = Hi       [8849]
gh-Strich ! Unterstrichtaste > _ <= Low-St
rich";
2660 RETURN                                      [555]
2670 '---> Menue                                 [1013]
2680 CLS#2,:INPUT#2,"D-rucken  L-aden  S-p       [5898]
eichern  N-eu + ENTER";w$
2690 IF w$="" THEN 100 ELSE w$=UPPER$(w$)        [2542]
2700 ON INSTR("DLSN",w$)GOTO 580,2710,2870       [2121]
,3030
2710 '---> Laden                                 [701]
2720 GOSUB 2990                                  [821]
2730 IF name$="" THEN 100                        [387]
2740 OPENIN name$                                [1050]
2750 INPUT#9,hq,zf,fze(0)                        [1665]
2760 FOR s=1 TO hq                               [1014]
2770 a$="":INPUT#9,a$:c$(s)=SPACE$(80)           [1804]
2780 MID$(c$(s),3,LEN(a$))=a$                    [2337]
2790 NEXT                                        [350]
2800 FOR s=1 TO zf:INPUT#9,f$(s):NEXT            [2644]
2810 FOR s=1 TO zf:INPUT#9,fze(s):NEXT           [3616]
2820 CLOSEIN                                     [752]
2830 FOR s=hq+1 TO zeilen                        [1231]
2840 c$(s)=SPACE$(80)                            [615]
2850 NEXT                                        [350]
2860 GOTO 100                                    [417]
2870 '---> Speichern                             [804]
2880 GOSUB 2990                                  [821]
2890 IF name$="" THEN 100                        [387]
2900 OPENOUT name$                               [700]
2910 PRINT#9,hq,zf,fze(0)                        [952]
2920 FOR s=1 TO hq                               [1014]
2930 PRINT#9,MID$(c$(s),1,80)                    [2148]
2940 NEXT                                        [350]
2950 FOR s=1 TO zf:PRINT#9,f$(s):NEXT            [1940]
2960 FOR s=1 TO zf:PRINT#9,fze(s):NEXT           [3024]
2970 CLOSEOUT                                    [902]
2980 GOTO 100                                    [417]
2990 '---> Filename                              [748]
3000 name$="":CLS#2,:PRINT#2,me$;:INPUT#2,       [5609]
" Bitte Name fuer File eingeben --> ";name
$
3010 IF LEN(name$)>12 THEN 3000                  [1487]
3020 RETURN                                      [555]
3030 '---> NEU                                   [382]
3040 RUN 80                                      [303]
3050 '*********************************          [1391]
```

Listing "IMPULS"

Die Welt der Elemente



Abb. 1 CHEMYSTICAL: Das Periodensystem

CHEMYSTICAL: Chemie einfach gehandhabt

Programme, die das Lernen erleichtern, gibt es zwar wie Sand am Meer, ihre Aktualität verlieren Sie trotzdem nie. Diesmal handelt es sich um ein Chemieprogramm, welches Ihnen in übersichtlicher Form Informationen über jedes Element des Periodensystems näherbringt, dazu noch mit einigen interessanten Optionen.

CHEMYSTICAL behandelt nicht nur das Periodensystem der Elemente, sondern gleichzeitig ist eine kleine Datenbank und ein Vokabelprogramm implementiert, das auch für chemische Formeln geeignet ist. Es ist vollständig menügesteuert (Pulldownmenüs) und benötigt nur sechs Tasten: Cursortasten und COPY zum Anwählen und ENTER zum Verlassen, somit ist eine leichte Bedienung gewährleistet. CHEMYSTICAL läuft auf Schneider Rechnern mit Diskettenstation.

Zur Eingabe

Das Programm besteht aus zwei Teilen. Teil 2 muß unter dem Namen CHESTER.BAS abgespeichert werden. Wen die Länge von ca. 19 KB für beide Programme abschreckt, der kann beruhigt werden: alle REM-Zeilen können weggelassen werden. Außerdem kann der Befehl AUTO benutzt werden.

Die Bedienung

Nach dem Start von Teil 1 und dem Schreiben des Copyrights wird der Hauptbildschirm geschrieben, alle Variablen werden eingelesen und der zweite Teil wird nachgeladen. Deshalb folgt an dieser Stelle eine Wartezeit, der Rechner ist also keinesfalls abgestürzt! Danach befindet sich der Benutzer (kurz: Sie) in der Menüzeile am oberen Bildschirmrand. Mit den Cursortasten kann nun ein Menüpunkt angesteuert und mit COPY angewählt werden. Dies gilt auch für alle Untermenüs (verlassen eines Untermenüs mit ENTER).

Die Menüpunkte

PERIOD: Hier finden Sie sich im Periodensystem wieder. (Verlassen mit ENTER). Drücken Sie jetzt auf COPY und die verschiedenen Informationen des betreffenden Stoffes werden aufgezeigt. Zusätzlich erfolgt eine grafische Darstellung der Elektronenschalen:
Die erste Zahl stellt die vorhandene Anzahl dar, während die Zweite die maximale Anzahl der Elektronen dieser Schale darstellt. Durch Tastendruck gelangt man wieder ins Periodensystem. Die grafische Darstellung kann mit TAB ein- bzw. ausgeschaltet werden. Steht z.B. bei Siedepunkt keine Zahl, konnte er nicht bestimmt werden, weil der Stoff nicht stabil genug ist (oder aus irgendeinem anderen Grund).
SUCHEN: Hier können Sie auf drei verschiedene Arten ein Element suchen: über den Namen, das Symbol oder die Ordnungszahl. Einfach die Eingabe tätigen und schon erscheinen die gewünschten Informationen oder ein "Nicht vorhanden".
FORMELN1 eingeben: Hier können chemische. Formeln eingegeben werden, wobei die tiefgestellten Zahlen über die Nummerntastatur erreicht werden. (Verlassen durch Eingabe von ENTER bei Formel).
BLÄTTERN: Mit den Cursortasten können Sie hier Ihre Formelsammlung einsehen. (Verlassen durch ENTER).
ÄNDERN: Ansteuern der zu ändernden Formel wie bei BLÄTTERN, mit COPY anwählen und dann mit den Cursortasten und COPY das Feld bestimmen. (Verlassen durch ENTER).
SUCHEN: Den Feldindex mit den Cursortasten und COPY anwählen und den Suchbegriff eingeben.
LERNEN: Hier können Sie Ihre eingegebenen Formeln lernen. Geben Sie "F" ein, wenn die Formel vorgegeben werden soll, ansonsten "N".
FORMELN2 laden und speichern: Nach dem Direktory aller unter CHEMYSTICAL abgespeicherten Dateien sollten Sie den Dateinamen eingeben. (Verlassen durch Eingabe von ENTER, falls Sie nur aus Versehen hier gelandet sind).
MET+NIMET: +SÄURE+MET: +LAUGBD: drei Möglichkeiten der Berechnung von chem. Gleichungen: man kann hier ein Metall mit einem Nichtmetall oder einer SÄURE reagieren lassen, ohne sich die Wohnung zu demolieren, auch könnenzwei Möglichkeiten der Laugenbildung dargestellt werden. Bei allen drei Punkten wird nach den nötigen Infos gefragt. Metalleund Säuren werden als Symbole bzw. als Formeln eingegeben. Auch hier den Zahlenblock für Indexzahlen benutzen. Die gefragten Wertigkeiten dürfen Drei nicht übersteigen. Alle Eingaben sollten korrekt und sinnvoll sein, sonst erscheinen falsche Ergebnisse.
EXTRAS:
UHR: Eingabe der Uhrzeit.
WECKER: Eingabe der Weckzeit. Geweckt werden Sie durch ein Aufblinken des Randes.
FARBE: Mit den angegebenen Tasten können die einzelnen Farben geändert werden.
ENDE: Spricht für sich, beinhaltet aber eine Sicherheitsabfrage.

(H. Roes/D. Jepsen/jb)



Abb. 2: Informationen und Elektronenschalen eines Elements

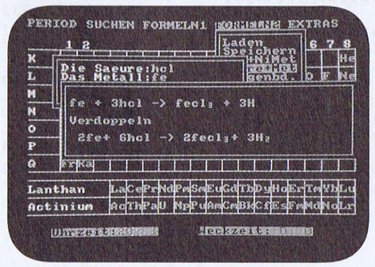


Abb. 3: Formeln lassen sich problemlos berechnen.

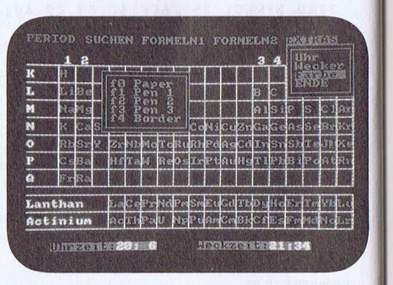


Abb. 4: Sonderfunktionen, über EXTRAS einstellbar

Für 464-664-6128

```
10 '*********************************          [1416]
20 '*   C H E M Y S T I C   V1.0    *          [1308]
30 '*                       by      *          [875]
40 '*Henning Roes & Dominique Jepsen*          [1310]
50 '*                               *          [1206]
60 '*         2300 Kiel 14          *          [917]
70 '*                               *          [1392]
80 '*********************************          [1416]
90 DEFINT a-z:DEFREAL c,d,g,p,s                [1811]
100 MODE 1:INK 1,24:INK 2,17:INK 3,10:INK      [4274]
0,0:BORDER 0:PEN 1:PAPER 0
110 DATA 9,3,3,C H E M Y S T I C   V1.0,12     [6966]
,9,1,Copyright  1987 by,15,12,2,Henning Ro
es,21,14,1,&,13,16,2,Dominique Jepsen,15,2
4,1,Bitte warten
120 FOR n=1 TO 6:READ a,b,c,a$:PEN c:FOR m     [3319]
=1 TO LEN(a$)
130 LOCATE a,b:PRINT MID$(a$,m,1):a=a+1:FO     [6498]
R x=1 TO 150:NEXT:NEXT:FOR x=1 TO 800:NEXT
:NEXT
140 KEY DEF 72,1,240:KEY DEF 73,1,241:KEY      [9281]
DEF 74,1,242:KEY DEF 75,1,243:KEY DEF 76,1
,224:KEY DEF 69,1,97,65,234:KEY DEF 42,1,1
17,85,236:KEY DEF 34,1,111,79,235
150 KEY DEF 5,1:KEY DEF 13,1:KEY DEF 14,1:     [2454]
KEY DEF 15,1:KEY DEF 20,1
160 WINDOW#1,5,40,5,23:WINDOW#2,1,40,25,25      [5111]
:PEN#1,1:PAPER#2,1:PEN#2,0:DEG
170 DIM sy$(103),el$(103),sd(103),sm(103),      [11715]
fm$(50),na$(50),bm$(50),gw(103),x(103),y(1
03),me$(5),ume$(5,5),me(5),po(5),fb(5),ff$
(3),eig$(8),si(45),co(45),m(7),s(7),dc(103
),k(50),z(50),gas$(6),ind$(4)
180 FOR n=0 TO 45 STEP 2:si(n)=SIN(n):co(n     [4049]
)=COS(n):NEXT
190 nu=1:st=20:gra=1:push=42331:pop=42343:      [5257]
fe$="Nicht vorhanden"
200 IF HIMEM>42400 THEN SYMBOL AFTER 212        [2442]
210 MEMORY 25935:FOR s=42331 TO 42395:READ      [3547]
 a:POKE s,a:NEXT
220 OPENOUT"d":MEMORY HIMEM-1:CLOSEOUT          [2613]
230 DATA 33,0,192,17,80,101,1,255,63,237,1      [11642]
76,201,33,80,101,17,0,192,1,255,63,237,176
,201,229,42,153,165,34,128,188,58,155,165,
50,130,188,205,128,188,33,115,165,34,129,1
88,33,128,188,54,195,225,216,200,254,26,55
,63,192,183,55,201,223,139,168
240 IF PEEK(&19)=&BF THEN POKE &BC80,195:P      [4924]
OKE &BC81,115:POKE &BC82,165
250 SYMBOL 230,0,0,14,17,35,44,40,16:SYMBO      [3738]
L 231,0,0,112,248,252,124,60,56
260 SYMBOL 232,16,16,8,15,15,15,7:SYMBOL 2      [4113]
33,36,36,68,200,240,192,128
270 SYMBOL 234,72,0,120,12,124,204,118:SYM      [3081]
BOL 235,36,0,60,102,102,102,60
280 SYMBOL 236,68,0,102,102,102,102,62:SYM      [5575]
BOL 237,14,10,14:SYMBOL 238,90,60,102,102,
126,102,102
290 kern1$=CHR$(230)+CHR$(231):kern2$=CHR$      [2265]
(232)+CHR$(233)
300 FOR n=213 TO 222:READ a,b,c,d,e:SYMBOL      [4560]
 n,0,0,0,a,b,c,d,e:NEXT
310 FOR n=0 TO 9:KEY n,CHR$(213+n):NEXT         [2213]
320 DATA 56,40,40,40,56,16,48,16,16,56,56,      [6729]
8,56,32,56,56,8,56,8,56,40,40,56,8,8,56,32
,56,8,56,56,32,56,40,56,56,8,16,16,16,56,4
0,56,40,56,56,40,56,8,56
330 FOR n=1 TO 103:READ sy$(n),el$(n),x(n)      [4144]
,y(n),gw(n),sm(n),sd(n),dc(n):NEXT
340 FOR n=1 TO 5:READ me$(n),me(n),po(n):F      [4256]
OR m=1 TO me(n):READ ume$(n,m):NEXT:NEXT
350 FOR n=1 TO 7:READ a$:eig$(n)=CHR$(15)+      [3414]
CHR$(2)+a$+CHR$(15)+CHR$(1):NEXT
360 FOR n=1 TO 3:READ a$:ff$(n)=CHR$(15)+C      [3508]
HR$(2)+a$+CHR$(15)+CHR$(1):NEXT
370 FOR n=1 TO 7:READ m(n):NEXT                 [1786]
380 FOR n=1 TO 6:READ gas$(n):NEXT              [2382]
390 el$(42)="Molybd"+CHR$(234)+"n":el$(55)      [12365]
="C"+CHR$(234)+"sium":sy$(53)="J"+CHR$(24)
+"I"+CHR$(24):ume$(3,2)="Bl"+CHR$(234)+"tt
ern":ume$(3,3)=CHR$(238)+"ndern  ":ume$(4,
4)="S"+CHR$(234)+"ure+Met":ind$(2)=CHR$(21
5):ind$(3)=CHR$(216):ind$(4)=CHR$(217)
400 DATA H,Wasserstoff,1,1,1.01,-259.2,-25      [9803]
2.8,0.071,He,Helium,35,1,4,-272.1,-268.9,0
.13,Li,Lithium,1,3,6.9,1347,180.5,0.53,Be,
```

Listing "Chemystical"

```
Beryllium,3,3,9,1285,2477,1.85
410 DATA B,Bor,25,3,10.8,2180,3660,2.46,C,      [9407]
Kohlenstoff,27,3,12,3750,3370,3.514,N,Stic
kstoff,29,3,14,-210,195.8,1.03,O,Sauerstof
f,31,3,16,-218.8,-183,1.27
420 DATA F,Flour,33,3,19,-219.6,-187.5,1.1      [8773]
1,Ne,Neon,35,3,20.2,-248.6,-246.1,1.204,Na
,Natrium,1,5,23,97.8,881.3,0.97,Mg,Magnesi
um,3,5,24.3,650,1105,1.74
430 DATA Al,Aluminium,25,5,27,660.2,2330,2      [8414]
.7,Si,Silicium,27,5,28.1,1410,2477,2.33,P,
Phosphor,29,5,31,44.2,280.5,1.82,S,Schwefe
l,31,5,32.1,119.6,444.6,2.06
440 DATA Cl,Chlor,33,5,35.5,101,-34.1,1.57      [9535]
,Ar,Argon,35,5,39.9,-189.4,-185.9,1.37,K,K
alium,1,7,39.1,63.6,753.8,0.86,Ca,Calcium,
3,7,40.1,845,1483,1.54
450 DATA Sc,Scandium,5,7,45,1539,2832,2.99      [9005]
,Ti,Titan,7,7,47.9,1677,3267,4.51,V,Vanadi
um,9,7,50.9,1919,3400,6.09,Cr,Chrom,11,7,5
2,1903,2640,7.14
460 DATA Mn,Mangan,13,7,54.9,1247,2030,7.4      [8592]
4,Fe,Eisen,15,7,55.8,1539,3070,7.87,Co,Kob
alt,17,7,58.9,1492,3100,8.89,Ni,Nickel,19,
7,58.7,1452,2730,8.91
470 DATA Cu,Kupfer,21,7,63.5,1083,2595,8.9      [7576]
2,Zn,Zink,23,7,65.4,419.4,908.5,7.14,Ga,Ga
llium,25,7,69.7,29.8,2250,5.91,Ge,Germaniu
m,27,7,72.6,947.4,2830,5.32
480 DATA As,Arsen,29,7,74.9,8.17,616,5.72,      [6576]
Se,Selen,31,7,79,220.5,684.8,4.82,Br,Brom,
33,7,79.9,-7.3,58.8,3.14,Kr,Krypton,35,7,8
3.8,-157.2,-153.4,2.155
490 DATA Rb,Rubidium,1,9,85.5,38.9,688,1.5      [8183]
4,Sr,Strontium,3,9,88.9,771,1385,2.63,Y,Yt
trium,5,9,88.9,1552,3337,4.47,Zr,Zirkonium
,7,9,91.2,2128,3578,6.51
500 DATA Nb,Niob,9,9,92.9,2468,4930,8.58,M      [7070]
o,M,11,9,95.9,2620,3410,10.28,Tc,Technetiu
m,13,9,98,2250,4700,11.49,Ru,Ruthenium,15,
9,101.1,2450,4150,12.45
510 DATA Rh,Rhodium,17,9,102.9,1960,3670,1      [8939]
2.41,Pd,Palladium,19,9,106.4,1552,2930,12.
02,Ag,Silber,21,9,107.9,960.8,2212,10.49,C
d,Cadmium,23,9,112.4,320.9,767.3,8.642
520 DATA In,Indium,25,9,114.8,156.2,2070,7      [8421]
.31,Sn,Zinn,27,9,118.9,231.9,2687,7.29,Sb,
Antimon,29,9,121.8,630.5,1635,6.69,Te,Tell
ur,31,9,127.6,449.8,1390,6.25
530 DATA JI,Jod,33,9,126.9,113.6,185.24,4.      [8097]
94,Xe,Xenon,35,9,131.3,-111.8,-108.1,3.52,
Cs,C,1,11,132.9,28.5,705,1.91,Ba,Barium,3,
11,137.3,726,1696,3.62
540 DATA La,Lanthan,7,16,138.9,920,3454,6.      [7298]
16,Ce,Cer,9,16,140.1,798,3257,6.77,Pr,Pras
eodym,11,16,140.9,935,3017,6.48,Nd,Neodym,
13,16,144.2,1016,3127,7.01
550 DATA Pm,Promethium,15,16,145,1168,2730      [6433]
,7.22,Sm,Samarium,17,16,150.4,1072,1900,7.
54,Eu,Europium,19,16,152,826,1439,5.25,Gd,
Gadolinium,21,16,157.3,1312,3000,7.89
560 DATA Tb,Terbium,23,16,158.9,1356,2480,      [9447]
8.25,Dy,Dysprosium,25,16,162.5,1407,2335,8
.56,Ho,Holmium,27,16,164.9,1470,2720,8.78,
Er,Erbium,29,16,167.3,1522,2510,9.05
570 DATA Tm,Thulium,31,16,168.9,1545,1725,      [9278]
9.32,Yb,Ytterbium,33,16,173.1,816,1193,6.9
7,Lu,Lutetium,35,16,175,1675,3315,9.84,Hf,
Hafnium,7,11,178.5,2150,5400,13.31
580 DATA Ta,Tantal,9,11,180.9,2996,5425,16      [7877]
.68,W,Wolfram,11,11,183.9,3410,5700,19.26,
Re,Rhenium,13,11,186.2,3180,5870,21.03,Os,
Osmium,15,11,190.2,5020,3050,22.61
590 DATA Ir,Irdium,17,11,192.2,2454,4530,2      [8605]
2.65,Pt,Platin,19,11,195.1,1769.3,3830,21.
45,Au,Gold,21,11,197,1063,2660,19.32,Hg,Qu
ecksilber,23,11,200.6,-38.8,357,13.53
600 DATA Tl,Thallium,25,11,204.4,302.5,145      [6866]
3,11.85,Pb,Blei,27,11,207.2,327.4,1751,11.
34,Bi,Wismut,29,11,209,271,1580,9.8,Po,Pol
onium,31,11,209,254,962,9.2
610 DATA At,Astat,33,11,210,300,370,,Rn,Ra      [7699]
don,35,11,222,-71.1,-62.1,4.4,Fr,Francium,
1,13,223,30,680,,Ra,Radium,3,13,226,700,11
40,5.5
620 DATA Ac,Actinium,7,18,227,1050,3300,10      [8894]
.07,Th,Thorium,9,18,232,1755,4800,11.72,Pa
,Protactinium,11,18,231,1568,4200,15.37,U,
Uran,13,18,238,1132,3900,18.97
630 DATA Np,Neptunium,15,18,237,639,3900,2      [6502]
0.48,Pu,Plutonium,17,18,244,639.5,3200,19.
```

Listing "Chemystical"

```
74,Am,Americium,19,18,243,1173,2600,13.67
640 DATA Cm,Curium,21,18,247,1350,,13.51,B [7057]
k,Berkelium,23,18,247,986,,13.25,Cf,Califo
rnium,25,18,251,900,,,Es,Einsteinium,27,18
,252,,,
650 DATA Fm,Fermium,29,18,257,,,,Md,Mendel [6400]
evium,31,18,258,,,,No,Nobelium,33,18,259,,
,,Lr,Lawrencium,35,18,260,,,
660 DATA PERIOD,,1,SUCHEN,3,8,"Name    "," [10519]
Symbol  ",Ord.zahl,FORMELN1,5,15,Eingeben,
,,"Suchen  ","Lernen  ",FORMELN2,5,24,"Lad
en    ",Speichern,Met+NiMet,,Laugenbd.,EXT
RAS,4,33,"Uhr   ",Wecker,"Farbe ","ENDE  "
670 DATA "Name:","Symbol:","A masse:","Ord [10970]
nungszahl:","Siedepunkt:","Schmelzpunkt:",
"Dichte:","Formel :","Name   :","Bemerk.:"
,2,8,18,32,50,72,98
680 DATA N,O,F,Cl,H,Br                    [890]
690 '** Periodensystem                    [1090]
700 CLS                                   [91]
710 FOR n=1 TO 103:LOCATE#1,x(n),y(n):PRIN [3683]
T#1,sy$(n);:NEXT
720 PEN 2:LOCATE 1,20:PRINT"Lanthan":LOCAT [3112]
E 1,22:PRINT"Actinium"
730 FOR n=75 TO 81:LOCATE 1,2*n-145:PRINT  [1521]
CHR$(n):NEXT
740 FOR n=121 TO 365 STEP 32:MOVE 0,n:DRAW [2551]
R 640,0,2:NEXT
750 FOR n=63 TO 639 STEP 32:MOVE n,121:DRA [1967]
WR 0,224:NEXT
760 FOR n=159 TO 639 STEP 32:MOVE n,41:DRA [2228]
WR 0,64:NEXT
770 FOR n=41 TO 105 STEP 32:MOVE 0,n:DRAWR [1971]
 640,0:NEXT
780 TAG:PLOT 72,360,2:PRINT"1 2"SPC(21)"3  [3978]
4 5 6 7 8";:TAGOFF
790 PEN 1:LOCATE 1,1:FOR n=1 TO 5:PRINT me [3128]
$(n)" ";:NEXT
800 LOCATE#2,4,1:PRINT#2,"Uhrzeit:":LOCATE [3381]
#2,22,1:PRINT#2,"Weckzeit:":PEN#2,3
810 CALL push                             [263]
820 CHAIN MERGE"chester"                  [1185]
```

```
10 CLOSEIN:EVERY 3000,0 GOSUB 2520:GOSUB 2 [2639]
550
20 '** Menue                              [476]
30 GOSUB 2570:GOSUB 2560                  [1148]
40 IF w=224 THEN ON nu GOSUB 260,100,100,1 [4948]
00,100:CALL pop:GOSUB 2550:GOTO 30
50 LOCATE po(nu),1:PRINT me$(nu)          [1642]
60 IF w=242 THEN nu=nu-1:IF nu=0 THEN nu=5 [2022]
70 IF w=243 THEN nu=nu+1:IF nu=6 THEN nu=1 [2164]
80 GOTO 30                                [340]
90 '** Untermenue                         [636]
100 y=1                                   [354]
110 a=po(nu):b=LEN(ume$(nu,1)):c=2:d=me(nu [4784]
):GOSUB 2220
120 FOR n=1 TO me(nu):PRINT ume$(nu,n);:NE [2590]
XT
130 LOCATE 1,y:PRINT CHR$(24)ume$(nu,y)CHR [4338]
$(24);
140 GOSUB 2560                            [827]
150 IF w=9 THEN gra=(gra+1)MOD 2          [2181]
160 IF w=224 THEN 220                     [1417]
170 IF w=13 THEN WINDOW 1,40,1,25:RETURN  [1845]
180 LOCATE 1,y:PRINT ume$(nu,y);          [2944]
190 IF w=240 THEN y=y-1:IF y=0 THEN y=me(n [2640]
u)
200 IF w=241 THEN y=y+1:IF y=me(nu)+1 THEN [2679]
 y=1
210 GOTO 130                              [371]
220 ON 5*nu-10+y GOSUB 460,460,410,230,230 [4933]
,530,1110,1250,830,630,950,950,1640,1860,2
040,1520,1520,1440,1590,230
230 CALL pop:GOSUB 2570:GOSUB 2550        [1410]
240 GOTO 110                              [423]
250 '** Periodensystem                    [1090]
260 n=1                                   [236]
270 LOCATE#1,x(n),y(n):PEN#1,3:PRINT#1,sy$ [4277]
(n):aa=n:MOVE x(n)*16+78,352-16*y(n):DRAWR
 0,-16,2
280 GOSUB 2560                            [827]
290 IF w=9 THEN gra=(gra+1)MOD 2          [2181]
```

Listing "Chemystical"

```
300 IF w=224 THEN GOSUB 2250:CALL pop:GOSU [4273]
B 2550:GOTO 270
310 IF w=13 THEN RETURN                   [795]
320 IF w=242 THEN n=n+(n>1)               [1843]
330 IF w=243 THEN n=n-(n<103)             [1153]
340 IF w=240 THEN r=1:t=-1:GOSUB 380      [1647]
350 IF w=241 THEN r=103:t=1:GOSUB 380     [3012]
360 LOCATE#1,x(aa),y(aa):PEN#1,1:PRINT#1,s [3960]
y$(aa):MOVE x(aa)*16+78,352-16*y(aa):DRAWR
 0,-16
370 GOTO 270                              [415]
380 FOR m=n+t TO r STEP t:IF x(m)=x(n)THEN [2995]
 n=m:m=r
390 NEXT:RETURN                           [940]
400 '** Ordnungszahl                      [877]
410 a=6:b=20:c=5:d=3:GOSUB 2220           [1993]
420 PRINT eig$(4):INPUT"",n               [639]
430 IF n<1 OR n>103 THEN PRINT fe$:GOSUB 2 [3388]
560:RETURN
440 GOSUB 2250:RETURN                     [918]
450 '** Symbol+Name                       [1268]
460 a=6:b=20:c=5:d=3:e=0:GOSUB 2220:PRINT  [3950]
eig$(y):INPUT"",a$
470 FOR n=1 TO 103                        [926]
480 IF UPPER$(a$)=UPPER$(el$(n))OR UPPER$( [4227]
a$)=UPPER$(sy$(n))THEN e=n:n=103:GOTO 490
490 NEXT                                  [350]
500 IF e>0 THEN n=e:GOSUB 2250:RETURN     [1769]
510 PRINT fe$:GOSUB 2560:RETURN           [1107]
520 '** F.eingeben                        [839]
530 a=8:b=27:c=6:d=9:GOSUB 2220:GOTO 590  [2352]
540 CLS:PEN 3:LOCATE 1,1:PRINT"Formelzahl: [3147]
";:PEN 1:PRINT az
550 LOCATE 1,3:PRINT ff$(1);:INPUT"",fm$(a [1467]
z)
560 IF fm$(az)=""THEN az=az-1:RETURN      [2339]
570 LOCATE 1,5:PRINT ff$(2);:INPUT"",na$(a [1737]
z)
580 LOCATE 1,7:PRINT ff$(3);:INPUT"",bm$(a [1945]
z)
590 az=az+1                               [697]
600 IF az=51 THEN GOSUB 2560:RETURN       [2853]
610 GOTO 540                              [425]
620 '** F.lernen                          [565]
630 a=8:b=27:c=8:d=9:GOSUB 2220           [1487]
640 f=0:r=0:FOR n=1 TO az:k(n)=0:NEXT     [2885]
650 INPUT"(F)ormel (N)ame";a$:CLS         [1823]
660 IF UPPER$(a$)="F"THEN fo=1 ELSE fo=0  [1811]
670 FOR n=1 TO az:CLS                     [601]
680 PEN 2:PRINT"Richtig:";:PEN 1:PRINT r:L [6088]
OCATE 14,1:PEN 2:PRINT"Falsch:";:PEN 1:PRI
NT f
690 m=INT(RND*az)+1:IF k(m)=1 THEN 690    [1883]
700 LOCATE 1,3:IF fo THEN 720             [815]
710 PRINT ff$(2)+na$(m):LOCATE 1,5:PRINT f [2459]
f$(1);:GOTO 730
720 PRINT ff$(1)+fm$(m):LOCATE 1,5:PRINT f [2798]
f$(2);
730 INPUT"",a$:a$=UPPER$(a$):IF a$="E"THEN [3005]
 k=n-1:n=az:GOTO 800
740 IF fo THEN 770                        [117]
750 IF a$<>UPPER$(fm$(m))THEN f=f+1:z(f)=m [3546]
:PRINT"Falsch: "fm$(m)ELSE r=r+1
760 GOTO 780                              [409]
770 IF a$<>UPPER$(na$(m))THEN f=f+1:z(f)=m [4364]
:PRINT"Falsch: "na$(m)ELSE r=r+1
780 PEN 1:LOCATE 9,1:PRINT r:LOCATE 21,1:P [3088]
RINT f
790 GOSUB 2560                            [827]
800 NEXT                                  [350]
810 RETURN                                [555]
820 '** F.suchen                          [827]
830 a=6:b=8:c=6:d=8:h=0:GOSUB 2220        [1875]
840 LOCATE 1,3:PRINT ff$(1):LOCATE 1,5:PRI [5159]
NT ff$(2):LOCATE 1,7:PRINT ff$(3)
850 fla=1:GOSUB 1280:a$=UPPER$(a$)        [1336]
860 FOR n=1 TO az                         [636]
870 IF m=1 THEN IF a$=UPPER$(fm$(n))THEN 9 [2657]
10
880 IF m=2 THEN IF a$=UPPER$(na$(n))THEN 9 [2292]
10
890 IF m=3 THEN IF a$=UPPER$(be$(n))THEN 9 [2275]
10
900 GOTO 920                              [332]
910 GOSUB 1120:h=1:n=az:fla=0             [1482]
920 NEXT                                  [350]
930 IF h=0 THEN PRINT f$:GOSUB 2560:RETURN [4629]
 ELSE h=0:GOSUB 2560:RETURN
940 '** Daten speichern+laden             [1399]
950 a=4:b=20:c=5:d=9:GOSUB 2220           [1693]
960 a$="*.che":¦DIR,@a$                   [1143]
```

Listing "Chemystical"


```
970 a=20:b=13:c=14:d=3:GOSUB 2220          [1785]
980 PEN 2:PRINT"Dateiname:":PEN 1          [1854]
990 LOCATE 1,3:INPUT"",a$                  [617]
1000 IF a$=""THEN RETURN                   [1147]
1010 IF INSTR(a$,".")>0 OR LEN(a$)>8 THEN  [1772]
990
1020 a$=a$+".che"                          [385]
1030 IF y=1 THEN 1070                      [1078]
1040 OPENOUT a$                            [765]
1050 PRINT#9,az:FOR n=1 TO az:WRITE#9,fm$( [3954]
n),na$(n),bm$(n):NEXT
1060 CLOSEOUT:RETURN                       [1447]
1070 OPENIN a$                             [715]
1080 INPUT#9,az:FOR n=1 TO az:INPUT#9,fm$( [3322]
n),na$(n),bm$(n):NEXT
1090 CLOSEIN:RETURN                        [651]
1100 '** Blaettern                         [1004]
1110 n=1:IF az=0 THEN RETURN               [615]
1120 a=8:b=27:c=6:d=9:GOSUB 2220           [1738]
1130 LOCATE 1,1:PEN 3:PRINT"Formelnr.:";:P [2625]
EN 1:PRINT n
1140 PRINT:PRINT ff$(1)+fm$(n)             [1406]
1150 LOCATE 1,5:PRINT ff$(2)+na$(n)        [1729]
1160 LOCATE 1,7:PRINT ff$(3)+bm$(n)        [1690]
1170 IF fla=1 THEN RETURN                  [883]
1180 GOSUB 2560                            [827]
1190 IF w=240 THEN n=n+1:IF n=az+1 THEN n= [1808]
1
1200 IF w=241 THEN n=n-1:IF n=0 THEN n=az  [2587]
1210 IF w=224 AND flag=1 THEN fl=1:RETURN  [2247]
1220 IF w=13 THEN RETURN                   [795]
1230 CLS:GOTO 1130                         [1025]
1240 '** F.aendern                         [898]
1250 n=1                                   [236]
1260 flag=1:fl=0:GOSUB 1120                [1658]
1270 IF fl=0 THEN flag=0:RETURN            [1485]
1280 m=1                                   [184]
1290 LOCATE 1,2*m+1:PRINT CHR$(24)MID$(ff$ [2937]
(m),3,8)CHR$(24)
1300 GOSUB 2560                            [827]
1310 IF w=224 THEN 1360                    [1498]
1320 LOCATE 1,2*m+1:PRINT ff$(m)           [1909]
1330 IF w=240 THEN m=m-1:IF m=0 THEN m=3   [1665]
1340 IF w=241 THEN m=m+1:IF m=4 THEN m=1   [2512]
1350 GOTO 1290                             [353]
1360 a=18:b=20:c=14:d=3:GOSUB 2220         [2053]
1370 IF fla=1 THEN PEN 2:PRINT"Suchbegriff [4658]
:":PEN 1:LOCATE 1,2:INPUT"",a$:RETURN
1380 PEN 2:PRINT"Neuer Inhalt:":PEN 1      [1872]
1390 LOCATE 1,2:INPUT"",a$                 [802]
1400 IF m=1 THEN fm$(n)=a$                 [1654]
1410 IF m=2 THEN na$(n)=a$ ELSE bm$(n)=a$  [1979]
1420 GOTO 1260                             [351]
1430 '** Farben                            [559]
1440 a=10:b=9:c=5:d=5:GOSUB 2220           [2381]
1450 PRINT"f0 Paper f1 Pen 1 f2 Pen 2 f3 P [4284]
en 3 f4 Border";
1470 i$=INKEY$:IF i$<>""THEN i=ASC(i$)ELSE [2438]
 1470
1480 IF i=13 THEN RETURN                   [1326]
1490 FOR n=1 TO 5:IF i=212+n THEN fb(n)=fb [5271]
(n)+1:IF fb(n)>26 THEN fb(n)=0:INK n-1,fb(
n)ELSE INK n-1,fb(n)
1500 NEXT:BORDER fb(5):GOTO 1470           [1147]
1510 '** Uhr/Wecker                        [813]
1520 a=10:b=17:c=5:d=3:GOSUB 2220          [2205]
1530 PRINT"Stunden,Minuten:"               [1824]
1540 LOCATE 1,3:INPUT"(St,Mi)",e,f         [1747]
1550 IF e>24 OR e<0 OR f>60 OR f<0 THEN 15 [3252]
40
1560 IF y=2 THEN wmi=f:ws=e ELSE mi=f:st=e [2357]
1570 RETURN                                [555]
1580 '** ENDE                              [542]
1590 a=7:b=16:c=7:d=3:GOSUB 2220           [2184]
1600 PRINT:PRINT"Beenden (J/N) ?"          [1708]
1610 GOSUB 2560:IF UPPER$(a$)<>"J"THEN RET [3794]
URN
1620 INK 1,24:INK 0,1:BORDER 1:PEN 1:PAPER [2718]
 0:MODE 2:END
1630 '** Met-NiMet                         [469]
1640 a=4:b=22:c=5:d=4:GOSUB 2220           [2349]
1650 r=4:GOSUB 2580                        [754]
1660 PEN 2:PRINT"Das Nichtmetall:";:PEN 1: [5557]
INPUT"",n$:IF n$=""THEN 1660
1670 PEN 2:PRINT"Wertigkeit:";:PEN 1:INPUT [4487]
"",wn:IF wn<1 OR wn>4 THEN 1670
1680 a=6:b=30:c=8:d=7:GOSUB 2220           [2027]
1690 GOSUB 2170:gm=0:gn=0:FOR i=1 TO 6:IF  [3135]
m$=gas$(i)THEN gm=1
1700 NEXT:FOR i=1 TO 6:IF n$=gas$(i)THEN g [2259]
n=1
```

Listing "Chemystical"

```
1710 NEXT                                  [350]
1720 IF gm=1 AND am=2 THEN mf$=m$+ind$(2): [5554]
am$=""ELSE IF gm=1 AND am=4 THEN mf$=m$+in
d$(2):am$="2"ELSE mf$=m$:um=1:IF am=1 THEN
 am$=""ELSE am$=STR$(am)
1730 IF gn=1 AND an=2 THEN nf$=n$+ind$(2): [7677]
an$=""ELSE IF gn=1 AND an=4 THEN nf$=n$+in
d$(2):an$="2"ELSE nf$=n$:un=1:IF an=1 THEN
 an$=""ELSE an$=STR$(an)
1740 IF um=1 AND gm=1 THEN um=1 ELSE um=0  [1631]
1750 IF un=1 AND gn=1 THEN un=1 ELSE un=0  [1297]
1760 CLS:PRINT                             [356]
1770 PRINT am$mf$" + "an$nf$" -> "m$ind$(a [2881]
m)n$ind$(an)
1780 IF um=1 OR un=1 THEN PRINT:PRINT"Verd [4077]
oppeln":PRINT ELSE 1840
1790 IF gm=1 AND am=1 THEN am$=""ELSE IF g [3356]
m=1 THEN am$=STR$(am)
1800 IF gm=1 THEN mf$=m$+ind$(2)ELSE mf$=S [1287]
TR$(am*2)+m$:am$=""
1810 IF gn=1 AND an=1 THEN an$=""ELSE IF g [3392]
n=1 THEN an$=STR$(an)
1820 IF gn=1 THEN nf$=n$+ind$(2)ELSE nf$=S [2326]
TR$(an*2)+n$:an$=""
1830 PRINT am$mf$" + "an$nf$" -> 2"m$ind$( [2686]
am)n$ind$(an)
1840 GOSUB 2560:RETURN                     [1840]
1850 '** Saeure+Met                        [1238]
1860 a=4:b=22:c=5:d=4:GOSUB 2220           [2349]
1870 PEN 2:PRINT"Die Saeure:";:PEN 1:INPUT [5449]
"",sa$:IF LEN(sa$)<2 THEN 1860
1880 r=3:GOSUB 2580                        [1476]
1890 a=5:b=34:c=8:d=7:GOSUB 2220           [2008]
1900 wn=1:FOR i=2 TO 3:IF MID$(sa$,2,1)=in [2202]
d$(i)THEN wn=i
1910 NEXT:IF ASC(MID$(sa$,2,1))>122 AND wn [3751]
=1 THEN CLS:GOTO 1860
1920 IF wn=1 THEN n$=MID$(sa$,2)ELSE n$=MI [2016]
D$(sa$,3)
1930 GOSUB 2170                            [873]
1940 IF LEN(n$)>1 AND an>1 AND MID$(n$,2,1 [1980]
)<"a"THEN kn$="("+n$+")"ELSE kn$=n$
1950 IF am>1 THEN am$=STR$(am)ELSE am$=""  [848]
1960 IF an>1 THEN an$=STR$(an)ELSE an$=""  [1434]
1970 ah=an*wn:IF ah MOD 2=0 THEN h$=STR$(a [5748]
h/2)+"H"+ind$(2):h=0 ELSE h$=STR$(an*wn)+"
H":h=1
1980 IF LEFT$(h$,2)=" 1"THEN h$=MID$(h$,3) [1260]
1990 f$=m$+ind$(am)+kn$+ind$(an)           [2169]
2000 CLS:PRINT:PRINT am$m$" +"an$sa$" -> " [1435]
f$" +"h$
2010 PRINT:IF ah=1 OR ah=3 THEN PRINT"Verd [6640]
oppeln":PRINT:PRINT STR$(2*am)m$"+"STR$(2*
an)sa$" -> 2"f$"+"h$+ind$(2)
2020 GOSUB 2560:RETURN                     [1840]
2030 '** Laugenbd.                         [1467]
2040 a=4:b=22:c=5:d=4:GOSUB 2220           [2349]
2050 r=3:GOSUB 2580                        [1476]
2060 a=6:b=30:c=8:d=11:GOSUB 2220          [2496]
2070 IF m$="NH"+ind$(3)THEN PRINT:PRINT m$ [4617]
" + H"ind$(2)"O -> NH"ind$(4)"OH":GOTO 215
0
2080 CLS:PRINT"Metall + H"ind$(2)"O":PRINT [9680]
:IF wm=1 THEN PRINT m$" + H"ind$(2)"O -> "
m$"(OH) + H":PRINT:PRINT"Verdoppeln":PRINT
:PRINT"2"m$" + 2H"ind$(2)"O -> 2"m$"(OH) +
 H"ind$(2)
2090 IF wm=2 THEN PRINT m$" + 2H"ind$(2)"O [3742]
 -> "m$"(OH)"ind$(2)" + H"ind$(2)
2100 IF wm=3 THEN PRINT m$" + 3H"ind$(2)"O [6955]
 -> "m$"(OH)"ind$(3)" + 3H":PRINT:PRINT"Ve
rdoppeln":PRINT:PRINT"2"m$" + 6H"ind$(2)"O
 -> 2"m$"(OH)"ind$(3)" + 3H"ind$(2)
2110 PRINT:PRINT"Metalloxid + H"ind$(2)"O" [4109]
:PRINT
2120 IF wm=1 THEN PRINT m$ind$(2)"O + H"in [3876]
d$(2)"O -> 2"m$"OH"
2130 IF wm=2 THEN PRINT m$"O + H"ind$(2)"O [3798]
 ->"m$"(OH)"ind$(2)
2140 IF wm=3 THEN PRINT m$ind$(2)"O"ind$(3 [2868]
)" + 3H"ind$(2)"O -> 2"m$"(OH)"ind$(3)
2150 GOSUB 2560:RETURN                     [1840]
2160 '** KGV                               [461]
2170 mk=wm*wn                              [423]
2180 kl=MIN(wm,wn):gr=MAX(wm,wn)           [1141]
2190 IF gr MOD kl=0 THEN kgv=mk/kl ELSE kg [3607]
v=mk
2200 am=kgv/wm:an=kgv/wn:RETURN            [2122]
2210 '** Routinen                          [1128]
2220 WINDOW a,a+b+1,c,c+d+1:CLS:MOVE a*16- [8011]
8,407-c*16:DRAWR b*16+16,0,1:DRAWR 0,-d*16
```

Listing "Chemystical"

```
-16:DRAWR-b*16-16,0:DRAWR 0,d*16+16:IF nu=
5 THEN uu=2 ELSE uu=0
2230 MOVE a*16-16,415-c*16:DRAWR 0,-d*16-3 [4963]
2:DRAWR b*16+32-uu,0:DRAWR 0,d*16+32:DRAWR
-b*16-32+uu,0
2240 WINDOW a+1,a+b,c+1,c+d:RETURN          [2333]
2250 a=10:b=22:c=3:d=7                      [1763]
2260 GOSUB 2220                             [879]
2270 PRINT eig$(1)el$(n)                    [578]
2280 PRINT eig$(2)sy$(n)                    [1031]
2290 PRINT eig$(3)gw(n)"u"                  [1436]
2300 PRINT eig$(4)n                         [755]
2310 PRINT eig$(5);:IF sd(n)=0 THEN PRINT"  [4423]
 - ";ELSE PRINT sd(n);
2320 PRINT CHR$(8)CHR$(237)"C":PRINT eig$(  [4830]
6);:IF sd(n)=0 THEN PRINT" - ";ELSE PRINT
sm(n);
2330 PRINT CHR$(8)CHR$(237)"C"              [2037]
2340 PRINT eig$(7);:IF dc(n)=0 THEN PRINT"  [3166]
 - ";ELSE PRINT dc(n)" g/ccm"
2350 IF gra=0 THEN 2510                     [778]
2360 a=2:b=19:c=c+d+1:d=9:GOSUB 2220        [2431]
2370 ORIGIN a*16,(25-c-d)*16-8              [839]
2380 IF n<3 THEN s(1)=n:sc=1:GOTO 2450 ELS  [2150]
E s(1)=2
2390 IF n<11 THEN s(2)=n-2:sc=2:GOTO 2450   [3328]
ELSE s(2)=8
2400 IF n<19 THEN s(3)=n-10:sc=3:GOTO 2450  [3135]
 ELSE s(3)=8
2410 IF n<37 THEN s(3)=s(3)+VAL("&"+MID$("  [9148]
0012355678aaaaaaaa",n-18,1)):s(4)=VAL("&"+
MID$("1222212222212345678",n-18,1)):sc=4:GO
TO 2450 ELSE s(3)=18:s(4)=8
2420 IF n<55 THEN s(4)=s(4)+VAL("&"+MID$("  [8914]
001245578aaaaaaaaa",n-36,1)):s(5)=VAL("&"+
MID$("1222112111012345678",n-36,1)):sc=5:GO
TO 2450 ELSE s(4)=18:s(5)=8
2430 IF n<87 THEN s(4)=s(4)+VAL("&"+MID$("  [10296]
00023456779abcdeeeeeeeeeeeeeeee",n-54,1))
:s(5)=s(5)+VAL("&"+MID$("00100000010000001
2345679aaaaaaaa",n-54,1)):s(6)=VAL("&"+MID
```

Listing "Chemystical"

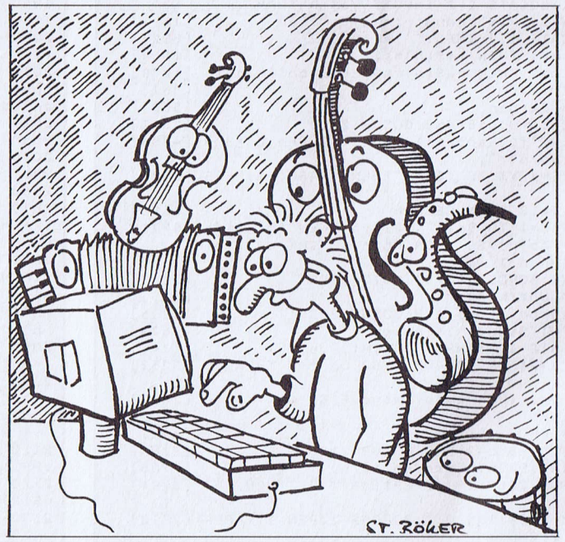
```
$("12222222222222222222222112345678",n-54,
1)):sc=6:GOTO 2450
2440 s(4)=32:s(5)=18:s(6)=8:s(5)=s(5)+VAL(  [10014]
"&"+MID$("0000023457789abcde",n-86,1)):s(6)
=s(6)+VAL("&"+MID$("00121111011111111",n-8
6,1)):s(7)=VAL("&"+MID$("12222222222222222
",n-86,1)):sc=7
2450 LOCATE 9,1:PEN 3:PRINT"Schale Max.":P  [6572]
EN 1:TAG:PLOT 62,94,3:PRINT kern1$;:PLOT 6
2,78:PRINT kern2$;:TAGOFF
2460 FOR z1=sc TO 1 STEP-1:z2=10+z1*9       [2001]
2470 FOR z3=0 TO 45 STEP 2:z4=z2*si(z3):z5  [2834]
=z2*co(z3)
2480 PLOT 78+z4,78+z5,2:PLOT 78+z4,78-z5:P  [8843]
LOT 78-z4,78+z5:PLOT 78-z4,78-z5:PLOT 78+z
5,78+z4:PLOT 78+z5,78-z4:PLOT 78-z5,78+z4:
PLOT 78-z5,78-z4:NEXT
2490 TAG:MOVE 78,88+z1*9:DRAW 178,3+z1*18,  [7464]
1:PLOT XPOS,YPOS,2:PRINT CHR$(74+z1)":";:P
LOT XPOS,YPOS,1:PRINT USING"##";s(z1);:PLO
T XPOS,YPOS,2:PRINT USING"###";m(z1);:TAGO
FF
2500 NEXT:ORIGIN 0,0                        [711]
2510 PEN 1:GOSUB 2560:RETURN                [1279]
2520 mi=mi+1:IF mi=60 THEN st=st+1:mi=0     [2546]
2530 IF st=24 THEN st=0                     [968]
2540 IF wmi=mi AND ws=st THEN BORDER 0,24:  [2457]
FOR s=1 TO 6000:NEXT:BORDER fb(5)
2550 LOCATE#2,12,1:PRINT#2,USING"##:##";st  [7124]
,mi:LOCATE#2,31,1:PRINT#2,USING"##:##";ws,
wmi:RETURN
2560 a$=INKEY$:IF a$=""THEN 2560 ELSE w=AS  [2221]
C(a$):RETURN
2570 WINDOW 1,40,1,25:LOCATE po(nu),1:PRIN  [3849]
T CHR$(24)me$(nu)CHR$(24):RETURN
2580 PEN 2:PRINT"Das Metall:";:PEN 1:INPUT  [4644]
"",m$:IF m$=""THEN 2580
2590 IF m$="NH"+ind$(3) THEN RETURN ELSE P  [5590]
EN 2:PRINT"Wertigkeit:";:PEN 1:INPUT"",wm:
IF wm<1 OR wm>r THEN 2590
2600 RETURN                                 [555]
```

Listing "Chemystical"

MUSICUS

Joystick-Kompositionen mit dem CPC

Wer behauptet, der CPC sei alles andere bloß keine Musikmaschine, der sollte sich mit MUSICUS eines Besseren belehren lassen. Mit diesem Programm können auf ganz einfache Weise Musikstücke komponiert und abgespielt werden. Das Programm läuft auf jedem CPC-Computer.



Nach dem Starten des Programmes erscheint eine Bildschirmmaske, mit der das ganze Programm gesteuert wird. Direkt unter dem oberen Bildschirmrand befindet sich die Kopfleiste mit den Funktionen DIR, LOAD, SAVE, SOUND und ERASE. Darunter sind die Notenlinien für die drei Tonkanäle A,B und C. Hier werden die Noten, Pausenzeichen, usw. grafisch dargestellt und auch eingegeben. Jeder Kanal kann nun eine Stimme spielen, so sind max. drei Stimmen gleichzeitig möglich. Bei mehrstimmigen Musikstücken ist darauf zu achten, daß die Kanäle genau synchron laufen. Pro Kanal können 1198 Noten gespeichert werden. Unter der Notenlinie ist die Befehlsleiste. Mit ihrer Hilfe können Noten in die einzelnen Kanäle eingegeben werden. Nun folgt die Ausschnittsteuerung. Damit kann der Notenausschnitt verändert werden, der auf dem Bildschirm dargestellt wird. Darunter folgt schließlich die Steuerung für die

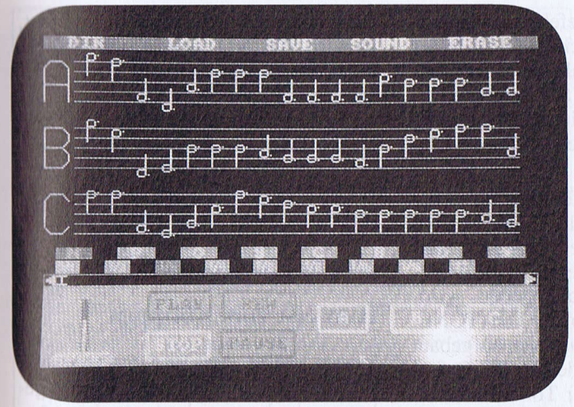


Abb. 1: Der Hauptteil mit der Noteneingabe.

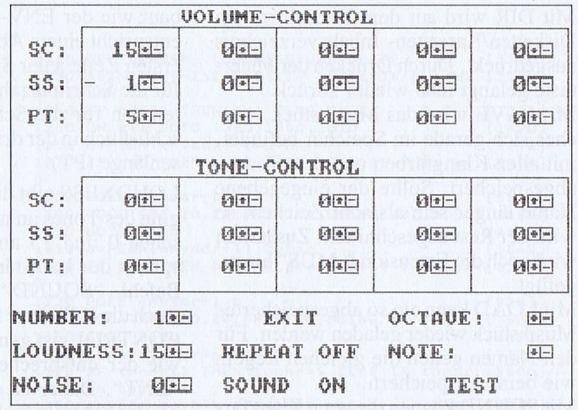


Abb. 2: In diesem Teil des Programmes werden die Töne erzeugt.

Tonausgabe. Gesteuert wird das ganze Programm mit einem kleinen Pfeil, der sich zu Beginn in der Mitte des Bildschirms befindet. Er kann wahlweise mit dem Joystick oder den Cursortasten bewegt werden, wobei die COPY-Taste der Feuertaste des Joysticks entspricht.

Die Befehlsleiste

Die einzelnen Befehle können durch Anklicken der entsprechenden Felder aufgerufen werden. Je nachdem, welcher Befehl eingeschaltet ist, erscheinen unter der Befehlsleiste verschiedene Parameter, die auf die gleiche Art und Weise gewählt werden wie die Befehle selbst. Noten, Pause, usw. werden eingegeben, indem mit dem Pfeil auf die gewünschte Position und auf eine Notenlinie gefahren wird, und nach Drücken der Feuertaste erscheint das gewünschte Zeichen auf dem Bildschirm.
Damit kein Zeichen überschrieben werden kann, rutschen alle Zeichen dahinter um eine Position nach hinten. Es können auch während des Abspielens der Stücke Noten eingegeben oder gelöscht werden.
Mit dem Befehl "N" werden Noten eingegeben. Als Parameter kann die Tonlänge gewählt werden. Die Tonhöhe wird dadurch bestimmt, auf welcher Notenlinie der Pfeil beim Eingeben gerade zeigt. Es können Noten von C bis zu G' eingegeben werden. Um eine Note um einen halben Ton zu erhöhen (z.B. Fis), muß das Kreuz ganz rechts angeklickt werden. Durch ein weiteres Anklicken kann es wieder gelöscht werden.
Durch den Befehl "P" werden Pausezeichen eingegeben. Die Pausenlänge wird auf gleiche Art und Weise bestimmt, wie zuvor bei "N". Um während des Abspielens der Noten die Oktaven zu wechseln, benötigt man den Befehl "O". Die Oktaven sind von 1 bis 8 durchnumeriert. In den Notenlinien wird der Befehl durch ein "O" und die Nummer der gewählten Oktave dargestellt. Am Anfang des Musikstückes wird immer automatisch auf die Oktave 5 geschaltet, in ihr liegt auch der Kammerton A. Um während des Abspielens z.B. eine Oktave nach oben zu gehen, muß auf die Oktave 6 geschaltet werden.
Damit beim Abspielen auch verschiedene Klangfarben verwendet werden können, gibt es den Befehl "S". Die verschiedenen Klangfarben lassen sich durch die SOUND- Funktion definieren. Die definierbaren Klangfarben sind von 1 bis 9 durchnumeriert. In den Notenlinien wird der Befehl durch ein "S" und die Nummer der gewählten Klangfarbe dargestellt. Am Anfang des Musikstückes wird automatisch die Klangfarbe 1 eingestellt.
Mit dem Befehl "C" lassen sich Noten kopieren. Zuerst muß mit "BEGIN" der Anfang und mit "END" das Ende des zu kopierenden Blockes festgelegt werden. Der so definierte Block wird zur besseren Darstellung farbig markiert. Mit "COPY" kann der definierte Block dann an jede beliebige Stelle einkopiert werden, auch in den Block selbst. Damit auch hier nicht überschrieben wird, rutschen alle Noten um die Länge des Blockes nach hinten und in diese Lücke wird der Block dann kopiert. Um die Noten etwas übersichtlicher zu gestalten, können mit "B" Taktstriche gesetzt werden. Diese haben beim Abspielen jedoch keine Wirkung.
Damit falsche Noten auch gelöscht werden können, gibt es den Befehl "D". Die so entstehende Lücke wird sofort durch die Note geschlossen, die sich hinter der gelöschten befindet.
Mit dem Befehl "*" können Leerzeichen eingefügt werden. Diese haben, ebenfalls wie die Taktstriche, beim Abspielen keine Wirkung.

Die Ausschnittsteuerung

Durch Anklicken der zwei kleinen Pfeile am linken und rechten Bildschirmrand kann der Notenausschnitt um ein Zeichen nach links bzw. rechts verschoben werden. Das "I" in dem freien Feld in der Mitte zeigt die Position des Ausschnittes im Notenspeicher an. Um größere Schritte zu machen, kann man mit dem Pfeil in dieses Feld fahren und durch Drücken der Feuertaste wandert der Ausschnitt dann an diese Position.

Die Steuerung für die Tonausgabe

Sie funktioniert ähnlich wie ein Kassettenrekorder: Durch Anklicken der PLAY-Taste wird das Musikstück von vorne abgespielt und mit STOP wird es wieder gestoppt. Um das Musikstück kurz zu unterbrechen,kann die PAUSE-Taste angeklickt werden, durch wiederholtes Anklicken wird es wieder fortgesetzt. Wenn man eine Stelle wiederholen will, kann mit REW das Stück kurz rückwärts und mit PLAY anschließend wieder normal abgespielt werden. Der weiße Balken über "SPEED" zeigt an, mit welchem Tempo das Stück abgespielt wird. Je länger es ist, um so größer ist das Tempo. Durch Anklicken eines Punktes über oder in diesem Balken kann das Tempo jederzeit verändert werden. Mit den Tasten "A", "B", "C" können die entsprechenden Kanäle an- oder abgeschaltet werden. Ist ein Kanal ausgeschaltet, so spielt es normal weiter, nur die Lautstärke ist auf Null gesetzt.

Die Kopfleisten

Um eine der fünf Funktionen aufzurufen, muß der entsprechende Name angeklickt werden.

Mit DIR wird auf dem Bildschirm ein Disketten/Kassetten- Inhaltsverzeichnis ausgedruckt. Durch Drücken der Feuertaste gelangt man wieder zurück.
Mit SAVE wird das Musikstück, welches sich gerade im Speicher befindet, mit allen Klangfarben und dem Tempo abgespeichert. Sollte der eingegebene Name länger sein als acht Zeichen, so wird der Rest abgeschnitten. Zusätzlich wird noch die Extension ".MUS" hinzugefügt.
Mit LOAD kann ein so abgespeichertes Musikstück wieder geladen werden. Für den Namen gelten die gleichen Regeln wie beim Abspeichern.
Mit SOUND können die neun Klangfarben definiert werden, die durch den Befehl "S" eingestellt werden können. Die neue Bildschirmmaske ist in folgende Abschnitte eingeteilt:
Zwei Hüllkurvenblöcke für die Lautstärke und den Ton und verschiedene Parameter. Mit den kleinen Rechtecken kann die davorstehende Zahl verändert werden. Durch Anklicken des Pluszeichens wird die Zahl um eins erhöht, durch das Minuszeichen um eins verringert. NUMBER gibt die Nummer der Klangfarbe an, die gerade angezeigt wird und auch verändert werden kann. Die Klangfarben sind von 1 bis 9 numeriert.
Die Hüllkurve für die Lautstärke (VOLUME-CONTROL) ist genauso aufgebaut wie der ENV-Befehl. Jede Spalte entspricht einem Abschnitt (1-5). In der ersten Zeile jeder Spalte steht der Wert für die Schrittanzahl (SC), in der zweiten den für die Schrittgröße (SS) und schließlich in der dritten der für die Pausenlänge (PT).

LOUDNESS gibt die Lautstärke zu Beginn des Tones an und kann Werte zwischen 0 und 15 annehmen. Dies entspricht der Lautstärke, die im BASIC-Befehl "SOUND" angegeben wird. Auch die Hüllkurve für den Ton (TONE-CONTROL) ist genau gleich aufgebaut wie der entsprechende BASIC-Befehl "ENT". Daher ist der Tonhüllkurvenblock ebenso aufgebaut wie der für die Lautstärke. Mit REPEAT kann die Tonhüllkurve wiederholt werden. Dies entspricht einer negativen Hüllkurvennummer beim ENT-Befehl.

Mit NOISE kann ein Rauschen dazugemischt werden. Der Wertebereich geht dabei von 0 bis 31. Bei 0 ist das Rauschen abgeschaltet.

Mit SOUND kann der Tongenerator zu- und abgeschaltet werden, so daß nur noch das Rauschen hörbar ist. Dies ist z.B. bei einem Schlagzeug notwendig.

Mit TEST kann die eingestellte Klangfarbe getestet werden. Die Tonhöhe für den Test kann mit OCTAVE und NOTE verändert werden. Standardwert ist das internationale A. Wird EXIT angeklickt, so wird SOUND beendet und man gelangt wieder zurück in den Hauptteil.
Die letzte Funktion ist ERASE. Mit ihr wird der gesamte Notenspeicher gelöscht. Beim Löschen bleiben die Definitionen für die Klangfarben jedoch unverändert. Wird bei der Sicherheitsabfrage die Feuertaste gedrückt, ohne daß sich der Pfeil auf "YES" befindet, so wird die Funktion abgebrochen.

Hinweise zum Abtippen

Zuerst sollte das BASIC-Programm "MUSTAR.BAS" (Listing 1) eingegeben und abgespeichert werden. Anschließend muß der DATA-Lader "LADER.BAS" (Listing 2) abgetippt und vorsichtshalber ebenfalls abgespeichert werden. Nach dem Starten des DATA-Laders wird das Binärfile "MUSTAR.BIN" erzeugt und abgespeichert. Für Kassettenbenutzer ist wichtig, daß das Binärfile direkt hinter dem Hauptprogramm abgespeichert werden sollte. Bevor das Programm dann gestartet werden kann, muß der Computer zuerst zurückgesetzt werden. Wird das Programm durch einen Bedienfehler abgebrochen und der Pfeil ist noch eingeschaltet, so kann dieser durch "CALL &9C9D" wieder abgeschaltet werden.

(Ralph Würthner/jb)

für 464-664-6128

```
100 REM ********************************    [1576]
110 REM *     M U S I C - S T A R      *    [1255]
120 REM *     by Ralph Wuerthner 1988 *     [1926]
130 REM ********************************    [1576]
140 GOSUB 4250:REM Initialisierung          [2311]
150 GOSUB 3580:REM Bild aufbauen            [1739]
160 REM *** Warteschleife ***               [2286]
170 WHILE INKEY(76)AND INKEY(9)             [2555]
180 IF taflag(1)OR PEEK(sdflag)THEN 220     [2679]
190 CALL hidep:PEN#2,1:FOR n=0 TO 3         [1363]
200 IF taflag(n)THEN PRINT#2,taste$(n);:ta  [4099]
flag(n)=0
210 NEXT n:PEN#2,3:taflag(1)=-1:PRINT#2,ta  [3237]
ste$(1);:CALL showp
220 WEND                                    [390]
230 px=PEEK(xadr)+PEEK(xadr+1)*256:py=PEEK  [3513]
(yadr)
240 IF py<8 THEN GOSUB 320:GOTO 170:REM Ko  [776]
pfleiste
250 IF py>7 AND py<128 THEN GOSUB 1940:REM  [3872]
 Note, Pause, ect. eintragen
260 IF py>127 AND py<135 THEN GOSUB 2460:R  [2437]
EM Befehlsleiste
270 IF py>135 AND py<144 THEN GOSUB 2550:R  [3357]
EM Datenleiste
280 IF py>143 AND py<152 THEN GOSUB 2850:R  [3665]
EM Ausschnittsteuerung
290 IF py>154 THEN GOSUB 2980:REM Soundste  [3230]
uerung
300 GOTO 170                                [427]
310 REM ** Kopfleiste **                    [1546]
320 CALL hidep                              [299]
330 ON INT(px/64)+1 GOSUB 360,430,550,670,  [2003]
1810
```

Listing "Musicus"

```
340 GOTO 3580                               [493]
350 REM * DIR *                             [764]
360 MODE 1:WINDOW 1,40,1,1:PAPER 1:PEN 3:C  [4527]
LS:PRINT TAB(18);"DIR"
370 WINDOW 1,40,25,25:CLS:WINDOW 1,40,2,24  [2922]
:PAPER 0:PEN 3
380 CAT                                     [96]
390 WINDOW 1,40,25,25:PAPER 1:PEN 3:PRINT   [3808]
TAB(15);"PRESS FIRE"
400 WHILE INKEY(76)AND INKEY(9):WEND        [3276]
410 RETURN                                  [555]
420 REM * LOAD *                            [486]
430 MODE 1:WINDOW 1,40,1,1:PAPER 1:PEN 3:C  [4584]
LS:PRINT TAB(18);"LOAD"
440 WINDOW 1,40,25,25:CLS:WINDOW 1,40,2,24  [2922]
:PAPER 0:PEN 3
450 CAT                                     [96]
460 WINDOW 1,40,25,25:PAPER 1:PEN 3         [1752]
470 WHILE INKEY$<>"":WEND                   [1786]
480 LINE INPUT"    NAME:",a$                [1600]
490 IF a$=""THEN RETURN                     [1147]
500 WINDOW 1,40,2,24:PAPER 0:PEN 3:CLS:WIN  [2114]
DOW 6,34,5,24
510 a$=LEFT$(a$,8)+".MUS":CALL loadms,@a$:  [2816]
sped=PEEK(ospeed)
520 FOR a=1 TO 3000:NEXT                    [1580]
530 RETURN                                  [555]
540 REM * SAVE *                            [800]
550 MODE 1:WINDOW 1,40,1,1:PAPER 1:PEN 3:C  [4279]
LS:PRINT TAB(18);"SAVE"
560 WINDOW 1,40,25,25:CLS:WINDOW 1,40,2,24  [2922]
:PAPER 0:PEN 3
570 CAT                                     [96]
580 WINDOW 1,40,25,25:PAPER 1:PEN 3         [1752]
590 WHILE INKEY$<>"":WEND                   [1786]
600 LINE INPUT"    NAME:",a$                [1600]
610 IF a$=""THEN RETURN                     [1147]
620 WINDOW 1,40,2,24:PAPER 0:PEN 3:CLS:WIN  [2114]
DOW 6,34,5,24
```

Listing "Musicus"


```
630 a$=LEFT$(a$,8)+".MUS":CALL savems,@a$   [2418]
640 FOR a=1 TO 3000:NEXT                    [1580]
650 RETURN                                  [555]
660 REM * SOUND *                           [668]
670 CALL sdpaus                             [365]
680 MODE 1:WINDOW#1,6,40,3,9:PEN#1,3:PAPER  [4772]
#1,2:CLS#1:PAPER 0:PEN 3
690 WINDOW#2,6,40,12,18:PEN#2,3:PAPER#2,2:  [3363]
CLS#2:a$=CHR$(228)+CHR$(229)
700 WINDOW#3,1,40,1,25:PEN#3,1:PAPER#3,0:W  [4800]
INDOW#4,1,40,1,25:PEN#4,3:PAPER#4,2
710 WINDOW#5,6,40,3,9:PEN#5,1:PAPER#5,2:WI  [4499]
NDOW#6,6,40,12,18:PEN#6,1:PAPER#6,2
720 LOCATE 13,2:PRINT"VOLUME-CONTROL"       [2575]
730 LOCATE 2,4:PRINT"SC:":PRINT:PRINT" SS:  [4703]
":PRINT:PRINT" PT:"
740 MOVE 78,366:DRAW 638,366,3:DRAW 638,25  [5420]
6:DRAW 638,256:DRAW 78,256
750 DRAW 78,366:FOR n=0 TO 3:MOVE 190+112*  [2423]
n,366:DRAWR 0,-110:NEXT n
760 FOR a=0 TO 2:FOR n=0 TO 4:LOCATE#1,5+n  [3957]
*7,2+a*2:PRINT#1,a$:NEXT n,a
770 LOCATE 14,11:PRINT"TONE-CONTROL"        [2464]
780 LOCATE 2,13:PRINT"SC:":PRINT:PRINT" SS  [3910]
:":PRINT:PRINT" PT:"
790 MOVE 78,222:DRAW 638,222:DRAW 638,112:  [4131]
DRAW 638,112:DRAW 78,112
800 DRAW 78,222:FOR n=0 TO 3:MOVE 190+112*  [2017]
n,222:DRAWR 0,-110:NEXT n
810 FOR a=0 TO 2:FOR n=0 TO 4:LOCATE#2,5+n  [4183]
*7,2+a*2:PRINT#2,a$:NEXT n,a
820 LOCATE 1,20:PRINT"NUMBER:     ";a$;"    [2708]
";:PAPER 2
830 PRINT" EXIT ";:PAPER 0:PRINT;"    OCTA  [2050]
VE:    ";a$
840 LOCATE 1,22:PRINT"LOUDNESS:  ";a$;SPC(  [3447]
14);"NOTE:      ";a$
850 LOCATE 1,24:PRINT"NOISE:     ";a$;SPC(  [4091]
17);:PAPER 2:PRINT" TEST ";:PAPER 0
860 GOSUB 1380                              [935]
870 LOCATE#3,37,20:PRINT#3,CHR$(48+okt):LO  [3791]
CATE#3,36,22:PRINT#3,note$(note)
880 CALL showp                              [325]
890 WHILE INKEY(76)AND INKEY(9):WEND        [3276]
900 px=PEEK(xadr)+PEEK(xadr+1)*256:py=PEEK  [4050]
(yadr):CALL hidep
910 IF py>23 AND py<63 THEN GOSUB 1000:REM  [3337]
 ENV
920 IF py>95 AND py<135 THEN GOSUB 1020:RE  [1555]
M ENT
930 IF px>135 AND px<184 AND py>151 AND py  [2720]
<160 THEN 970:REM EXIT
940 IF py>151 THEN GOSUB 1170:REM untere T  [2587]
aste
950 CALL showp:GOTO 890                     [1176]
960 REM EXIT                                [440]
970 CALL &BCA7:IF NOT taflag(3)THEN CALL s  [2417]
dcont
980 RETURN                                  [555]
990 REM ENV                                 [485]
1000 adr=envadr+2:y=24:win=5:GOTO 1030      [1503]
1010 REM ENT                                [479]
1020 adr=entadr+2:y=96:win=6                [855]
1030 a=INT((px-72)/56):n=INT((py-y)/16)     [1954]
1040 IF 86+a*56<px OR y+6+n*16<py OR a=-1   [2933]
THEN RETURN
1050 IF 79+a*56<px THEN b=-1 ELSE b=1       [1473]
1060 c=PEEK(adr+a*3+n):IF c>127 THEN c=c-2  [3851]
56
1070 c=c+b                                  [684]
1080 IF c>127 THEN c=127                    [1587]
1090 IF n=1 THEN d=-128 ELSE d=0            [757]
1100 IF c<d THEN c=d                        [965]
1110 LOCATE#win,1+a*7,2+n*2:IF c<0 THEN PR  [5101]
INT#win,"-";ELSE PRINT#win," ";
1120 PRINT#win,USING"###";ABS(c)            [2147]
1130 IF c<0 THEN c=c+256                    [1025]
1140 POKE adr+a*3+n,c                       [380]
1150 GOTO 3560                              [465]
1160 REM * Untere Tasten *                  [1116]
1170 IF px>87 AND px<103 THEN 1230          [1552]
1180 IF px>119 AND px<200 AND py>167 THEN   [1933]
1490
1190 IF px>295 AND px<311 THEN 1550         [962]
1200 IF px>239 AND px<288 AND py>183 AND p  [1855]
y<192 THEN 1650
1210 RETURN                                 [555]
```

Listing "Musicus"

```
1220 REM NUMBER, LOUDNESS und NOISE         [2190]
1230 a=INT((py-152)/16):IF 158+a*16<py THE  [3220]
N RETURN
1240 IF px<95 THEN n=1 ELSE n=-1            [1375]
1250 ON a+1 GOTO 1360,1260,1310             [1420]
1260 b=PEEK(envadr):b=b+n                   [2278]
1270 IF b<0 THEN b=15 ELSE IF b>15 THEN b=  [2465]
0
1280 POKE envadr,b                          [661]
1290 LOCATE#3,10,22:PRINT#3,USING"##";b     [3065]
1300 GOTO 3560                              [465]
1310 b=PEEK(envadr+1):b=b+n                 [1642]
1320 IF b<0 THEN b=31 ELSE IF b>31 THEN b=  [3358]
0
1330 POKE envadr+1,b                        [926]
1340 LOCATE#3,10,24:PRINT#3,USING"##";b     [1568]
1350 GOTO 3560                              [465]
1360 tonnr=tonnr+n                          [1122]
1370 IF tonnr<1 THEN tonnr=9 ELSE IF tonnr  [2260]
>9 THEN tonnr=1
1380 envadr=envblk+17*tonnr:entadr=entblk+  [2971]
17*tonnr
1390 LOCATE#3,11,20:PRINT#3,CHR$(48+tonnr)  [2212]
1400 LOCATE#3,10,22:PRINT#3,USING"##";PEEK  [3238]
(envadr)
1410 LOCATE#3,10,24:PRINT#3,USING"##";PEEK  [2925]
(envadr+1)
1420 LOCATE#4,16,22                         [552]
1430 IF PEEK(entadr)THEN PRINT#4,"REPEAT O  [5812]
N "ELSE PRINT#4,"REPEAT OFF"
1440 LOCATE#4,16,24                         [558]
1450 IF PEEK(entadr+1)THEN PRINT#4,"SOUND   [4213]
 OFF"ELSE PRINT#4,"SOUND  ON "
1460 CALL wrenvt,envadr+2,5:CALL wrenvt,en  [1751]
tadr+2,6
1470 GOTO 3560                              [465]
1480 REM REPEAT ON/OFF und SOUND ON/OFF     [1440]
1490 a=INT((py-168)/16):IF 175+a*16<py THE  [3067]
N RETURN
1500 IF PEEK(entadr+a)=0 THEN b=-1 ELSE b=  [2075]
0
1510 POKE entadr+a,-b*255:IF a THEN b=NOT   [2001]
b
1520 LOCATE#4,23,22+a*2:IF b THEN PRINT#4,  [3850]
"ON "ELSE PRINT#4,"OFF"
1530 GOTO 3560                              [465]
1540 REM OCTAVE und NOTE                    [1268]
1550 a=INT((py-152)/16):IF 159+a*16<py OR   [4112]
a>1 THEN RETURN
1560 IF px<303 THEN n=1 ELSE n=-1           [2232]
1570 IF a=1 THEN 1610                       [998]
1580 okt=okt+n:IF okt>8 THEN okt=1 ELSE IF  [3070]
 okt<1 THEN okt=8
1590 LOCATE#3,37,20:PRINT#3,CHR$(48+okt)    [2110]
1600 GOTO 3560                              [465]
1610 note=note+n:IF note>11 THEN note=0 EL   [5145]
SE IF note<0 THEN note=11
1620 LOCATE#3,36,22:PRINT#3,note$(note)     [2457]
1630 GOTO 3560                              [465]
1640 REM TEST                               [591]
1650 CALL &BCA7                             [534]
1660 FOR n=0 TO 14                          [578]
1670 a(n)=PEEK(envadr+2+n):IF a(n)>127 THE  [2944]
N a(n)=a(n)-256
1680 b(n)=PEEK(entadr+2+n):IF b(n)>127 THE  [2918]
N b(n)=b(n)-256
1690 NEXT n                                 [366]
1700 FOR n=2 TO 14 STEP 3                   [1564]
1710 IF b(n)=0 THEN b(n)=1                  [669]
1720 NEXT n                                 [366]
1730 ENV 1,a(0),a(1),a(2),a(3),a(4),a(5),a  [4602]
(6),a(7),a(8),a(9),a(10),a(11),a(12),a(13)
,a(14)
1740 IF PEEK(entadr)THEN a=-1 ELSE a=1      [2570]
1750 ENT a,b(0),b(1),b(2),b(3),b(4),b(5),b  [3547]
(6),b(7),b(8),b(9),b(10),b(11),b(12),b(13)
,b(14)
1760 adr=tondat+24*(okt-1)+2*note           [1926]
1770 IF PEEK(entadr+1)THEN th=0:ENT 1,0,0,  [4471]
0 ELSE th=PEEK(adr)+256*PEEK(adr+1)
1780 SOUND 2,th,0,PEEK(envadr),1,1,PEEK(en  [3159]
vadr+1)
1790 GOTO 3560                              [465]
1800 REM * ERASE *                          [675]
1810 MODE 1:WINDOW 1,40,1,1:PAPER 1:PEN 3:  [4643]
CLS:PRINT TAB(17);"ERASE"
1820 WINDOW 1,40,25,25:CLS:WINDOW 1,40,2,2  [2922]
4:PAPER 0:PEN 3
```

Listing "Musicus"

```
1830 LOCATE 13,9:PRINT"ARE YOU SURE ?"      [2138]
1840 LOCATE 17,11:PRINT CHR$(150);STRING$(  [4222]
3,154);CHR$(156)
1850 LOCATE 17,12:PRINT CHR$(149);"YES";CH  [2867]
R$(149)
1860 LOCATE 17,13:PRINT CHR$(147);STRING$(  [2406]
3,154);CHR$(153)
1870 CALL showp                             [325]
1880 WHILE INKEY(76)AND INKEY(9):WEND       [3276]
1890 px=PEEK(xadr):py=PEEK(yadr)            [1035]
1900 IF px>130 AND px<165 AND py>90 AND py  [3020]
<109 THEN CALL resram
1910 CALL hidep                             [299]
1920 RETURN                                 [555]
1930 REM ** Note, Pause, ect. eintragen **  [1068]
1940 IF px<20 OR px>307 THEN RETURN         [1032]
1950 npos=INT((px-20)/16)+notpos            [1746]
1960 kanal=INT((py-8)/40)                   [1107]
1970 ON keyflag+1 GOSUB 2010,2090,2120,215  [3189]
0,2180,2370,2400,2430
1980 IF keyflag<>4 THEN CALL wrlin,npos-no  [3870]
tpos,kanal
1990 GOTO 3560                              [465]
2000 REM Note                               [665]
2010 th=INT((38-((py-8)-kanal*40))/3)       [1861]
2020 IF th=-1 THEN th=0                     [1082]
2030 IF th=12 THEN th=11                    [940]
2040 IF nflag>0 THEN CALL setbyt,2,nflag-1  [4803]
+notedat(th,0)*8,npos,kanal:RETURN
2050 IF notedat(th,1)=0 THEN RETURN         [2284]
2060 CALL setbyt,2,-nflag-1+notedat(th,1)*  [3321]
8,npos,kanal
2070 RETURN                                 [555]
2080 REM Pause                              [759]
2090 CALL setbyt,3,nflag-1,npos,kanal       [2870]
2100 RETURN                                 [555]
2110 REM Oktave                             [760]
2120 CALL setbyt,4,nflag,npos,kanal         [1370]
2130 RETURN                                 [555]
2140 REM Sound                              [782]
2150 CALL setbyt,5,nflag,npos,kanal         [1483]
2160 RETURN                                 [555]
2170 REM Kopieren                           [869]
2180 IF nflag=3 THEN 2280                   [1152]
2190 IF nflag=1 THEN anf=npos               [1126]
2200 IF nflag=2 THEN ende=npos              [1017]
2210 IF anf>ende THEN a=anf:anf=ende:ende=  [1663]
a
2220 IF anf=2000 THEN anf=ende              [1179]
2230 IF ende=3000 THEN ende=anf             [1896]
2240 POKE banf,anf AND &FF:POKE banf+1,INT  [2209]
(anf/256)
2250 POKE bend,ende AND &FF:POKE bend+1,IN  [3318]
T(ende/256)
2260 POKE bkanal,kanal:CALL wrnts,notpos    [2106]
2270 RETURN                                 [555]
2280 IF anf=2000 THEN RETURN                [1118]
2290 CALL copy,anf,ende-anf+1,PEEK(bkanal)  [1658]
,npos,kanal
2300 a=ende-anf:anf=npos:ende=MIN(1197,npo  [3505]
s+a)
2310 POKE banf,anf AND &FF:POKE banf+1,INT  [2209]
(anf/256)
2320 POKE bend,ende AND &FF:POKE bend+1,IN  [3318]
T(ende/256)
2330 POKE bkanal,kanal                      [682]
2340 CALL wrnts,notpos                      [1511]
2350 RETURN                                 [555]
2360 REM Taktstrich                         [1075]
2370 CALL setbyt,1,0,npos,kanal             [1418]
2380 RETURN                                 [555]
2390 REM Loeschen                           [1316]
2400 CALL clrbyt,npos,kanal                 [957]
2410 RETURN                                 [555]
2420 REM Leerzeichen                        [862]
2430 CALL setbyt,0,0,npos,kanal             [1155]
2440 RETURN                                 [555]
2450 REM ** Befehlsleiste **                [556]
2460 a=INT((px-8)/40)                       [1927]
2470 IF a*40+31<px OR a=keyflag THEN RETUR  [927]
N
2480 POKE bkanal,4:anf=2000:ende=3000:IF k  [2834]
eyflag=4 THEN CALL wrnts,notpos
2490 keyflag=a:nflag=1:CALL hidep:GOSUB 38  [1872]
50
2500 PAPER 0:LOCATE 1,17:PRINT SPACE$(39);  [1860]
2510 ON keyflag+1 GOSUB 3920,4000,4080,415  [2231]
0,4170
```

Listing "Musicus"

```
2520 CALL showp                             [325]
2530 RETURN                                 [555]
2540 REM ** Parameterleiste **              [917]
2550 ON keyflag+1 GOTO 2580,2630,2680,2730  [1961]
,2780
2560 RETURN                                 [555]
2570 REM Notenparameter                     [1484]
2580 a=INT((px-8)/32):IF a*32+23<px OR a>8  [5940]
 OR a+1=ABS(nflag)THEN RETURN
2590 IF a=8 THEN nflag=-nflag ELSE IF nfla  [4160]
g<0 THEN nflag=-a-1 ELSE nflag=a+1
2600 CALL hidep:GOSUB 3920:CALL showp       [2198]
2610 RETURN                                 [555]
2620 REM Pausenparameter                    [1091]
2630 a=INT((px-8)/32):IF a*32+23<px OR a>4  [4920]
 OR a+1=nflag THEN RETURN
2640 nflag=a+1                              [602]
2650 CALL hidep:GOSUB 4000:CALL showp       [1607]
2660 RETURN                                 [555]
2670 REM Oktavenparameter                   [2364]
2680 a=INT((px-8)/32):IF a*32+31<px OR a>7  [4856]
 OR a+1=nflag THEN RETURN
2690 nflag=a+1                              [602]
2700 CALL hidep:GOSUB 4080:CALL showp       [954]
2710 RETURN                                 [555]
2720 REM Soundpaprameter                    [2031]
2730 a=INT((px-8)/32):IF a*32+31<px OR a>8  [4270]
 OR a+1=nflag THEN RETURN
2740 nflag=a+1                              [602]
2750 CALL hidep:GOSUB 4150:CALL showp       [1291]
2760 RETURN                                 [555]
2770 REM Kopierparameter                    [1452]
2780 IF px>7 AND px<48 THEN a=1             [764]
2790 IF px>63 AND px<104 THEN a=2           [500]
2800 IF px>119 AND px<152 THEN a=3          [1165]
2810 IF a=nflag THEN RETURN                 [1593]
2820 nflag=a:CALL hidep:GOSUB 4170:CALL sh  [1675]
owp
2830 RETURN                                 [555]
2840 REM ** Ausschnittsteuerung **          [1994]
2850 IF px<9 THEN npos=MAX(0,notpos-1):GOS  [3496]
UB 2900:GOTO 3560
2860 IF px>310 THEN npos=MIN(1180,notpos+1  [2938]
):GOSUB 2900:GOTO 3560
2870 IF px<12 THEN px=12                    [1100]
2880 IF px>307 THEN px=307                  [192]
2890 npos=(px-12)*4                         [968]
2900 IF notpos=npos THEN RETURN             [777]
2910 CALL hidep:MOVE 18+notpos/2,98:DRAWR   [4078]
12,0,0:MOVER 0,10:DRAWR-12,0
2920 MOVE 24+notpos/2,98:DRAWR 0,10         [1997]
2930 notpos=npos                            [1575]
2940 MOVE 18+notpos/2,98:DRAWR 12,0,3:MOVE  [4302]
R 0,10:DRAWR-12,0
2950 MOVE 24+notpos/2,98:DRAWR 0,10         [1997]
2960 CALL wrnts,notpos:RETURN               [2514]
2970 REM ** Soundsteuerung **               [820]
2980 IF px>178 THEN 3410:'Kanalsteuerung    [1486]
2990 IF px>24 AND px<31 THEN 3480:'Tempo    [2315]
3000 IF px>114 AND px<165 THEN 3050         [1287]
3010 IF px<67 OR px>108 THEN 3560           [944]
3020 IF py>154 AND py<173 THEN 3090:'PLAY   [1024]
3030 IF py>178 THEN 3240:'STOP              [1334]
3040 GOTO 3560                              [465]
3050 IF py>154 AND py<173 THEN 3170:'REW    [1123]
3060 IF py>178 THEN 3330:'PAUSE             [887]
3070 GOTO 3560                              [465]
3080 REM * PLAY *                           [430]
3090 IF taflag(0)THEN RETURN                [1165]
3100 CALL hidep                             [299]
3110 IF taflag(1)THEN taflag(1)=0:PEN#2,1:  [4082]
PRINT#2,taste$(1);:CALL sdon
3120 IF taflag(2)THEN taflag(2)=0:PEN#2,1:  [4684]
PRINT#2,taste$(2);:POKE chdire,PEEK(chdire
)XOR &FF
3130 taflag(0)=-1:PEN#2,3:PRINT#2,taste$(0  [3172]
);
3140 CALL showp                             [325]
3150 RETURN                                 [555]
3160 REM * REW *                            [762]
3170 IF taflag(1)OR taflag(2)THEN RETURN    [2597]
3180 POKE chdire,PEEK(chdire)XOR &FF:CALL   [1635]
hidep
3190 taflag(0)=0:PEN#2,1:PRINT#2,taste$(0)  [3178]
;
3200 taflag(2)=-1:PEN#2,3:PRINT#2,taste$(2  [3564]
);
3210 CALL showp                             [325]
```

Listing "Musicus"
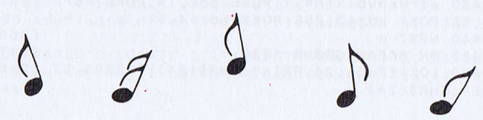

```
3220 GOTO 3560                              [465]
3230 REM * STOP *                           [430]
3240 IF taflag(1)THEN RETURN                [1393]
3250 CALL sdoff:CALL hidep                  [636]
3260 PEN#2,1:FOR n=0 TO 3                   [1870]
3270 IF taflag(n)THEN PRINT#2,taste$(n);:t  [4099]
aflag(n)=0
3280 NEXT n                                 [366]
3290 PEN#2,3:taflag(1)=-1:PRINT#2,taste$(1  [2493]
);
3300 CALL showp                             [325]
3310 RETURN                                 [555]
3320 REM * PAUSE *                          [678]
3330 IF taflag(1)THEN RETURN                [1393]
3340 CALL hidep                             [299]
3350 IF taflag(3)THEN CALL sdcont ELSE CAL  [3509]
L sdpaus
3360 taflag(3)=NOT taflag(3)                [1735]
3370 PEN#2,-taflag(3)*2+1:PRINT#2,taste$(3  [1547]
);
3380 CALL showp                             [325]
3390 GOTO 3560                              [465]
3400 REM * Kanalsteuerung *                 [1042]
3410 a=INT((px-179)/48)                     [1916]
3420 IF 212+a*48<px OR py<163 OR py>180 TH  [2036]
EN RETURN
3430 taflag(a+4)=NOT taflag(a+4)            [2441]
3440 chanflag=chanflag XOR 2^a:CALL conoff  [3950]
,chanflag
3450 PEN#2,-taflag(a+4)*2+1:CALL hidep:PRI  [4678]
NT#2,taste$(a+4);:CALL showp
3460 GOTO 3560                              [465]
3470 REM * Tempo *                          [471]
3480 IF py<160 OR py>190 THEN 3560          [1282]
3490 a=-(159-py):IF a=sped THEN 3560        [1970]
3500 CALL setspe,a,(a/sped)*PEEK(tcount)+1  [5334]
:sped=a
3510 CALL hidep                             [299]
3520 MOVE 54,78:DRAW 54,80-sped*2,1:DRAW 5  [3093]
4,18,3
3530 MOVE 56,78:DRAW 56,80-sped*2,1:DRAW 5  [1547]
6,18,3
3540 CALL showp                             [325]
3550 REM * Kurz warten *                    [1422]
3560 FOR t=1 TO 200:NEXT:RETURN             [1438]
3570 REM *** Bildschirm aufbauen ***        [1803]
3580 CALL poff:MODE 1:WINDOW#1,1,40,20,25:  [3768]
PEN#1,3:WINDOW#2,1,40,2,19
3590 PEN 3:PAPER 1:PRINT"  DIR     LOAD     [3719]
 SAVE   SOUND   ERASE  ":PAPER 0
3600 WINDOW SWAP 0,1:PAPER 2:CLS            [1578]
3610 PEN 1:PAPER 2                          [956]
3620 FOR n=2 TO 4:LOCATE 4,n:PRINT CHR$(25  [3711]
2):NEXT:LOCATE 4,5:PRINT CHR$(253)
3630 PEN 3:LOCATE 2,6:PRINT"SPEED"          [1844]
3640 MOVE 54,18:DRAW 54,80-sped*2,3:MOVE 5  [3208]
6,18:DRAW 56,80-sped*2
3650 IF PEEK(sdflag)=0 AND taflag(1)=0 THE  [5337]
N FOR n=0 TO 3:taflag(n)=0:NEXT:taflag(1)=
-1
3660 FOR n=0 TO 6:PEN-taflag(n)*2+1:PRINT   [4021]
taste$(n);:NEXT n
3670 MOVE 0,320:DRAWR 0,40,3:DRAWR 8,8:DRA  [4408]
WR 16,0:DRAWR 8,-8:DRAWR 0,-40
3680 MOVER 0,20:DRAWR-32,0                  [1653]
3690 MOVE 0,240:DRAWR 0,48:DRAWR 24,0:DRAW  [3260]
R 8,-8:DRAWR 0,-8:DRAWR-8,-8
3700 MOVER-24,0:DRAWR 24,0:DRAWR 8,-8:DRAW  [3714]
R 0,-8:DRAWR-8,-8:DRAWR-24,0
3710 MOVE 32,168:DRAWR-8,-8:DRAWR-16,0:DRA  [4142]
WR-8,8:DRAWR 0,32
3720 DRAWR 8,8:DRAWR 16,0:DRAWR 8,-8        [1705]
3730 WINDOW SWAP 0,2:PEN 3:PAPER 0          [1924]
3740 LOCATE 1,18:PRINT CHR$(254):LOCATE 40  [3183]
,18:PRINT CHR$(255);
3750 MOVE 0,96:DRAWR 638,0,1:DRAWR 0,14:DR  [3361]
AWR-638,0:DRAWR 0,-14
3760 MOVE 16,96:DRAWR 0,14:MOVE 622,96:DRA  [1856]
WR 0,14
3770 MOVE 18+notpos/2,98:DRAWR 12,0,3:MOVE  [4302]
R 0,10:DRAWR-12,0
3780 MOVE 24+notpos/2,98:DRAWR 0,10         [1997]
3790 GOSUB 3850                             [991]
3800 ON keyflag+1 GOSUB 3920,4000,4080,415  [2231]
0,4170
3810 PAPER 0:PEN 3                          [963]
3820 CALL pon:CALL wrnts,notpos             [1174]
3830 RETURN                                 [555]
```

Listing "Musicus"

```
3840 REM * Befehlsleiste zeichnen *         [1243]
3850 FOR n=0 TO 7                           [581]
3860 IF keyflag=n THEN PAPER 1:PEN 2 ELSE   [3057]
PAPER 2:PEN 3
3870 LOCATE 2+n*5,16:PRINT keyname$(n)      [3387]
3880 NEXT n                                 [366]
3890 MOVE 0,128:DRAW 640,128,0              [1534]
3900 RETURN                                 [555]
3910 REM Notenparameter                     [1484]
3920 FOR n=0 TO 7                           [581]
3930 IF ABS(nflag)=n+1 THEN PAPER 1:PEN 2   [1930]
ELSE PAPER 2:PEN 3
3940 LOCATE n*4+2,17:PRINT CHR$(n*2+231);C  [3348]
HR$(n*2+230)
3950 NEXT                                   [350]
3960 IF nflag<0 THEN PAPER 1:PEN 2 ELSE PA  [2635]
PER 2:PEN 3
3970 LOCATE 34,17:PRINT CHR$(247);CHR$(246  [1679]
)
3980 RETURN                                 [555]
3990 REM Pausenparameter                    [1091]
4000 RESTORE 4060                           [938]
4010 FOR n=0 TO 4                           [572]
4020 IF nflag=n+1 THEN PAPER 1:PEN 2 ELSE   [2567]
PAPER 2:PEN 3
4030 READ a:LOCATE n*4+2,17:PRINT CHR$(a+1  [2889]
);CHR$(a)
4040 NEXT n                                 [366]
4050 RETURN                                 [555]
4060 DATA 230,232,236,240,244               [1114]
4070 REM Oktavenparameter                   [2364]
4080 a=7                                    [337]
4090 FOR n=0 TO a                           [803]
4100 IF nflag=n+1 THEN PAPER 1:PEN 2 ELSE   [2567]
PAPER 2:PEN 3
4110 LOCATE n*4+2,17:PRINT" ";CHR$(n+49);"  [2332]
"
4120 NEXT n                                 [366]
4130 RETURN                                 [555]
4140 REM Soundparameter                     [1158]
4150 a=8:GOTO 4090                          [735]
4160 REM Kopierparameter                    [1452]
4170 FOR n=0 TO 2                           [566]
4180 IF nflag=n+1 THEN PAPER 1:PEN 2 ELSE   [2567]
PAPER 2:PEN 3
4190 LOCATE n*7+2,17                        [1136]
4200 IF n=0 THEN PRINT"BEGIN":GOTO 4220     [1895]
4210 IF n=1 THEN PRINT" END "ELSE PRINT"CO  [1091]
PY"
4220 NEXT n                                 [366]
4230 RETURN                                 [555]
4240 REM *** Initialisierung ***            [1137]
4250 DEFINT a-z:POKE 0,&C9:ON BREAK GOSUB   [3608]
4930
4260 MODE 1:BORDER 0:INK 0,0:INK 1,26:INK   [2698]
2,15:PEN 2
4270 LOCATE 8,2:PRINT CHR$(150);STRING$(21  [3321]
,154);CHR$(156)
4280 LOCATE 8,3:PRINT CHR$(149);            [2182]
4290 PEN 1:PRINT" M U S I C - S T A R ";:P  [3330]
EN 2:PRINT CHR$(149)
4300 LOCATE 8,4:PRINT CHR$(147);STRING$(21  [3152]
,154);CHR$(153):PEN 1
4310 LOCATE 7,9:PRINT CHR$(164);" BY RALPH  [3620]
 WUERTHNER 1988"
4320 resram=&9600:copy=&9628:setbyt=&9692:  [5172]
clrbyt=&96E2:wrnts=&9726:wrlin=&976E
4330 sdon=&9EE4:sdoff=&9F0E:sdpaus=&9EA3:s  [3533]
dcont=&9EB8
4340 conoff=&9ED0:setspe=&9F23:chdire=&A5C  [2969]
0:ospeed=&A406
4350 envblk=&8CA0:entblk=&8D50:tondat=&A32  [3127]
7:wrenvt=&A41C
4360 pon=&9C75:poff=&9C9D:showp=&9CAF:hide  [3363]
p=&9CC6:xadr=&9E39:yadr=&9E3B
4370 loadms=&A494:savems=&A510              [1558]
4380 tcount=&A5C1:sdflag=&A5B5:banf=&A60E:  [2954]
bend=&A610:bkanal=&A612
4390 IF HIMEM<&7080 THEN 4450               [689]
4400 SYMBOL AFTER 256:MEMORY &707F          [1643]
4410 LOAD"MUSTAR.BIN":CALL resram           [3569]
4420 FOR n=1 TO 9                           [819]
4430 adr=envblk+n*17:POKE adr,15:POKE adr+  [3678]
2,15:POKE adr+3,255:POKE adr+4,5
4440 NEXT n                                 [366]
4450 ON BREAK GOSUB 4920                    [647]
4460 LOCATE 15,24:PRINT CHR$(24);"PRESS FI  [2968]
RE";CHR$(24)
```

Listing "Musicus"

```
4470 WHILE INKEY(76)AND INKEY(9):WEND       [3276]
4480 MODE 1:BORDER 0,0:INK 1,16:INK 2,8:IN [2233]
K 3,26
4490 SYMBOL AFTER 228                       [1452]
4500 SYMBOL 254,0,3,15,63,63,15,3:SYMBOL 2 [3349]
55,0,192,240,252,252,240,192
4510 SYMBOL 252,126,60,126,60,126,60,126,6 [4706]
0:SYMBOL 253,126,60,126,60,126,60,126
4520 SYMBOL 230,0,128,144,48,16,16,16:SYMB [3293]
OL 231,0,8,24,9,9,10,2
4530 SYMBOL 232,0,128,184,8,56,32,56:SYMBO [3442]
L 233,0,8,24,9,9,10,2
4540 SYMBOL 234,0,128,168,40,56,8,8:SYMBOL [3325]
 235,0,24,8,25,9,26,2
4550 SYMBOL 236,0,128,168,40,56,8,8:SYMBOL [2163]
 237,0,8,24,9,9,10,2
4560 SYMBOL 238,0,128,184,40,16,40,56:SYMB [3812]
OL 239,0,24,8,25,9,26,2
4570 SYMBOL 240,0,128,184,40,16,40,56:SYMB [3057]
OL 241,0,8,24,9,9,10,2
4580 SYMBOL 242,0,0,80,208,92,84,92:SYMBOL [3798]
 243,0,49,17,50,18,52,4
4590 SYMBOL 244,0,0,80,208,92,84,92:SYMBOL [3184]
 245,0,17,49,18,18,20,4
4600 SYMBOL 246,0,64,224,64,64,224,64:SYMB [3163]
OL 247,0,4,15,4,4,15,4
4610 SYMBOL 228,255,129,153,189,153,129,25 [4326]
5:SYMBOL 229,254,2,2,122,2,2,254
4620 DIM keyname$(7),notedat(11,1),taste$( [3642]
6),taflag(6),note$(11),a(14),b(14)
4630 notpos=0:keyflag=7:nflag=15:chanflag= [3897]
7:okt=5:note=9:tonnr=1:taflag(1)=-1
4640 sped=PEEK(ospeed):POKE bkanal,4:CALL  [3135]
conoff,chanflag:CALL sdoff
4650 FOR n=envblk TO envblk+16:POKE n,0:NE [3697]
XT
4660 FOR n=entblk TO entblk+16:POKE n,0:NE [2547]
XT
4670 RESTORE 4690                           [892]
4680 FOR n=0 TO 7:READ a$:keyname$(n)=" "+ [2682]
a$+" ":NEXT
4690 DATA N,P,O,S,C,B,D,*                   [1141]
4700 FOR m=0 TO 1:FOR n=0 TO 11:READ noted [3075]
at(n,m):NEXT n,m
4710 DATA 0,2,4,5,7,9,11,12,14,16,17,19     [1662]
4720 DATA 1,3,0,6,8,10,0,13,15,0,18,20      [1130]
4730 FOR n=0 TO 11:READ note$(n):NEXT       [1523]
4740 DATA " C","#C"," D","#D"," E"," F","# [2766]
F"," G","#G"," A","#A"," H"
4750 FOR n=72 TO 77:KEY DEF n,0,0,0,0:NEXT [1740]
4760 r1$=CHR$(150)+STRING$(4,154)+CHR$(156 [4231]
)+CHR$(10)+STRING$(6,8)+CHR$(149)
4770 r2$=CHR$(149)+CHR$(10)+STRING$(6,8)+C [5872]
HR$(147)+STRING$(4,154)+CHR$(153)
4780 taste$(0)=CHR$(31)+CHR$(9)+CHR$(1)+r1 [3868]
$+"PLAY"+r2$
4790 taste$(1)=CHR$(31)+CHR$(9)+CHR$(4)+r1 [3508]
$+"STOP"+r2$
4800 r1$=CHR$(150)+STRING$(5,154)+CHR$(156 [3641]
)+CHR$(10)+STRING$(7,8)+CHR$(149)
4810 r2$=CHR$(149)+CHR$(10)+STRING$(7,8)+C [4829]
HR$(147)+STRING$(5,154)+CHR$(153)
4820 taste$(2)=CHR$(31)+CHR$(15)+CHR$(1)+r [3442]
1$+" REW "+r2$
4830 taste$(3)=CHR$(31)+CHR$(15)+CHR$(4)+r [3339]
1$+"PAUSE"+r2$
4840 r1$=CHR$(150)+STRING$(3,154)+CHR$(156 [2895]
)+CHR$(10)+STRING$(5,8)+CHR$(149)+" "
4850 r2$=" "+CHR$(149)+CHR$(10)+STRING$(5, [4368]
8)+CHR$(147)+STRING$(3,154)+CHR$(153)
4860 FOR n=0 TO 2                           [566]
4870 taste$(n+4)=CHR$(31)+CHR$(23+n*6)+CHR [4612]
$(2)+r1$+CHR$(65+n)+r2$
4880 taflag(n+4)=-1                        [1516]
4890 NEXT n                                 [366]
4900 RETURN                                 [555]
4910 REM ** Abbruch-Routine **             [1439]
4920 CALL poff:CALL sdoff                   [672]
4930 MODE 1:INK 0,1:INK 1,24:BORDER 1:PAPE [2673]
R 0:PEN 1
4940 END                                    [110]
```



Listing "Musicus"

```
100 REM **   DATA-Lader zu MUSIC-STAR   ** [1440]
110 REM ** erzeugt das File  MUSTAR.BIN ** [2104]
120 REM ** (C) by  Ralph Wuerthner 1988 ** [1254]
130 MEMORY &95FF:DEFINT a-z:adr=&9600       [1509]
140 FOR z=250 TO 1500 STEP 10               [1088]
150 sum=10:READ z$                          [353]
160 FOR n=0 TO 31                           [577]
170 b=VAL("&"+MID$(z$,n*2+1,2))             [1544]
180 POKE adr+n,b:sum=(sum+b) XOR b          [581]
190 NEXT n                                  [366]
200 READ n:IF n<>sum THEN PRINT"Fehler in  [3661]
Zeile";z:STOP
210 adr=adr+32                              [411]
220 NEXT z                                  [354]
230 PRINT"Alles O.K."                      [1275]
240 SAVE"MUSTAR.BIN",B,&9600,&FAB          [1639]
250 DATA F3218270CD159621E279CD1596214283C [3899]
D1596FBC92B2B3606232336005D5413,1798
260 DATA 015C09EDB03606C9DD7E00DD4604CB27C [3192]
B27B0DD5E02DD5603DD4E06DD4607DD,3026
270 DATA 6E08DD6609D5C5F5E603118270B728091 [3790]
1E2793D2803114283CB25CB14191100,2370
280 DATA 8ECB21CB10EDB0F1CB3FCB3FED47C1D12 [4407]
1008EC546234E23E5D5ED57CDA196D1,4910
290 DATA E1C1137AFE0438047BFEAED00B78B120E [5103]
2C9DD7E00DD5E02DD5603DD4E04DD46,3358
300 DATA 06F3C5DD218270B7280BDD21E2793D280 [2852]
4DD214283CB23CB12D5215C09B7ED52,3938
310 DATA 444D0B0B78B1280EDDE5E1115C09192BE [3755]
5D12B2BEDB8D1C1DD19DD7000DD7101,3570
320 DATA FBC9DD7E00DD5E02DD5603F3218270B72 [3314]
80921E2793D2803214283CB23CB1219,2398
330 DATA E5215C09B7ED52444D0B0B78B1280FE1E [5366]
52323D1EDB02B36002B3600FBC9E136,3726
340 DATA 00233600FBC9CB23CB12ED5313A6D5CDC [3734]
69CCDAF9BD12182700E00CD519721E2,4078
350 DATA 790E01CD5197214283OE02CD5197C3AF9 [6251]
C19D5511E000612CD5D97D1C9C54623,3478
360 DATA 4E23D5E5CDAA97E1D11CC110F0C9CDC69 [3885]
CDD4600DD5E02D5C5CDE29BC1782182,4254
370 DATA 70B7280921E2793D2803214283ED5B13A [5128]
619D14BCB2316001933E129159502807,3094
380 DATA 47CD5D97C3AF9C46234E78B7C842FE01C [4755]
A2A99FE02200C79E607CB39CB39CB39,3694
390 DATA C34698FE03200479C3CA98D52105C0420 [2519]
411900119110FDD1CB23CB2316001906,2478
400 DATA 4FFE0428020653C578CDF097C179C6304 [3796]
FCD06B9E5EB2100380600CB21CB10CB,3946
410 DATA 21CB10CB21CB1009EB0608C5E51ACB3FC [5730]
B3FCB3FCB3F4FAF0604CB3FCB393002,3558
420 DATA F68810F6B677231A4FAF0604CB3FCB393 [2784]
002F68810F6B67713E1CD2B9EC110CB,4606
430 DATA E12323C309B9F5C5C5214999CB2106000 [4404]
94E2346E560691600CB23CB2319D1C1,3094
440 DATA D5110000CB38300511900118O7CB38300 [3331]
311200319D179FE02DCB798EB7E2B2B,2378
450 DATA BEEBCCA898C111739979FE0B380711500 [3544]
01911639AF1E5EBB72807111E004719,3238
460 DATA 10FDD1EB060F1871E5114E001911539B0 [4848]
608CD1999E1C9E5014F18097EE6CCF6,3630
470 DATA 30772336F02336F0E1C9F5CB23CB236B2 [3623]
60078010500FE012005019511807FE,3818
480 DATA 0220030125030 9F111A1E8B7281A11F1C [3960]
83D28143D2819118F9B3D2803119F9B,1882
490 DATA 01F1C0090608181111911639B060318091 [3410]
1A1D01911699B0613E51AB67723131A,1642
500 DATA B67713E1CD2B9E10F0C921C7F6CB23CB2 [2896]
3160019041190011910FD06197EE677,3642
510 DATA F68077CD2B9E10F5C947D147D1F7F8F7F [4028]
8F7E0F7C8F7C8A7F0A7F0A7D8A7D8A7,6418
520 DATA C057E857E857D057D007F807E007E007C [3442]
807C8000000000000000000000000,2742
530 DATA 0000000007788884488448847788000 [3555]
00044004400440044004400440044004400,2530
540 DATA 44004477CC884488448844847788000004 [3503]
40044004400440044004400440044004400,1626
550 DATA 4477CC8844884488557788000000044004 [3978]
40044004400440044004400440044004477,1830
560 DATA CCFFCCFFCCFFCC7788000000440044004 [3524]
40044004400440044004400440044004477CCFF,2562
570 DATA CCFFCCFFDD7788000000440066005 5004 [3197]
40044004400440044004477CCFFCCFF,3502
580 DATA CCFFCC7788000000440066005500440044 [4563]
4004400440044004477CCFFCCFFCCFF,3038
590 DATA DD7788000000440066005500440066005 [2754]
500440044004477CCFFCCFFCCFFCC77,3218
600 DATA 880000000077888844884488447788000 [2356]
000000000000000000000000000000000,1394
610 DATA 0000007788884488448844FF888800880 [3117]
```

Listing "Musicus"


```
088008800880088008800880088000,1634
620 DATA 007788885588448844FF8888008800880 [3692]
088008800880088008800880080000077,3918
630 DATA 88FFCCFFCCFFCCFF8888008800880880 [3109]
088008800880088008800000007788FF,4634
640 DATA DDFFCCFFCCFF888800880088008800880 [4093]
088008800880088000000007788FFCCFF,4602
650 DATA CCFFCCFF888800880088008800880880 [2836]
0AA00CC008800000007788FFDDFFCCFF,4934
660 DATA CCFF888800880088008800880088008AA0 [3492]
0CC008800000007788FFCCFFCCFFCCFF,4910
670 DATA 8888008800880088008AA00CC008800AA00CC0 [4004]
08800114433EE1144114433EE114400,2458
680 DATA 000000FFFFFFFFFFFF220011880044004 [3623]
400880088110011002200220011880O,1654
690 DATA 6600CC33004400880088006600110022 4 [4242]
411CC008800881100110022002200022,1678
700 DATA 4411CC008844883300110022002200011 [3465]
203C5053A12A6B8CAC29BAF5FCD0D9C,1870
710 DATA 181C2A13A6CB3CCB1D1E00CD579CC5D5E [3381]
50E01CD0D9CE1D1C11C230D20EDC110,4006
720 DATA D1C93E1293C84F3A12A6B83E00201E2A1 [3065]
3A6CB3CCB1D160019CD579CC5D5E50E,2518
730 DATA 01CD0D9CE1D1C11C230D20EDC9D5ED472 [4824]
1C5BE1190010419 10FDD1CB23CB2316,3682
740 DATA 0019CB21CB21ED57571E07CD499C0605C [3841]
51101F0CD499CED57571E05CD499CC1,3770
750 DATA 10EEED57571E03435841E5722310FCE1C [4272]
D2B9E4310F2C9D5E5ED5B0EA6B7ED52,4190
760 DATA 380F2A10A6D1D5B7ED523805E1D13E0FC [3256]
9E1D1AFC93A3D9EB7C03EFF323D9E32,4642
770 DATA 16A6213E9E2219A6ED5B399E2A3B9ECD7 [3182]
89D2127A611D59C0681C3D7BC3A3D9E,4218
780 DATA B7C8AF323D9ECDBB9D2127A6C3DDBC3A1 [4863]
6A6B7C0F33EFF3216A6ED5B399E2A3B,4666
790 DATA 9ECD789DFBC93A16A6B7C8F3AF3216A6C [3951]
DBB9DFBC9CD24BBE521779E0100047E,4430
800 DATA 23C5E5CD1EBBE1C1280137CB1110F0E17 [4862]
CB1E60F2007213E9E2219A6C9F5CD69,3994
810 DATA 9DF1ED5B399E2A3B9ECB3F3003B7ED42C [3660]
B3F300109EBCB3F3003B7ED42CB3F30,3378
820 DATA 0109EB3EFABD30042EC318073EC3BD300 [4713]
22E00E521FAFFB7ED523005113B0118,3502
830 DATA 0B213B01B7ED523003110000E13A399EB [2648]
B20053A3B9EBDC8ED53399E223B9E3A,2934
840 DATA 16A6B7C8CD03B918202A19A67E4F0600F [4151]
E03C8232219A6C97BE6033215A6CDFD,3590
850 DATA 9D2217A6CDA19D184CD5E5CDBB9DE1D17 [3743]
BE6033215A6CDFD9D2217A6CDA19D18,5146
860 DATA 342A17A6111DA60605480602E57E12231 [4960]
310FA41E1CD2B9E10EFC92A17A6111D,3034
870 DATA A60605480602E51A77231310FA41E1CD2 [4044]
B9E10EFC93A15A6217B9EB728074711,3454
880 DATA 0A001910FDED5B17A6EB0605E51AB6772 [4011]
3131AB67713E1CD2B9E10F0C9CB3ACB,2962
890 DATA 1B535DD5CB3BCB3BCB3B430421B0BF115 [3618]
0001910FDD1D57BE607280747110008,3678
900 DATA 1910FDD15ACB3B160019C97CC60867D07 [3285]
DD6B06F7CDE3F67C9A0006400000100,3742
910 DATA 0000010000000100000000100000001000 [3585]
0010000010000010000010000010000,10
920 DATA 0101010101010101020101020101020010 [3754]
1020202020203010802OOFF00CC00AA,602
930 DATA 00990000887788660055004488004433C [4058]
C33002288224400221 1EE1188114411,2042
940 DATA 220011F33EFF32BDA50100033E08C5CDE [2778]
F9FC13C10F8FBC9F3AF32BDA521CCA5,4478
950 DATA 3E080603C54ECDEF9F233CC110F6FBC9F [4157]
3DD7E002FE6074FCB27CB27CB27B132,4450
960 DATA C2A5FBC9F3CD169F21CFA5063F3600231 [2784]
0FB2107A411ACA5011500EDB03A06A4,3770
970 DATA 32C1A5213BA611329F0681C3E0BCF3213 [4954]
BA6CDE6BCF301000BAFC5CDEF9FC13C,4482
980 DATA 10F8C9F3DD7E023206A4DD7E0032C1A5F [4071]
BC93ABDA5B7C0DDE53ABCA53D32BCA5,3594
990 DATA F5CD7FA1F1F5CD56A2F1F5CD55A0F1DDE [4045]
1B7C03E0332BCA5AF32BFA53AC0A5B7,5010
1000 DATA C40FA03AC1A53D20083EFF32BFA53A06 [5555]
A432C1A5CD829FCDDE9F3AB5A5B7CA0E,4742
1010 DATA 9FC921F0A511C4A50603C5010200EDB0 [4451]
0E0809C110F43AC3A51213D501000321,2746
1020 DATA EEA511F5FF7EFE01CB111910F8CB21CB [3279]
21CB21D13AC2A5E638B11213D521CFA5,3B90
1030 DATA 06037E1213C5010B0009C110F5E13AC2 [3631]
A54F0603CB3930023600231 0F7C921CE,2790
1040 DATA A53E0BF34E3DCDEF9F2BB720F7FBC9F5 [5380]
06F4ED7906F6ED78F6C0ED79E63FED79,4850
1050 DATA 06F4ED4906F64FF680ED79ED49F1C9AF [4318]
32C0A53AC1A5473A06A4903C32C1A506,3686
```

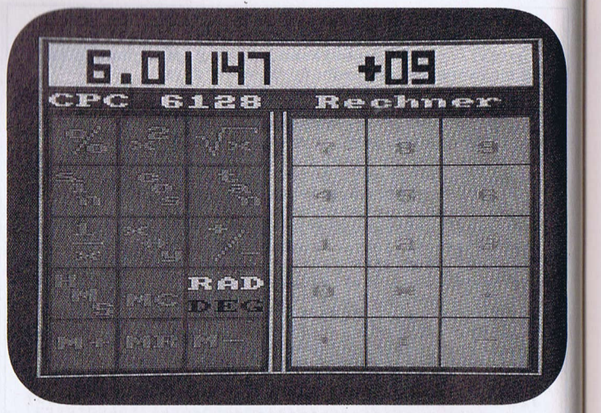
Listing "Musicus"

```
1060 DATA 03783DCD05A2DD7E08DD9607DD770710 [3583]
F03ABEA52F32BEA5010400B7280301FC,3082
1070 DATA FF3E0321B6A55E2356EB09EB722B7323 [4346]
233D20F2C94FCD3CA1D8C579CDEDA0D1,4178
1080 DATA 78B7C80505284C05282F05280A21ACA5 [3623]
160019714B18E32127A30D79B72809D5,1902
1090 DATA 111800411910FDD14B7BCB210600EB21 [3870]
AFA5097323724F18C12101A40600094E,2914
1100 DATA 21F9A3097B4EF5CD46A1F10600110000 [4281]
C316A2C579E6074F060021F9A3094E21,2842
1110 DATA ACA5160019467BC5F5CD46A1F1D1C1D5 [4515]
5FCB23160021AFA5195E2356EBC39CB,3550
1120 DATA 39CB810600095E2356C1C316A221B6A5 [3990]
4F5FCB231600195E2356EB7EFE062828,2782
1130 DATA FE023011C53ABEA5010200B7280301FE [3581]
FF09C118E646234E233ABEA5B728042B,3282
1140 DATA 2B2B2BEB722B73C9EB722B73793CFE03 [5261]
20013C2F21B5A5A6770600C9CD05A2DD,3330
1150 DATA 7E07B7C837C9CD05A2DD700ADD7107DD [5167]
710878CD14A37EDD7700237EDD7709B7,4514
1160 DATA 280332C3A5237EDD7702DD7704237EDD [3014]
7705237EDD7701DD7706DD360300C9CD,3762
1170 DATA 05A2DD7E07B7C83ABFA5B7280FDD7E07 [4883]
3DDD7707B72005DD360000C9DD7E03FE,3294
1180 DATA 05C8DD7E04B72005CDD1A118EFDD7E01 [5598]
B728053DDD7701C0DD7E06DD7701DD7E,4218
1190 DATA 00DD8605E60FDD7700DD7E023DDD7702 [3226]
C0DD7E033CDD7703FE05C8DD7E0ACD14,3870
1200 DATA A32323DD5E037B87835F1600197EB728 [4528]
E0DD7704DD7702237EDD7705237EDD77,1518
1210 DATA 06DD7701C9DD21CFA5B7C8DD21DAA53D [4046]
C8DD21E5A5C9CD03A378DD7709DD7300,4950
1220 DATA DD7201CD22A37EDD7708237EB7280DDD [3878]
360000DD360100DD360405C9237EDD77,2850
1230 DATA 03DD7705237EDD7706237EDD7702DD77 [4107]
07DD360400C9CD03A3DD7E08B72820DD,3006
1240 DATA 7E09CD22A323237E0604232323B610FA [3641]
B72012DD360800180CDD7E08B72006DD,2434
1250 DATA 7E04FE05C8DD7E05B72005CDC5A218E9 [4002]
DD7E02B728053DDD7702C0DD7E07DD77,1694
1260 DATA 02DD6E00DD6601DD5E0616007BE68028 [4396]
0216FF197CE60FDD7500DD7701DD7E03,3890
1270 DATA 3DDD7703C0DD7E043CDD7704FE052009 [3657]
DD7E08B7C8DD360400DD7E09CD22A323,3082
1280 DATA 23DD5E047B87835F1600197EB728D6DD [2320]
7705DD7703237EDD7706237EDD7707DD,2230
1290 DATA 7702C9DD21F0A5B7C8DD21FAA53DC8DD [3093]
2104A6C921A08C5F87878787835F1600,5230
1300 DATA 19C921508D18F0EE0E180E4D0D8E0CDA [3227]
0B2F0B8F0AF7096809E1086108E90777,2782
1310 DATA 070C07A7064706ED0598054705FC04B4 [4211]
0470043104F403BC03860353032403F6,1758
1320 DATA 02CC02A4027E025A023802 1802FA01DE [4026]
01C301AA0192017B01660152013F012D,2146
1330 DATA 011C010C01FD00EF00E100D500C900BE [4259]
00B300A9009F0096008E0086007F0077,3078
1340 DATA 0071006A0064005F005900540050004B [3354]
00470043003F003C003800350032002F,962
1350 DATA 002D002A0028002600240022000201E [3408]
001C001B0019001800160015001400 13,390
1360 DATA 0012001100100000F000E000D000D000C [4139]
000B000B000A0009001008OC04060203,150
1370 DATA 0100010305070A01010187A387A387A3 [3396]
07827OE279428303000OFF00DD7E00CD,2630
1380 DATA B4BBDD5E02DD5603210201060F1AC5D5 [3620]
E5CD49A4E1D1C1137DC6026FFE082006,3054
1390 DATA 2E027CC6076710E5C9F5CD75BBC10405 [3893]
3E20F25BA478ED44473E2DCD5ABB78FE,3826
1400 DATA 643E200E00380878D664473E310EFFCD [4626]
5ABB1E0ACD7BA40EFF1E01781 62F1493,2746
1410 DATA 30FC83477A0C0DC25ABBFE30C25ABB3E [4492]
20C35ABBED731BA6DD6E00DD66014623,3350
1420 DATA 5E2356EB11008ECD77BCD292BCCD0096 [3754]
CDF8A43206A421B18CCDD7A421618DCD,4994
1430 DATA D7A4218270CDE5A421E279CDE5A42142 [4540]
83CDE5A4C37ABC0699C5E5CDF8A4E1C1,5858
1440 DATA 772310F5C9E5CDF8A4E1FE06C87723E5 [3844]
CDF8A4E1772318EDCD80BC300CFE19C0,5478
1450 DATA F5CD80BC3003C180C9ED7B1BA6C37DBC [3255]
ED731BA6DD6E00DD660146235E2356EB,3602
1460 DATA 11008ECD8CBCD292BC3A06A4CD88A521 [3647]
B18CCD50A521618DCD50A5218270CD5E,4198
1470 DATA A521E279CD5EA5214283CD5EA5C38FBC [4295]
06997EC5E5CD88A5E1C12310F5C9E501,3602
1480 DATA 5C09092B2B7E0B0BFE062819B728F4E1 [4033]
03037EC5E5CD88A5E1C1230B78B120F2,3066
1490 DATA 3E06C388A5E118F8FE1A280FFE192814 [2553]
CD95BCD8ED7B1BA6C392BC3E19CD90A5,3446
1500 DATA 3E0118EC3E19CD90A5AF18E400000000 [4874]
000000000000000000000000000000000,1702
```

Listing "Musicus"

Taschenrechner im Großformat

Meistens gebraucht man seinen Computer zu verschiedenen Dingen wie Textverarbeitung, Datenverwaltung oder als Spielgerät. Die eigentliche Funktion, nämlich ein Rechengerät zur sein, geht dabei allmählich ganz unter, dazu hat man meistens seinen programmierbaren Taschenrechner zu Verfügung. Daß man den Spieß auch mal umdrehen kann, zeigt Ihnen unser Programm, welches, in BASIC geschrieben, aus Ihrem CPC einen vollwertigen technisch- wissenschaftlichen Taschenrechner macht, mit dem Sie die komplexesten Berechnungen durchführen können.



Benutzen Sie doch einmal Ihren CPC als technisch- wissenschaftlichen Taschenrechner.

Alle Funktionen sind über die Tasten des Zehnerblockes einschließlich den Cursor- Steuertasten zu bedienen. Der Bildschirm wird in zwei Hauptteile und in ein Ausgabefenster aufgeteilt. Es kann mit der Taste ']' zwischen zwei Ausgabeformaten (Zahlengröße) gewählt werden.

Auf der rechten Bildschirmseite sind die Funktionen der Zehnertastatur dargestellt, die im Normalzustand ausgeführt werden. Dieses sind erstens die Ziffereingabe und zweitens die vier Grundrechenarten. Links davon ist die zweite Ebene abgebildet, die mit SHIFT erreichbar ist. Sie bedarf eigentlich keiner weiteren Erklärung, da die Symbole allein schon aussagekräftig genug sind. Außer Prozentrechnung, Potenzieren, Radizieren, Winkelfunktionen, 1/X, Vertauschen und Vorzeichenwechsel gibt es auch die Funktion 'HMS', die Eingaben in der Form Stunden (oder Grade), Minuten und Sekunden in gegebene Zeiten oder Winkel in dezimalen Stunden bzw. Grade umsetzt.

Die Funktionen

Mit SHIFT und '.' des Zehnerblockes ist die Wahl des Modus zwischen Bogenmaß und Winkelmaß möglich. Der jeweilig eingestellte Modus wird durch weiße Schriftzeichen angezeigt.

Mit SHIFT bzw. CONTROL und den Cursortasten sind die beiden arithmetischen Speicher anwählbar. Dabei bedeutet 'M+', daß der angezeigte Wert zu dem abgespeicherten Wert dazuaddiert wird, während bei 'M-' subtrahiert wird. Ob sich Werte im Speicher befinden, kann am Wechsel der Farbe für 'MR' (1.Speicher mit SHIFT) und 'MC' (2.Speicher mit CONTROL) erkannt werden. 'MC' löscht den entsprechenden Speicherinhalt und 'MR' gibt den Speicherinhalt aus. Zusätzlich zu diesen zwei arithmetischen Speichern können Sie noch über 26 Konstantenspeicher verfügen, die durch die alphanumerische Tastatur bedient werden. Durch Drücken einer beliebigen Taste im Normalmodus wird der angezeigte Wert abgespeichert. Über SHIFT 'A..Z' ist der Speicherinhalt wieder abrufbar.

Zurück zum Zehnerblock.

Die dritte Ebene, mit CONTROL anzuwählen, enthält folgende Optionen:

F7 = berechnen des prozentualen Unterschiedes zwischen den Inhalten von X- und Y-Register.

F8 = dient zur Berechnung beliebiger Potenzen.

F9 = Radizieren mit beliebigen Exponenten.

F4,F5,F6 = wandeln den Sinus, Cosinus bzw. Tangens in einen entsprechenden Winkel um.

F1 = löscht die vier Rechenregister.

F2 = zyklisches Vertauschen der Rechenregister nach 'unten'.

F3 = löscht alle Register und Speicher.

F0 = wandelt in dezimaler Form gegebene Zeiten oder Winkel in die Form Stunden (Grad), Minuten Sekunden um.

'.' = schreibt die Zahl PI in das angezeigte X-Register

Weitere 'festverdrahtete' Funktionen sind:

DEL = löscht das angezeigte X- Register,

'[' + 0..9 = wählt die Nachkommastellen. Wer den Dezimalpunkt lieber durch das Komma dargestellt haben möchte, wählt diese Funktion mit der Taste 'Semikolon' aus.

Auch an die Ausgabe auf einen angeschlossenen Drucker ist gedacht worden. Mit SHIFT ENTER wird das letzte Ergebnis gedruckt. CONTROL ENTER wirkt als Schalter. Nach dem ersten Drücken werden alle nachfolgenden Rechenoperationen auch ausgedruckt. Durch erneutes Drücken der Tasten CONTROL und ENTER wird die Druckausgabe wieder abgeschaltet.

Hier noch die Sondertasten mit ihren Funktionen:

'@' = gibt die vier Rechenregister X, Y, Z, Z0 aus,

'>' = die Fakultät der eingegebenen Zahl

'<' = ergibt den Rauminhalt einer Kugel

'_' = Heronsche Dreiecksflächenformel

'{' = Sinussatz; gegeben Winkel, Seite, Winkel

'}' = Sinussatz; gegeben Seite, Winkel, Winkel

'~' = Sinussatz; gegeben Seite, Seite, Winkel

'(' = Kosinussatz; gegeben Seite, Winkel, Seite

')' = Kosinussatz; gegebenSeite, Seite, Seite.

Zum Üben

Nachdem Sie jetzt erfahren haben, welche vielfältigen Möglichkeiten im Programm zu nutzen sind, sollen Ihnen die nachstehenden Beispiele den Einsatz des Programmes näherbringen. Die Rechenlogik basiert auf der sogenannten 'umgekehrten polnischen Notation'. Dieses Verfahren der automatischen Zwischenspeicherung von Rechendaten macht es Ihnen leicht, auch bei der Lösung komplexester Rechenaufgaben die Übersicht zu wahren, erlaubt die Eingabe ohne komplizierte Klammerung und die Rechnung kann jederzeit überprüft werden, da alle Zwischenergebnisse angezeigt werden. Ganz gleich, wie komplex ein Problem ist, es kann stets in eine Folge von Funktionen zerlegt werden, die sich aus entweder einer oder zwei Zahlen beziehen. Arbeiten Sie sich auf genau die gleiche Weise durch das Problem, wie Sie es vom handschriftlichen Rechnen mit Bleistift her gewohnt sind. Wenn mehr als eine Zahl in den Rechner einzugeben ist, werden diese beiden Zahlen mit Hilfe der ENTER-Taste voneinander getrennt.

Geben wir als Beispiel vor, Sie wollen die 12 und die 3 addieren.

Drücken Sie die Zahlen 1 und 2 für 12, danach ENTER, und jetzt die 3 als zweite Zahl. Nun noch CURSOR links, und schon steht das Ergebnis 15 auf der Anzeige.

Die übrigen arithmetischen Operationen werden in gleicher Weise ausgeführt.

Nun noch einige weitere Beispiele zur Wirkung von Funktionen zweier Variablen:

Aufgabe 1: 3 hoch 5.

Tasten	Anzeige	
3		Basis
ENTER	3	
5		Exponent
CTRL F8	243	Ergebnis

Aufgabe 2: Sie wollen von 120 DM 14% berechnen!

Tasten	Anzeige	
120		
ENTER	120	
14		
SHIFT F7	16.8	

Das Ergebnis soll nun zu 120 addiert werden.

CURSOR LEFT	136.8	Ergebnis (120+16.8)

Aufgabe 3: Sie wollen den prozentualen Unterschied zwischen den Werten im X- und Y-Register wissen!

Tasten	Anzeige	
70		
ENTER	70	
240		
CTRL F7	242.86	Ergebnis.

Wie Sie an den vorstehenden Aufgaben gesehen haben, hilft die altbekannte Schreibweise bei der handschriftlichen Ausführung arithmetischer Grundrechnungen, das Programm zu verstehen. Stets sind zuvor beide Zahlen in das Rechenregister in der natürlichen Reihenfolge einzuordnen; dann wird die Operation ausgeführt, wenn Sie die entsprechende Funktionstaste drücken. Von dieser einfachen Regel gibt es keine Ausnahme. Als erstes sind als jeweils die Zahlen mit Hilfe von 'ENTER' in das Rechenregister einzuordnen. Das hat zur Folge, wie übrigens auch die Eintastung aller weiteren Zahlen, daß die Inhalte der Rechenregister um eine Position nach oben verschoben werden. Bei vielen ENTER-Befehlen nacheinander geht die erste eingeschriebene Zahl verloren.

Das Herzstück dieses Programmes ist die selbständige Verwaltung der vier Rechenregister, die nach jeder Rechenoperation automatisch verschoben werden. Die Zwei-Variablen-Funktionen werden immer mit dem X- und Y-Register ausgeführt. Das Ergebnis wird in das X-Register geschrieben und die übrigen Register um eine Position nach unten verschoben. Bei der Ein-Variablen- Funktion wird immer auf das X- Register zurückgegriffen, und auch das Ergebnis steht wieder im X-Register, eine Verschiebung findet nicht statt.

Zum Programm

selbst ist noch zu sagen, daß es in der abgedruckten Form direkt nur auf einem CPC 664 und 6128 lauffähig ist, Besitzer eines CPC 464 müssen zuvor den in diesem Heft befindlichen Emulator starten.

(Brigitte Kristen/jb)

für 6128

464 mit Emulator

```
10 '************************************    [1460]
11 '**                                  **    [104]
12 '**  T A S C H E N R E C H N E R     **   [1573]
13 '**                                  **    [104]
14 '************************************    [1460]
15 '**                                  **    [104]
16 '**                                  **    [104]
17 '**      von Brigitte Kristen        **   [1157]
18 '**                                  **    [104]
19 '**                                  **    [104]
20 '**            31.01.1988            **    [751]
21 '**                                  **    [104]
22 '**                                  **    [104]
23 '************************************    [1460]
24 :                                            [174]
25 :                                            [174]
100 '************************   TITELBILD [3017]
   ********************************
101 MODE 0:GRAPHICS PEN 4,1:TAG                 [2240]
102 WINDOW 2,19,3,3:PAPER 7:PEN 1:INK 12,2 [4219]
6:INK 13,0:INK 14,14:INK 15,14
103 WINDOW #1,2,19,2,4:PAPER #1,7:CLS#1:PE [2179]
N#1,13
104 WINDOW #2,2,9,7,24:PAPER #2,6:CLS#2:PE [2722]
N#2,10
105 WINDOW #3,11,19,7,24:PAPER #3,3:CLS#3: [2282]
PEN #3,4
106 MOVE 40,367:PRINT CHR$(164)"(1988) B.K [2237]
risten";
107 MOVE 32,327,7:PRINT"CPC 6128  Rechner" [2491]
;
```

Listing "Taschenrechner"

```
108 MOVE 350,275,10:PRINT"7  8  9";          [1748]
109 MOVE 350,220:PRINT "4  5  6";            [1292]
110 MOVE 350,165:PRINT "1  2  3";            [1004]
111 MOVE 350,110:PRINT "0  *  .";            [1036]
112 MOVE 350,55:PRINT "+  :  -";             [1201]
113 FOR i=1 TO 3                             [435]
114     READ x,y,r:PLOT x,y,r                [1543]
115     FOR s=1 TO 4                         [1015]
116         READ x,y:DRAWR x,y               [857]
117     NEXT                                 [350]
118 NEXT                                     [350]
119 FOR i=1 TO 4                             [450]
120     READ x,y,r:PLOT x,y,r                [1543]
121     READ x,y:DRAWR x,y                   [857]
122 NEXT                                     [350]
123 FOR i=1 TO 2                             [436]
124     PLOT 323+90*i,16:DRAWR 0,286         [1567]
125     PLOT 35+80*i,16:DRAWR 0,286          [1157]
126 NEXT                                     [350]
127 FOR i=1 TO 4                             [450]
128     PLOT 35,16+58*i:DRAWR 253,0:PLOTR    [3094]
32,0:DRAWR 282,0
129 NEXT                                     [350]
130 GRAPHICS PEN 10,1                        [1031]
131 MOVE 50,290,10:PRINT "o  2";             [1266]
132 MOVE 75,275:PRINT "o";:MOVE 130,278:PR   [2821]
INT "x";
133 PLOT 90,290:DRAW 60,260                  [743]
134 PLOT 206,280:DRAWR 8,0:DRAWR 12,-20:DR   [2513]
AWR 12,30:DRAWR 40,0:DRAWR 0,-4
135 MOVE 245,278:PRINT "x";                  [2526]
136 MOVE 38,243:PRINT "s  c  t";             [2418]
137 MOVE 55,230:PRINT "i  o  a";             [2508]
138 MOVE 66,217:PRINT "n  s  n";             [1482]
139 MOVE 56,182:PRINT "1 x  +";              [678]
140 MOVE 60,160:DRAW 95,160:MOVE 140,167:P   [5166]
RINT CHR$(255);:PLOT 262,174:DRAWR -32,-34
```

Listing "Taschenrechner"

```
141 MOVE 64,155:PRINT "x  y  =";                                [1441]
142 MOVE 38,124:PRINT "H";:MOVER -10,-16:P                      [7146]
RINT"M";:MOVER -10,-16:PRINT"S";:MOVER 12,
10,15:PRINT"MC";
143 MOVE 46,55,10:PRINT "M+";:MOVER 16,0,1                      [3651]
4:PRINT"MR";:MOVER 16,0,10:PRINT"M-";
144 MOVE 197,120,12:PRINT"RAD";:MOVE 196,9                      [4013]
5,13:PRINT"DEG";:RAD
145 TAGOFF                                                      [1066]
146 SYMBOL AFTER 32                                             [1296]
147 SYMBOL 48,126,66,66,66,66,66,126,0                          [1335]
148 SYMBOL 49,2,2,2,2,2,2,2,0                                   [1745]
149 SYMBOL 50,126,2,2,126,64,64,126,0                           [1974]
150 SYMBOL 51,126,2,2,126,2,2,126,0                             [2464]
151 SYMBOL 52,66,66,66,126,2,2,2,0                              [1935]
152 SYMBOL 53,126,64,64,126,2,2,126,0                           [2380]
153 SYMBOL 54,126,64,64,126,66,66,126,0                         [2428]
154 SYMBOL 55,126,2,2,2,2,2,2,0                                 [1628]
155 SYMBOL 56,126,66,66,126,66,66,126,0                         [2582]
156 SYMBOL 57,126,66,66,126,2,2,126,0                           [1833]
157 SYMBOL 129,32,98,36,40,21,34,69,0                           [1616]
158 SYMBOL 136,6,9,2,4,8,175,64,160                             [2064]
159 SYMBOL 137,192,64,143,201,8,85,34,37                        [2222]
160 SYMBOL 149,5,5,2,138,82,32,80,136                           [1939]
161 SYMBOL 150,224,160,175,9,208,85,34,37                       [1365]
162 KEY 139,CHR$(139)                                           [1454]
163 KEY 140,CHR$(140)                                           [1002]
164 KEY DEF 6,0,13,139,140                                      [1401]
165 FOR i=128 TO 138                                            [520]
166     READ taste                                              [445]
167     KEY i,CHR$(i):KEY i+13,CHR$(i+13)                       [2286]
168     KEY DEF taste,0,i-80,i,i+13                             [1585]
169 NEXT                                                        [350]
170 KEY DEF 7,0,46,138,151                                      [1645]
171 DATA 22,10,11,0,378,594,0,0,-378,-594,                      [7003]
0,35,16,8,0,286,253,0,0,-286,-253,0,323,16
,8,0,286, 282,0, 0,-286,-282,0
172 DATA 312,10,11,0,298,298,10,11,0,298,2                      [2483]
4,330,8,589,0,24,310,8,589,0
173 DATA 15,13,14,5,20,12,4,10,11,3,7                           [1865]
174 CLEAR                                                       [217]
175 GOTO 230                                                    [423]
176 '*****************************  Grafik E                    [3255]
NDE  *********************************
180 :                                                           [174]
190 :                                                           [174]
200 :                                                           [174]
210 '****************  T A S C H E N R E C                     [2031]
 H N E R  ****** Hauptteil **********
220 MODE 2:PEN 1:PAPER 0:END                                    [1928]
230 ON BREAK GOSUB 220                                          [675]
240 ON ERROR GOTO 2310                                          [1469]
250 sk=46:GOSUB 590                                             [958]
260 s=2:'                                                       [2496]
    Standard 2 Nachkommastellen
270 GRAPHICS PEN 10,0:GRAPHICS PAPER 3                          [2286]
280 INK 12,26:INK 13,0:INK 14,14:INK 15,14                      [837]
290 '****************** Eingabeschleife ***                    [2723]
************************************
300 WHILE q<>126                                                [1106]
301     q$="":WHILE q$="":q$=INKEY$:WEND:                       [2452]
IF op THEN op=0:CLS #1:CLS
305     l=INSTR(a$,"."):IF l>10 OR LEN(a$                       [5321]
)>18 OR LEN(a$)-l>10 THEN PRINT"";:
    a$=LEFT$(a$,10):GOTO 301
310     q=ASC(q$):IF q=13 THEN 1700                             [1563]
320     IF q>=240 THEN q=q-88 ELSE IF q<4                       [1552]
0 THEN 370
330     IF q>139 THEN 360 ELSE IF q>94 TH                       [1857]
EN 350
340     ON q-39 GOSUB 400,410,1520,1540,4                       [12438]
60,1550,460,1530,480,490,500,510,520,530,5
40,550,560,570,1530,590,600,2010,620,2010,
640,650,660,670,680,690,700,710,720,730,74
0,750,760,770,780,790,800,810,820,830,840,
850,860,870,880,890,900,910,1530,930,1490
345     GOTO 370                                                [466]
350     ON q-94 GOSUB 950,960,970,980,990                       [13058]
,1000,1010,1020,1030,1040,1050,1060,1070,1
080,1090,1100,1110,1120,1130,1140,1150,116
0,1170,1180,1190,1200,1210,1220,1230,1240,
1250,1260,1270,1280,1290,1300,1310,1320,13
30,1340,1350,1360,1370,1380,1390
355     GOTO 370                                                [466]
360     ON q-139 GOSUB 1400,1410,1420,143                       [6527]
0,1440,1450,1460,1470,1480,1490,1500,1510,
```

Listing "Taschenrechner"

```
1520,1530,1540,1550,1560,1570,1580,1590,16
00,1610,1620,1630
370 WEND                                                        [390]
380 CLS#1:CLS:END:RUN 230:'                                     [2030]
    restart mit cont
390 '************************************                      [2982]
*** Funktionsauswahl ****************
400 IF f=0 THEN GOSUB 1380:'                                    [2808]
    Kosinussatz  S-W-S
401 GOSUB 2430:b=z2:wa=z1:c=z                                   [2218]
403 zz=SQR(b^2+c^2-2*b*c*COS(wa)):GOSUB 20                      [3261]
10
404 CALL &BB06                                                  [393]
405 zz=b*c*SIN(wa)/2:GOTO 2010                                  [1592]
410 IF f=0 THEN GOSUB 1380:'                                    [2727]
    Kosinussatz  S-S-S
411 GOSUB 2430:a=z2:b=z1:c=z                                    [2237]
412 FOR i=1 TO 3                                                [435]
413     z=(b^2+c^2-a^2)/(2*b*c):GOSUB 1461                      [3367]
414     IF i=1 THEN d=b:b=a:a=d ELSE d=c:c                      [2582]
=a:a=d
415     CALL &BB06:GOSUB 1411:CALL &BB06                        [1524]
416 NEXT:GOTO 951                                               [1004]
460 a$=a$+".":PRINT ".";:RETURN:'                               [2553]
    Wertzuweisung
480 a$=a$+"0":PRINT "0";:RETURN                                 [1409]
490 a$=a$+"1":PRINT "1";:RETURN                                 [1138]
500 a$=a$+"2":PRINT "2";:RETURN                                 [2031]
510 a$=a$+"3":PRINT "3";:RETURN                                 [1663]
520 a$=a$+"4":PRINT "4";:RETURN                                 [615]
530 a$=a$+"5":PRINT "5";:RETURN                                 [1167]
540 a$=a$+"6":PRINT "6";:RETURN                                 [1121]
550 a$=a$+"7":PRINT "7";:RETURN                                 [1415]
560 a$=a$+"8":PRINT "8";:RETURN                                 [1733]
570 a$=a$+"9":PRINT "9";:RETURN                                 [1979]
590 IF sk=44 THEN sk1=46 ELSE sk1=44:'                          [4307]
    Schalter /  Dezimalpunkt -komma
591 SYMBOL sk1,0,0,0,0,0,24,24,48                               [2139]
592 SYMBOL sk,0,0,0,0,0,24,24,0                                 [1261]
593 TAG:MOVE 526,110,10:PRINT " ";:MOVER -                      [3374]
16,0:PRINT ".";:TAGOFF
594 IF sk=44 THEN sk=46 ELSE sk=44                              [933]
595 GOTO 2010                                                   [305]
600 enter=1:GOSUB 2430:zz=z^3*PI/6:GOTO 20                      [3391]
10:'   Rauminhalt Kugel
620 GOSUB 2430:zz=z:'                                           [1927]
    Fakultaet
621 IF zz>33 THEN op=1:CLS:PRINT " N! zu g                      [3572]
ross";:RETURN
622 FOR i=2 TO z:zz=zz*(i-1):NEXT                               [1257]
623 GOTO 2010                                                   [305]
640 op=1:PRINT#1,ROUND(z,2);ROUND(z1,2):'                       [2942]
      X- Y- Rechenregister
641 LOCATE #1,1,3:PRINT#1,ROUND(z2,2);ROUN                      [3499]
D(z3,2)
642 RETURN                                                      [555]
650 zz=mema:GOTO 2460:'                                         [3026]
      Konstantenspeicher Ausgabe
660 zz=memb:GOTO 2460                                           [1693]
670 zz=memc:GOTO 2460                                           [1464]
680 zz=memd:GOTO 2460                                           [1726]
690 zz=meme:GOTO 2460                                           [1988]
700 zz=memf:GOTO 2460                                           [2213]
710 zz=memg:GOTO 2460                                           [1590]
720 zz=memh:GOTO 2460                                           [1531]
730 zz=memi:GOTO 2460                                           [1935]
740 zz=memj:GOTO 2460                                           [1562]
750 zz=memk:GOTO 2460                                           [1800]
760 zz=meml:GOTO 2460                                           [1518]
770 zz=memm:GOTO 2460                                           [2037]
780 zz=memn:GOTO 2460                                           [1313]
790 zz=memo:GOTO 2460                                           [1963]
800 zz=memp:GOTO 2460                                           [1911]
810 zz=memq:GOTO 2460                                           [2268]
820 zz=memr:GOTO 2460                                           [1340]
830 zz=mems:GOTO 2460                                           [2264]
840 zz=memt:GOTO 2460                                           [1967]
850 zz=memu:GOTO 2460                                           [2788]
860 zz=memv:GOTO 2460                                           [1860]
870 zz=memw:GOTO 2460                                           [1942]
880 zz=memx:GOTO 2460                                           [2278]
890 zz=memy:GOTO 2460                                           [1453]
900 zz=memz:GOTO 2460                                           [1915]
910 s$="":WHILE s$="":s$=INKEY$:WEND:s=VAL                      [4441]
(s$):GOTO 2010:' Nachkommastellen
```

Listing "Taschenrechner"


```
930 IF ag=1 THEN ag=0 ELSE ag=1:'                               [3023]
    Schalter fuer Ausgabeformat
933 GOTO 2010                                                   [305]
950 GOSUB 2430:a=z:b=z1:c=z2:'                                  [2468]
    Heron
951 zz=(a+b+c)/2                                                [1101]
952 zz=SQR(zz*(zz-a)*(zz-b)*(zz-c)):GOTO 2                      [4310]
010
960  IF f=0 THEN GOSUB 1380:'                                   [2727]
    Sinussatz   S-S-W
961 GOSUB 2430:a=z2:b=z1:wa=z                                   [2379]
963 z=b/a*SIN(wa):GOSUB 1451                                    [1965]
964 CALL &BB06:wb=zz                                            [1158]
965 zz=a*COS(wb)+b*COS(wa):GOSUB 2010:CALL                      [4239]
 &BB06
966 IF a>b THEN 951 ELSE IF qq THEN qq=0:G                      [3712]
OTO 951 ELSE GOSUB 951
967 CALL &BB06:CLS:PRINT" 2. Loesung                            [882]
968 wb=180-wb:qq=1:CALL &BB06:GOTO 965                          [2963]
970 GOSUB 2430:mema=z:GOTO 2010:'                               [2411]
    Konstantenspeicher Eingabe
980 GOSUB 2430:memb=z:GOTO 2010                                 [1827]
990 GOSUB 2430:memc=z:GOTO 2010                                 [1440]
1000 GOSUB 2430:memd=z:GOTO 2010                                [2239]
1010 GOSUB 2430:meme=z:GOTO 2010                                [2196]
1020 GOSUB 2430:memf=z:GOTO 2010                                [2139]
1030 GOSUB 2430:memg=z:GOTO 2010                                [2153]
1040 GOSUB 2430:memh=z:GOTO 2010                                [2072]
1050 GOSUB 2430:memi=z:GOTO 2010                                [1584]
1060 GOSUB 2430:memj=z:GOTO 2010                                [1513]
1070 GOSUB 2430:memk=z:GOTO 2010                                [1874]
1080 GOSUB 2430:meml=z:GOTO 2010                                [2170]
1090 GOSUB 2430:memm=z:GOTO 2010                                [1611]
1100 GOSUB 2430:memn=z:GOTO 2010                                [1546]
1110 GOSUB 2430:memo=z:GOTO 2010                                [1394]
1120 GOSUB 2430:memp=z:GOTO 2010                                [1626]
1130 GOSUB 2430:memq=z:GOTO 2010                                [1443]
1140 GOSUB 2430:memr=z:GOTO 2010                                [1963]
1150 GOSUB 2430:mems=z:GOTO 2010                                [1312]
1160 GOSUB 2430:memt=z:GOTO 2010                                [1848]
1170 GOSUB 2430:memu=z:GOTO 2010                                [1489]
1180 GOSUB 2430:memv=z:GOTO 2010                                [1098]
1190 GOSUB 2430:memw=z:GOTO 2010                                [1467]
1200 GOSUB 2430:memx=z:GOTO 2010                                [2084]
1210 GOSUB 2430:memy=z:GOTO 2010                                [2019]
1220 GOSUB 2430:memz=z:GOTO 2010                                [1640]
1230 IF f=0 THEN GOSUB 1380:'                                   [2872]
    Sinussatz   S-W-W
1231 GOSUB 2430:b=z2:wa=z1:wb=z                                 [2633]
1232 wc=180-wa-wb                                               [1360]
1233 FOR i=1 TO 2                                               [436]
1234 zz=SIN(wa)/SIN(wb)*b:GOSUB 2010                            [3624]
1235 CALL &BB06:IF wa<>wc THEN wa=wc:a=zz                       [2673]
ELSE c=zz
1236 NEXT:GOTO 951                                              [1004]
1240 CALL 0:' ***************                                   [1787]
    ENDE Ruecksetzung
1250 IF f=0 THEN GOSUB 1380:'                                   [2192]
    Sinussatz   W-S-W
1251 GOSUB 2430:wa=z2:b=z1:wb=z                                 [1709]
1252 wh=180-wa-wb:wc=wb:wb=wh:GOTO 1233                         [2816]
1260 RETURN:'                                                   [871]
    ENDE CTRL 2
1270 IF LEN(a$) THEN a$="" ELSE z=z1:z1=z2                      [2804]
:z2=z3:'          CLx
1271 CLS:PRINT 0:RETURN                                         [767]
1280 GOSUB 2430:grad=FIX(z):'                                   [1933]
    sedezi in dezi
1281 minuten=ROUND((z-grad)*100,0)                              [2096]
1282 sekunde=(minuten*60+(z-grad-minuten/1                      [4319]
00)*10000)/3600
1284 zz=grad+sekunde:GOTO 2010                                  [2360]
1290 GOSUB 2430:zz=1/z:fu$=CHR$(129):GOTO                       [3053]
2010:'1/x
1300 zz=z1:z1=z:GOTO 2010:'                                     [1883]
    x gegen y tauschen
1310 CLS:IF LEN(a$) THEN a$=STR$(VAL(a$)*-                      [3836]
1):PRINT a$ ELSE z=z*-1:PRINT z
1311 RETURN                                                     [555]
1320 GOSUB 2430:zz=SIN(z):GOTO 2010:'                           [3914]
    Sinus eines Winkels
1330 GOSUB 2430:zz=COS(z):GOTO 2010:'                           [2737]
    Kosinus e.  "
1340 GOSUB 2430:zz=TAN(z):GOTO 2010:'                           [3887]
    Tangens "   "
1350 GOSUB 2430:zz=z1*z/100:fu$="%":GOTO 2                      [2854]
010:'x Prozent y
```

Listing "Taschenrechner"

```
1360 GOSUB 2430:zz=z*z:fu$=CHR$(136):GOTO                       [3715]
2010:'Potenzieren
1370 GOSUB 2430:zz=SQR(z):fu$=CHR$(137):GO                      [4219]
TO 2010:'Radizieren
1380 IF f=26 THEN f1=0 ELSE f1=26:'                             [3181]
    Schalter Bogenmass/Winkelmass
1381 IF f THEN RAD ELSE DEG                                     [1223]
1385 INK 12,f:INK 13,f1                                         [1144]
1389 h=f:f=f1:f1=h:GOTO 2010                                    [699]
1390 PRINT #8,z:GOTO 2020:'                                     [2842]
    Ergebnis drucken
1400 IF prt=8 THEN prt=0 ELSE prt=8:'                           [2952]
    alles drucken
1401 CLS:PRINT zz:RETURN                                        [461]
1410 GOSUB 2430:'                                               [2158]
    dezi in sedizimal Grad,MinSek
1411 grad=FIX(z):minuten=FIX((z-grad)*60)                       [2758]
1412 sekunde=((z-grad)*60-minuten)*60                           [1203]
1414 zz=grad+minuten/100+sekunde/10^4                           [2022]
1415 GOTO 2010                                                  [305]
1420 z3=0:z2=0:z1=0:z=0:zz=0:GOTO 2010:'                        [2962]
    Register loeschen
1430 zz=z1:z1=z2:z2=z3:z3=z:GOTO 2010 :'                        [3552]
    Register verschieben
1440 RUN 230                                :'                  [2343]
    alle Speicher loeschen
1450 GOSUB 2430:'                                               [1366]
    Umkehr Sinus
1451 zz=ATN(z/SQR(1-z^2))                                       [2077]
1452 GOTO 2010                                                  [305]
1460 GOSUB 2430:'                                               [1940]
    Umkehr Kosinus
1461 zz=ATN(SQR(1-z^2)/z):IF zz<0 THEN zz=                      [3232]
180+zz
1462 GOTO 2010                                                  [305]
1470 GOSUB 2430:'                                               [1440]
    Umkehr Tangens
1471 zz=ATN(z):IF zz<0 THEN zz=180+zz                           [2471]
1472 GOTO 2010                                                  [305]
1480 GOSUB 2430:zz=100/z1*(z-z1):fu$="%":G                      [4537]
OTO 2010:'Prozentualer Unterschied
1490 p=1:GOSUB 2430:zz=z1^z:fu$=CHR$(149):                      [3341]
GOTO 2010:'x hoch y
1500 p=1:GOSUB 2430:zz=z1^(1/z):fu$=CHR$(1                      [4439]
50):GOTO 2010:'n.te Wurzel aus x
1510 a$=STR$(PI):GOSUB 2430:zz=z:fu$=CHR$(                      [4430]
184):GOTO 2010:'Kreiszahl pi
1520 p=1:GOSUB 2430:zz=z1*z:fu$="*":GOTO 2                      [2972]
010:'Multiplizieren
1530 p=1:GOSUB 2430:zz=z1/z:fu$=":":GOTO 2                      [3315]
010:'Dividieren
1540 p=1:GOSUB 2430:zz=z1+z:fu$="+":GOTO 2                      [2899]
010:'Addieren
1550 p=1:GOSUB 2430:zz=z1-z:fu$="-":GOTO 2                      [3247]
010:'Subtrahieren
1560 m=0:INK 14,14:GOTO 2010:'                                  [2983]
    1.Arithmetik-Speicher loeschen
1570 zz=m:GOTO 2460:'                                           [3358]
    1.Arithmetik-Speicher ausgeben
1580 fm=14:GOSUB 2450:m=m+z:GOTO 2010:'                         [2674]
    1.Arithmetik-Speicher addieren
1590 fm=14:GOSUB 2450:m=m-z:GOTO 2010:'                         [2809]
    1.Arithmetik-Speicher subtra.
1600 m1=0:INK 15,14:GOTO 2010:'                                 [3906]
    2.Arithmetik-Speicher loeschen
1610 zz=m1:GOTO 2460:'                                          [4586]
    2.Arithmetik-Speicher ausgeben
1620 fm=15:GOSUB 2450:m1=m1+z:GOTO 2010:'                       [3236]
    2.Arithmetik-Speicher addieren
1630 fm=15:GOSUB 2450:m1=m1-z:GOTO 2010:'                       [3559]
    2.Arithmetik-Speicher subtra.
1700 IF enter=0 THEN z3=z2:z2=z1:'                              [1548]
    Eingabe durch Enter trennen
1701 z1=z:enter=1                                               [1062]
1710 IF ABS(VAL(a$))>0 THEN zz=VAL(a$):a$=                      [1113]
""
1720 GOSUB 2010:z3=z2:z2=z1:z1=z:op=1:'                         [2883]
    Rechenregister
1730 GOTO 370                                                   [466]
1740 '**************** ENDE Funktionsauswah                     [2906]
l ********
1750 :                                                          [174]
1990 :                                                          [174]
2000 '***************** Ausgabe ***********                     [1577]
**********
2010 op=1:CLS#1:CLS                                             [704]
```

Listing "Taschenrechner"

```
2020 IF ag=0 THEN 2090:'                                    [2502]
 Ausgabeformat
2030 IF LEN(fu$)>0 AND p=1 THEN PRINT #pr, [12205]
ROUND(z1,s);fu$;ROUND(z,s)"="ROUND(zz,s);
ELSE IF LEN(fu$)>0 AND p=0 THEN PRINT #pr,
ROUND(z,s)fu$;ROUND(zz,s); ELSE PRINT #pr,
ROUND(zz,s);
2040 IF prt=8 AND pr=0 AND enter=0 THEN pr [2519]
=8:GOTO 2030
2050 IF prt=8 AND pr=8 THEN pr=0:PRINT#8    [2317]
2060 IF p THEN z1=z2:z2=z3:p=0                [1903]
2070 fu$="":z=zz                              [97]
2080 RETURN                                   [555]
2090 w$=STR$(ROUND(zz,s)):'                   [2943]
 vergroesserte Ausgabeform
2100 CLEAR INPUT                              [184]
2110 x=40:y=376:xx=x:br=4:hh=5.2              [3087]
2120 FOR a=1 TO LEN(w$)                       [1016]
2130      as=ASC(MID$ (w$,a,1)):adr=HIMEM+1 [1881]
+(as-32)*8
2140      IF MID$ (w$,a,1)="E" THEN x=400:x [2257]
x=x:GOTO 2250
2150      FOR b=adr TO adr+7                  [1693]
2160           ad$ =BIN$(PEEK(b),8)           [1049]
2170           FOR c=1 TO 8                   [571]
2180                IF MID$(ad$,c,1)<>"1" THE [1770]
N 2200
2190                     PLOT x,y,5:DRAW x+br,y [1938]
:DRAW x+br,y-hh:DRAW x,y-hh:DRAW x,y
2200                x=x+br                    [1278]
2210           NEXT                           [350]
2220           y=y-hh:x=xx                    [706]
2230      NEXT                                [350]
2240      x=x+br*8:xx=x:y=376                 [1845]
```

Listing "Taschenrechner"

```
2250 NEXT                                     [350]
2260 GOTO 2040                                [355]
2270 '******************** E N D E Ausgabe [2246]
 ****************
2280 :                                        [174]
2290 :                                        [174]
2300 '******************** F E H L E R - b [2425]
ehandlung    ****************
2310 IF ERR <=14 THEN op=1:e=ERR:ON e GOTO [6805]
 2320,2320,2320,2320,2330,2340,2350,2320,2
320,2320,2360,2320,2370,2350 ELSE 220
2320 GOTO 300                                 [504]
2330 CLS#1:CLS:PRINT" F E H L E R ":RESUME [2174]
 300
2340 CLS#1:CLS:PRINT" WERT > 1.7 E+-38":RE [2071]
SUME 300
2350 CLS#1:CLS:PRINT" SPEICHER ist voll":R [2857]
ESUME 300
2360 CLS#1:CLS:PRINT" DIVISION / 0 !":RESU [3007]
ME 300
2370 a$="":zz=0:GOSUB 2010:RESUME 300       [2798]
2380 :                                        [174]
2390 :                                        [174]
2400 :                                        [174]
2410 :                                        [174]
2420 '******************** sub routin **** [2061]
****************
2430 IF ABS(VAL(a$))>0 AND enter=0 THEN z3 [4169]
=z2:z2=z1:z1=z:z=VAL(a$) ELSE IF ABS(VAL(a
$))>0 AND enter THEN z=VAL(a$):enter=0
2440 zz=z:a$="":RETURN                        [1190]
2450 GOSUB 2430:INK fm,26:RETURN              [1718]
2460 z3=z2:z2=z1:z1=z:z=zz                    [1896]
2470 GOTO 2010                                [305]
```

Listing "Taschenrechner"

Mehr Farbe ins Bild

SCREEN-PACK-PLUS – das Superprogramm

Screen Pack Plus bietet Ihnen gleich zwei nützliche Eigenschaften an: Sie können erstens doppelt (weniger einer Farbe) so viele Farben wie üblich in jedem Mode darstellen, und zweitens haben Sie alternativ die Möglichkeit, zwei Auflösungen gleichzeitig auf dem gesamten Bildschirm darzustellen.

Das besondere hierbei ist, daß es keine Einschränkungen hinsichtlich der Auflösung gibt und die Eigenschaften nicht auf bestimmte Bildabschnitte beschränkt sind. Mit anderen Worten: Diese Leistungsmerkmale gelten für jeden einzelnen der max. 128.000 Pixel.

Was macht das Programm?

Es drängt sich natürlich die Frage auf, wie dieses realisiert werden kann, da bisher eine Steigerung der Farbfähigkeit oder eine Mischbarkeit von zwei oder mehreren Modes auf bestimmte Bildabschnitte beschränkt war. So konnte z.B. in Teil 3 der Serie Profi RSX (vergl. PC Schneider Int. 8/87) bis zu achtmal der Mode gewechselt werden. Auch waren bis zu 16 Farben in Mode 2 realisierbar. Das Ganze hatte jedoch einen Nachteil. Die Bildschirmgrafik war nur zeilenweise zu beeinflußen. Es konnten also nur nach mindestens drei Zeilen der Mode oder die Farben gewechselt werden. An einen Wechsel des Modes innerhalb einer Zeile war nicht zu denken. Erfreulicherweise ist Letzteres mit Screen-Pack-Plus jetzt möglich. Auch wenn nur zwei Modes zur gleichen Zeit verfügbar sind, dürfte dieses einen nicht unwesentlichen Fortschritt in den Grafikfähigkeiten Ihres CPCs bedeuten. Bevor wir zu den programmtechnischen Hintergründen kommen, hier kurz einen Überblick über die

Leistungsmöglichkeiten:

Version 1: Verdoppelung (-1) der max. Farbenzahl auf:
Mode 0: 31 Farben
Mode 1: 07 Farben
Mode 2: 03 Farben
Wobei die Zahlen für die Anzahl der verwendbaren Farbstifte gelten. Diesen können leider nur max. 27 Farben zugewiesen werden, da der CPC nicht mehr verschiedene Farben erzeugen kann. Die restlichen vier Stifte in Mode 0 sind somit in ihrer Verwendbarkeit leicht eingeschränkt. Sie lassen sich jedoch recht gut für zusätzliche Blink-Aktivitäten nutzen, ohne daß andere Stifte auch ins Blinken geraten.
Version 2: 2 Modes gleichzeitig auf dem gesamten Bildschirm.
Bei beiden Versionen ist zu beachten, daß der Hintergrund (das 'Papier') aus programmtechnischen Gründen immer die Farbe schwarz (Nr. 0) haben muß.

Funktionsweise

Kommen wir jetzt zu der Eingangs erwähnten Frage zurück, wie dieses eigentlich funktioniert und realisiert wurde. Wie Sie sicherlich wissen, handelt es sich bei den zum CPC serienmäßig mitgelieferten Monitoren (gleich ob grün oder farbig) um 50 Hz-Versionen. Diese erzeugen entsprechend viele Einzelbilder pro Sekunde. Screen Pack Plus greift hier ein und wendet einen kleinen Trick an: Es halbiert die Bildwiederholfrequenz. Folglich ist der Monitor dann nur zur Hälfte ausgelastet. Die freigewordene Zeit wird durch Screen Pack Plus dazu verwendet, ein zweites Bild einzublenden. Um das Ganze nun 'monitorgerecht' zu gestalten, wird immer abwechselnd ein Bild auf den Monitor geschrieben. Um Ihnen die Wirkungsweise zu verdeutlichen, bietet es sich an, ein kleinen Vergleich als Beispiel heranzuziehen:

Nehmen wir einmal an, Sie hätten zwei Transparentfolien. Auf der einen haben Sie eine farbige Grafik gezeichnet, auf der anderen dagegen haben Sie passend hierzu einige Texte abgebildet. Legen Sie diese nun übereinander, so sehen Sie beides gleichzeitig: Eine Grafik mit einigen Texten. Als Ergebnis haben Sie statt zwei Einzelbildern ein einziges Bild. Gleiches gilt für Screen Pack Plus, nur daß hier statt zwei Folien zwei Bilder eingeblendet werden. Da dieses bei jedem Bildaufbau immer abwechselnd geschieht, ergibt sich eine transparente Darstellung, die zwei Bilder erscheinen, als ob sie eins wären. Bedauerlich hat (fast) alles, neben vielen Vorteilen, auch oft mehr oder weniger gravierende Nachteile. Bei Screen Pack Plus gibt es in dieser Hinsicht auch einen kleinen 'Schönheitsfehler', der sich so bemerkbar macht, daß das Monitorbild instabiler wird, d.h., das es leicht anfängt zu flimmern. Dieses ist auch ganz logisch durch die halbierte Bildwiederholfrequenz begründbar. Allerdings ist es bei geeigneter Farbwahl und richtig eingestellter Helligkeit (bzw. Kontrast beim Grün-Monitor) noch erträglich, zumal es zum Beispiel die Firma Commodore bei Ihrem Model 'Amiga' allen Käufern auch zumutet. Wird dieser bei höchstmöglicher Auflösung (640*400 [512] Pixel) betrieben, so erreicht er diese Auflösung durch die sogenannte 'Interlace'-Technik. Bei dieser, auch Zeilensprung-Verfahren genannten Darstellungsmethode wird die Bildwiederholfrequenz ebenfalls praktisch halbiert. Somit befindet sich Ihr CPC in 'bester' Gesellschaft, da es optisch bei beiden Rechnern zu den gleichen Auswirkungen (sprich Flimmern) kommt.

Auswirkungen

auf die Programmierung in BASIC/ Assembler gibt es bei dem Programm so gut wie gar nicht. Sie können wie bisher weiter Ihre Programme nach gewohntem Muster erstellen. Allerdings ist hier zwischen den beiden zur Verfügung gestellten Varianten zu differenzieren. Bei der Version 1 ist nur zu bedenken, daß Sie mit entsprechend mehr Farben (s. Abbildung 3) arbeiten können. Hinsichtlich der Erzeugung von Hardcopys ist der gleiche Weg wie bei Version 2 zu beschreiten. Bei der zweiten Version ist in Betracht zu ziehen, daß Sie mit zwei Bildschirmen direkt arbeiten müssen. Zu Beginn sollten Sie über das Kommando 'BV.ON' die beiden gewünschten Auflösungen festlegen. Dann ist es ab sofort möglich, Grafik und Texte in einen der beiden Modes auf dem Bildschirm darzustellen. Hierzu Bedarf es lediglich der Anwahl des entsprechenden Bildschirmes durch den Befehl 'Bildschirm'. Screen Pack Plus verwaltet immer zwei Bildspeicher, wobei der eine an der Standardadresse (&C000-&FFFF), der andere jedoch im Bereich von &4000 bis &7fff liegt. Dadurch ergibt sich, daß vor einem Aufruf von Screen Pack Plus HIMEM auf einen Wert von max. &3fff gesetzt werden muß (mittels MEMORY &3fff). Screen Pack Plus selbst liegt ab Adresse &8000 und beansprucht einen Bereich bis einschl. &91ff. Sie sehen, Screen Pack Plus hat eine nicht unbedeutende Länge. Dieses rührt nicht daher, daß das gewählte Verfahren an sich so kompliziert ist, sondern das sehr viele Betriebssystem-Routinen abgeändert werden mußten, um eine (fast) vollständige Kompatibilität zu diesen zu erreichen. Welche Routinen und Steuerzeichen entsprechend korrigiert wurden, zeigen die Abbildungen 1 und 2.

Alle weiteren Informationen entnehmen Sie bitte aus der anschließend abgedruckten Befehlsliste.

Praxis-Einsatz

Um Ihnen die bis hier vorgestellten neuen Einsatz-Möglichkeiten auch praktisch zu demonstrieren, habe ich ein kleines Demo-Programm erstellt (Listing 4), welches Ansätze für mögliche Anwen-

Die folgenden Routinen wurden bei SCREEN PACK PLUS (Version 1) geändert:

Name der Routine	Adresse (RAM)
SCR MODE CLEAR (Indirection)	&BDEB
SCR WRITE (Ind.)	&BDE8
SCR READ (Ind.)	&BDE5
SCR SET MODE	&BC0E
SCR SET INK	&BC32
SCR FILL BOX	&BC44
SCR FLOOD BOX	&BC47
SCR HW ROLL	&BC4D
SCR SW ROLL	&BC50
TXT DRAW CURSOR (Ind.)	&BDCD
TXT UNDRAW CURSOR (Ind.)	&BDD0
TXT OUT ACTION (Ind.)	&BDD9
TXT WR CHAR	&BB5D
TXT CLEAR WINDOW	&BB6C
TXT PLACE CURSOR	&BB8A
TXT REMOVE CURSOR	&BB8D
TXT SET PEN	&BB90
TXT SET BACK	&BB9F
TXT GET BACK	&BBA2
GRA CLEAR WINDOW	&BBDB
GRA SET PEN	&BBDE

sowie folgende Steuerzeichen:

CHR$(0)	CHR$(15)
CHR$(1)	CHR$(16)
CHR$(4)	CHR$(17)
CHR$(8)	CHR$(18)
CHR$(9)	CHR$(19)
CHR$(10)	CHR$(20)
CHR$(11)	CHR$(22)
CHR$(12)	CHR$(28)
CHR$(13)	

Abb. 1: Betriebssystem- Änderungen bei Version 1

Die folgenden Routinen wurden bei SCREEN PACK PLUS (Version 2) geändert:

Name der Routine	Adresse (RAM)
SCR WRITE (Indirection)	&BDE8
SCR SET MODE	&BC0E
SCR SET INK	&BC32
SCR FILL BOX	&BC44
SCR FLOOD BOX	&BC47
SCR HW ROLL	&BC4D
SCR SW ROLL	&BC50
TXT DRAW CURSOR (Ind.)	&BDCD
TXT UNDRAW CURSOR (Ind.)	&BDD0
TXT OUT ACTION (Ind.)	&BDD9
TXT WR CHAR	&BB5D
TXT CLEAR WINDOW	&BB6C
TXT PLACE CURSOR	&BB8A
TXT REMOVE CURSOR	&BB8D
TXT SET BACK	&BB9F
TXT GET BACK	&BBA2
GRA CLEAR WINDOW	&BBDB

sowie folgende Steuerzeichen:

CHR$(0)	CHR$(13)
CHR$(1)	CHR$(16)
CHR$(4)	CHR$(17)
CHR$(8)	CHR$(18)
CHR$(9)	CHR$(19)
CHR$(10)	CHR$(20)
CHR$(11)	CHR$(22)
CHR$(12)	CHR$(28)

Abb. 2: Betriebssystem- Änderungen bei Version 2

Bei SCREEN PACK PLUS (V1) sind folgende Farbenzahlen/ Mode möglich:

MODE	max. Farbenzahl
0	31
1	7
2	3

(Die Zahlen verstehen sich ohne Rand, einschließlich Hintergrund)

Abb. 3: Maximale Farbmöglichkeiten bei Version 1

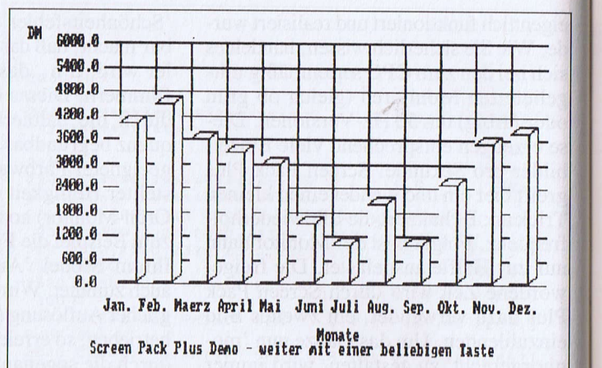


Abb. 4: Hardcopy des Mode 2 - Bildes aus dem Demo-Programm zur Version 2

dungen aufzeigt. Sie finden es ebenfalls im Anschluß abgedruckt. Insbesondere soll es Ihnen auch als Beispiel zum Einsatz der Erweiterung in eigenen Programmendienen.

Betrachten wir kurz das Listing. In der Zeile 150 können Sie zwischen zwei Demonstrationen wählen. Entscheiden Sie sich für Version 1, werden die Zeilen 200 bis 400 durchlaufen. Dabei werden die erhöhten Farbzahlen in allen Modes gezeigt. Durch die kombinierte Text- und Grafik-Darstellung wird augenscheinlich, daß die Erhöhung der Anzahl der Farben wirklich für alle Pixel gilt. Auch die Ansteuerung mittels der normalen und der neuen 'PEN' und 'INK'-Befehle ist ersichtlich. Zur komfortableren Farbwahl bei Grafiken haben Sie zusätzlich noch den Befehl 'GPEN' zur Verfügung.

Ab Zeile 440 finden Sie die Demo- Routine für die Version 2. Da sich hier weitaus mehr Möglichkeiten ergeben, ist sie entsprechend umfangreicher geworden. Ihre Aufgabe liegt darin, ein Säulendiagramm nach vorgegebenen Werten zu erstellen. Die Werte lassen sich entweder über die Input-Routinen in den Zeilen 520 bis 680 einlesen, oder es können die in den Zeilen 720 bis 780 enthaltenen Daten übernommen werden. Ist diese notwendige Abfrage erledigt, wird das Säulendiagramm ab der Programmzeile 790 erstellt. Dieses geschieht prinzipiell so, daß auf dem Bildschirm 1 die Säulen in Mode 0 erstellt werden, damit für jede Säule eine andere Farbe verwendet werden kann. In Mode 2 (in Bildschirm 2) wird das nötige Raster sowie die Beschriftung erstellt, da nur bei 80 Zeichen pro Zeile die Übersichtlichkeit gewährleistet ist. Das Zeichnen der Säulen (der Vorderfläche) geschieht in den Zeilen 1050 bis 1100. Um diesen Säulen einen dreidimensionalen Effekt zu geben, wird in den Zeilen 1120-1170 die nötige Quaderform erzeugt.

Nach Fertigstellung des Diagramms sehen Sie auf Ihren Monitor beide Auflösungen gleichzeitig. Jedoch befinden sich beide Darstellungen, wie schon angedeutet, in zwei verschiedenen Speicherbereichen. Um Ihnen dieses vor Augen zu führen, sollten Sie sich die Abbildungen 4 bis 6 ansehen. In Abbildung 4 sehen Sie als Hardcopy den Inhalt des 80 Zeichen Video-RAMs. Äquivalent zeigt hierzu Abbildung 5 die Mode 0 Grafik. Bei letzterer Hardcopy steht jede Rasterung für eine andere Farbe. Wenn nun beide Bildschirme übereinander gelegt werden (wie durch den Monitor), ergibt sich Abbildung 6. Sie zeigt, daß beide Diagramme zusammen eine absolut passende und von Auflösung und Farbigkeit wahrscheinlich nur mit Screen Pack Plus realisierbare Grafik ergeben.

Hardcopys anfertigen

Solche Hardcopys können Sie auch erstellen. Dazu benötigen Sie nur ein Hardcopy-Programm, wobei die hier abgedruckten Exemplare mit dem Programm 'Copyshop' erstellt worden sind. Dieses geschah wie folgt: Als erstes wurde die Grafiken erstellt und beide Grafiken auf Diskette abgespeichert. Dieses geschah mit nachstehender Befehlsfolge:

Bildschirm1 mit:

SAVE "Bild1",b,&C000,&4000

Bildschirm 2 mit:

SAVE "Bild2",b,&4000,&4000

Die so erzeugten Binär-Dateien konnten dann problemlos mit Copyshop weiterverarbeitet werden. Soll nur ein Bildschirm als Hardcopy zu Papier gebracht werden, so ist das Verfahren Copyshop üblich. Sollen zwei Bildschirme übereinander gedruckt werden, ist zu beachten, daß Sie die Papier-Position im Drucker markieren. Danach drucken Sie eine Hardcopy und bringen das Papier durch Zurückdrehen wieder in seine Ausgangsposition. Anschließend drucken Sie den zweiten Bildschirm aus. Falls Sie eine im Speicher resistente Routine besitzen, so ist diese ebenfalls verwendbar. Der Ausdruck muß dann nur zweimal erfolgen, wobei mittels des Befehles 'BILDSCHIRM' dann immer der jeweilige Bildschirm einzuschalten ist. Wichtig ist daß das Papier nach jeder ersten Hardcopy exakt auf seine Anfangsposition zurückgedreht wird.

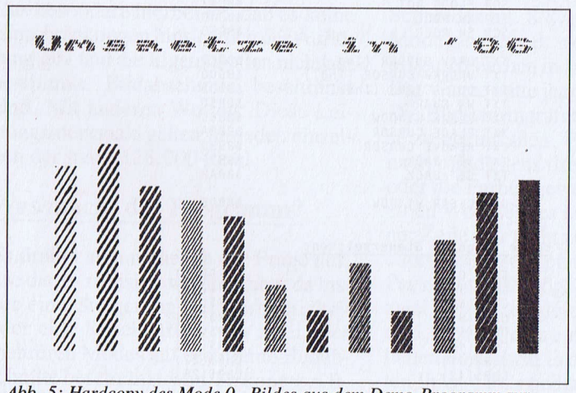


Abb. 5: Hardcopy des Mode 0 - Bildes aus dem Demo-Programm zur Version 2

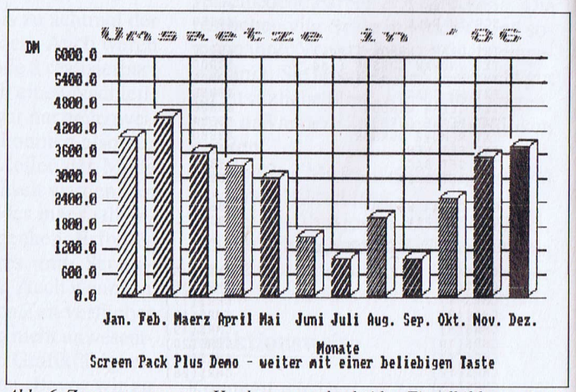


Abb. 6: Zusammengesetzte Hardcopy aus den beiden Einzelbildern, wie sie auch auf dem Monitor erscheint

Wenn Sie jetzt noch eine für sich sinnvolle und nützliche Anwendungsmöglichkeit von Screen Pack Plus finden, dürfte sich das Abtippen (oder die Bestellung der DATABOX) des zugegebenermaßen recht langen BASIC-Loaders lohnen. Nachdem dies erfolgt ist, sollten Sie zuerst das BASIC-Listing auf Diskette oder Kassette sichern. Danach setzen Sie den BASIC-Speicher mit

'MEMORY &7fff'

herab und starten den Loader mit 'RUN'. Sie können dabei das neue Binär-File automatisch abspeichern lassen. Nach Durchlauf des BASIC-Loaders löschen Sie diesen mit

'NEW'

und setzen den BASIC-Speicher durch

'MEMORY &3fff'

weiter herab. Die Initialisierung erfolgt jetzt mit 'CALL &8000'.

Ab jetzt kann Screen Pack Plus auch in dieser Reihenfolge geladen werden:

'MEMORY&3fff:LOAD"spp-464.bin":CALL &8000'

(Die Ziffernfolge "464" ist entsprechend Ihres Rechnertyps anzupassen).

Das Demo-Programm ist nach der Initialisierung von Screen Pack Plus ohne besondere Vorkehrungen ablaufbar.

Die Listings

Zu den Listings ist noch Folgendes zu sagen:

Listing 1 ist die Urversion,und sofort auf einem CPC 464 lauffähig. CPC 664-Besitzer müssen die Zeilen, die in Listing 2 stehen, in das Listing 1 einbauen, CPC 6128-Besitzer entsprechend die Zeilen des Listings 3. Das Demo-Programm fragt nach dem gewünschten Rechner ab, ist also für alle drei Rechner zugelassen.

Befehlsliste Screen Pack Plus

Alle Befehle von Screen Pack Plus sind als RSX eingebunden. Somit muß allen Befehlen ein senkrechter Strich vorangestellt werden. Screen Pack Plus ersetzt die interne (standardmäßige) Bildschirm-Verwaltung durch neue Routinen, sofern es nach der Initialisierung (durch 'CALL &8000') durch den Befehl 'BV.ON' eingeschaltet wurde. Ist dieses nicht der Fall, so beeinflußt es das System in keinster Weise, gleiches gilt für den Zustand nach dem Desaktivieren (durch 'BV.OFF'). Zu beachten ist, daß vor der Initalisierung der BASIC-Speicher durch 'Memory &3fff' begrenzt werden muß.

Zusätzliche Maschinensprache-Programme können erst ab &9200 abgelegt werden (bis zur Speicherobergrenze).

Unabhängig von der gewählten Version wird der Bildschirmspeicher verdoppelt und auch die Darstellung seines Inhaltes auf dem Monitor durch Screen Pack Plus besorgt. Dabei werden die beiden RAM-Bänke immer abwechselnd auf dem Monitor dargestellt, die Bildwiederholfrequenz wird pro 'Bild' auf die Hälfte herabgesetzt, damit Zeit für die Darstellung von 'zwei' Bildern frei wird. Die zusammengesetzte Bildfrequenz entspricht der Standardmäßigen (meistens 50 Hz). Der Anwender merkt hiervon nichts, da die zwei Video-RAMs an der Schnittstelle Benutzer/ Screen Pack Plus bzw. Betriebssystem als ein einziger Bildschirm angesehen werden. Dieses wird durch gleichmäßiges Scrolling beider Bildspeicher und dem Beschreiben beider RAMs mit den gewünschten Zeichen oder Pixels erreicht. Es werden alle Betriebssystem-Routinen entsprechend abgeändert, so daß eine nahezu volle Kompatibilität zum Standard-Betriebssystem erreicht wird.

Wichtig: Es ist unbedingt zu beachten, daß der Hintergrund (das 'Papier') immer die Farbe schwarz (Nr. 0) hat. Dieses gilt für beide Versionen. (Daraus erklärt sich auch die Verminderung der max. Farbenzahl um eins bei Version 1.)

Screen Pack Plus ermöglicht zwei Bildschirmvarianten: Bei der Version 1 kann nach der Aktivierung in jedem Mode die doppelte Anzahl von Farben (-1) verwendet werden (siehe Abbildung 3). Diese Erhöhung der verwendbaren Farbstifte hat keinerlei Einfluß auf die Programmierung. Sie können durch die normalen BASIC 'PEN'- und 'INK'-Kommandos angewählt werden, da das Betriebssystem entsprechend angepaßt wurde. Lediglich in Mode 0 müssen zur Auswahl der Stifte 16 bis 30 die Befehle 5 bis 7 angewendet werden. Sie entsprechen, bis auf der möglichen Stiftnummer, den normalen BASIC-Befehlen.

Bei der Version 2 können zwei Auflösungen gleichzeitig benutzt werden. Es ist somit möglich, daß Grafiken, die in Mode 0 erstellt wurden, an jeder (!) beliebigen Stelle in Mode 2 (80 Zeichen/Zeile) beschriftet werden können. Dabei geschieht die Ausgabe in zwei Bildschirme (Nr. 1 und 2). Für jeden der Bildschirme kann ein Mode separat festgelegt werden. Dieses geschieht durch den Befehl 'BV.ON'. Zur Ausgabe muß der gewünschte Bildschirm erst angewählt werden (mit 'Bildschirm'), danach erfolgen alle Angaben in diesen Bildschirm. Der andere Bildschirm wird dabei automatisch, sofern nötig, angepaßt. Es kann jederzeit zwischen den Bildschirmen umgeschaltet werden. Es ist zu beachten, daß bei jeder Umschaltung die Fenstergrenzen sowie alle anderen vom Mode abhängigen Werte auf ihre Standards gesetzt werden. Die Farben beider Bildschirme können unterschiedlich sein, lediglich die Hintergrundfarbe ist ausgenommen. Diese muß, wie oben bereits erwähnt, immer schwarz sein, da ansonsten die Bilddarstellung nicht korrekt ist bzw. stark flimmernd erscheint.

Die Befehle im einzelnen

1.) BV.ON (,Version, Parameter (abhängig von der Version))

Wie bei allen Befehlen steht hier das "BV" für Bildschirm- Verwaltung. Der Befehl kann in drei Variationen angewendet werden:

a) BV.ON, 1 (, Mode) - Hiermit wird die Version 1 eingeschaltet (siehe Programm-Beschreibung). Zusätzlich kann (wahlweise!) noch ein Mode für den Bildschirm angegeben werden. In diesem Fall wird der Bildschirm gelöscht, nachdem die Erweiterung eingeschaltet worden ist. Wird kein Mode angegeben, so wird der Momentane beibehalten.

b) BV.ON, 2, Mode Bildschirm 2, Mode Bildschirm 1 - Mit diesem Befehl kann die Version 2 (zwei Modes) eingeschaltet werden. Als Parameter werden die beiden Modes nach der Versionsnummer angehängt. Dabei gilt der letzte Parameter für den Bildschirm Nr. 1, während der zweite Parameter für den zweiten Bildschirm zuständig ist. Der Bildschirm wird gelöscht und Sie befinden sich auf dem "ersten" Bildschirm.

c) BV.ON (keine Parameter) - Durch diesen Befehl läßt sich die letzte Konfiguration, welche bei der letzten Anwendung des Befehles 'BV.OFF' gerade aktiv war, wieder herstellen.

2.) BV.OFF (keine Parameter)
Mit diesem Befehl kann die Erweiterung wieder ausgeschaltet werden. Der Bildschirm wird gelöscht und der alte Mode (der vor der Einschaltung mit 'BV.ON' definiert war) wird wieder eingestellt.

3.) BV.FLAG, 0 oder 1
Folgt hier eine "0" auf dem Befehlsnamen, so wird der Bildschirm nach Aufruf von BV.OFF nicht gelöscht. Ansonsten erfolgt die Herstellung des vor dem Aufruf der Erweiterung gültigen Modes (Standard).

4.) F.MELDUNG, 0 oder 1
Screen Pack Plus gibt bei falscher Anwendung der RSX-Befehle Fehlermeldungen aus. Durch diesen Befehl läßt sich dieser Service ein- (1) oder ausschalten (0).

5.) PEN, Nummer (0-30)
Durch BASIC sind nur maximal 16 Stifte (0 bis 15) auswählbar. Da jedoch bei der Version 1 in Mode 0 31 Stifte möglich sind, muß dieser Befehl zur Auswahl der Stifte 16 bis 30 angewendet werden. Dieser Befehl setzt analog zu BASIC die Nummer des Text- Stiftes.

6.) GPEN,Nummer(0-30)
Dieser Befehl ist das Äquivalent zu Nr. 5 für den Grafikbereich.
Anm. zu den Befehlen 5 und 6: Wurde einer der beiden Befehle angewendet, so muß bei dem Wechsel der Ausgabeart (Grafik/Text) der Befehl für den neuen Ausgabebereich erst erneut angewendet werden, bevor dort die Ausgabe erfolgen kann. Letzteres deshalb, da sich die beiden Befehle gegenseitig beeinflussen. Konkret heißt das, daß nach einer Grafikausgabe vor einer folgenden Textdarstellung erst ein 'PEN'-Kommando angewendet werden muß und umgekehrt.

7.) INK, Stift-Nr. (0-30), erste Farbe (0-26) (, zweite Farbe (0-26)) Der Befehl dient zum Setzen der Farben für die Stifte 16 bis 30. Ansonsten entspricht er genau dem BASIC-Befehl 'INK'.

8.) BILDSCHIRM, Bildschirm-Nr.
Bei eingeschalteter Version 2 ist es mit diesem Befehl möglich, von einem in den anderen Bildschirm zu wechseln und somit auch die Modes zu wechseln. Gültige Parameter sind '1' und '2'.
Der Bildschirm, in dem gewechselt wird, wird vollständig zurückgesetzt (Windows etc.) und der Cursor in die linke obere Ecke gesetzt.
Alle Ausgaben erfolgen jetzt in diesen Bildschirm, unter Beachtung der Schreibart (transparent/ undurchsichtig). Dabei wirkt sich eine Ausgabe von Text oder Grafik auch auf den anderen Bildschirm derart aus, daß verhindert wird, daß ein Pixel gleichzeitig in beiden Schirmen gesetzt ist. Da hier umfangreiche Umwandlungen notwendig sind (von einem Mode in den anderen unter Beachtung der Pixelzahl pro Byte), verlangsamt sich die Ausgabe ein wenig.

9.) WRITE.V, 0 oder 1
Hiermit kann bei der Version 2 bestimmt werden, wie die Grafik-ausgabe geschehen soll. Normalerweise geschieht dies über eine neue Routine von Screen Pack Plus, welche das Vorhandensein zweier Bildschirme berücksichtigt. Soll nun eine noch höhere Geschwindigkeit bei der Graphik-Ausgabe erreicht werden, so kann auch die 'normale' Betriebssystem-Routine zum Einsatz kommen. Diese sollte jedoch nur in Ausnahmefällen, und wenn sichergestellt ist, daß sich keine Grafik überlappt, gewählt werden.
Bei Angabe einer "1" wird die Screen Pack Plus-Routine zur Grafikausgabe verwendet (Standard), wodurch gesichert ist, daß kein Pixel gleichzeitig in beiden Bildschirmen gesetzt wird.

10.) CLS, Bildschirm-Nr.
Durch diesen Befehl wird der gewünschte Bildschirm gelöscht (für Version 2). Der Inhalt des anderen bleibt im Gegensatz zum BASIC-'CLS' erhalten.

11.) TRANSP., 0 oder 1
Dieser Befehl bewirkt ein ein- (1) oder ausschalten (0) der möglichen Transparent-Darstellung für den Text-Bildschirm.

(Klaus Kremer/jb)

für 464-664-6128

```
10 '********** BASIC-Loader 'SCREEN PACK P [2555]
LUS' **********
20 '                                       [117]
30 'Programm-Stand: 02.09.1987 ----------- [4072]
---------------
40 '                                       [117]
50 'Version 1.1 (fuer Schneider/ Amstrad C [3368]
PC 464!) ------
60 '                                       [117]
70 '                3/1987 by Klaus Kremer - [3041]
---------------
80 '                                       [117]
90 '---------------------------------------- [2829]
---------------
100 '- //////////////////////////////////// [2557]
/////////// -
110 '---------------------------------------- [2653]
---------------
120 '************************************** [2154]
**************
130 '                                      [117]
140 'Initalisierung mit CALL 32768 (&8000) [2564]
150 '                                      [117]
160 'Vor der Laden des M-Codes & der Inita [4312]
lisierung muss der
170 'Speicherbereich durch MEMORY &3fff ge [3066]
schuetzt werden!
180 '                                      [117]
190 MEMORY &7FFF                           [150]
200 '                                      [117]
210 FOR i=32768 TO 37301:READ a:POKE i,a:N [1946]
EXT
220 '                                      [117]
230 INPUT"Soll der M-Code (Screen Pack Plu [9920]
s) auf Diskette/ Kassette abgespeichert we
rden (j/n)";a$: IF a$="j" OR a$="J" THEN 2
40 ELSE 250
240 SAVE"spp-464 ",b,32768,4534            [1258]
250 MODE 1:PRINT"Initalisierung durch":PRI [12541]
NT:PRINT"  'memory &3fff:call &8000'":PRIN
T:PRINT"Dieses Ladeprogramm muss vorher":P
RINT"geloescht werden!":PRINT
260 END                                    [110]
270 '                                      [117]
280 DATA &01,&15,&80,&21,&11,&80,&CD,&D1,& [3181]
BC,&3E,&C9
290 DATA &32,&00,&80,&C3,&15,&87,&00,&00,& [1943]
00,&00,&38
300 DATA &80,&C3,&8A,&80,&C3,&9B,&81,&C3,& [2441]
37,&83,&C3
310 DATA &44,&83,&C3,&6E,&8D,&C3,&FA,&8E,& [2852]
C3,&A9,&8D
320 DATA &C3,&CD,&81,&C3,&B5,&85,&C3,&21,& [2476]
8A,&C3,&27
330 DATA &86,&42,&56,&2E,&4F,&CE,&42,&56,& [2050]
2E,&4F,&46
340 DATA &C6,&50,&45,&CE,&47,&50,&45,&CE,& [3057]
42,&49,&4C
350 DATA &44,&53,&43,&48,&49,&52,&CD,&46,& [2369]
2E,&4D,&45
360 DATA &4C,&44,&55,&4E,&C7,&57,&52,&49,& [1642]
54,&45,&2E
370 DATA &D6,&42,&56,&2E,&46,&4C,&41,&C7,& [2228]
43,&4C,&D3
380 DATA &49,&4E,&CB,&54,&52,&41,&4E,&53,& [2143]
50,&AE,&00
390 DATA &00,&00,&00,&00,&00,&00,&00,&00,& [1779]
01,&C0,&40
```



```
400 DATA &00,&02,&02,&02,&00,&00,&00,&FE,& [1961]
05,&D2,&1A
410 DATA &81,&B7,&CA,&8D,&81,&F5,&3A,&C8,& [832]
B1,&32,&99
420 DATA &81,&F1,&CD,&7D,&81,&DD,&E5,&47,& [1720]
DD,&23,&DD
430 DATA &23,&10,&FA,&47,&DD,&2B,&DD,&2B,& [1696]
DD,&7E,&00
440 DATA &DD,&E1,&FE,&01,&28,&72,&FE,&02,& [2721]
3E,&01,&C2
450 DATA &04,&8F,&05,&3E,&03,&CA,&04,&8F,& [2277]
C5,&CD,&0B
460 DATA &BC,&32,&98,&81,&CD,&28,&8D,&C1,& [1891]
11,&21,&00
470 DATA &21,&D1,&82,&FD,&E5,&FD,&21,&85,& [2011]
80,&DD,&7E
480 DATA &00,&FE,&03,&30,&2F,&CD,&1C,&86,& [2372]
77,&FD,&77
490 DATA &00,&FD,&23,&23,&71,&DD,&7E,&00,& [1378]
DD,&23,&DD
500 DATA &23,&19,&10,&E4,&FD,&E1,&3E,&C0,& [2042]
CD,&08,&BC
510 DATA &3E,&01,&32,&89,&80,&3A,&85,&80,& [1675]
CD,&0E,&BC
520 DATA &3E,&01,&CD,&6C,&81,&C3,&D5,&85,& [2609]
FD,&E1,&CD
530 DATA &68,&8D,&3E,&02,&C3,&04,&8F,&3E,& [1899]
03,&C3,&04
540 DATA &8F,&02,&02,&00,&01,&00,&00,&00,& [1808]
00,&00,&78
550 DATA &FE,&01,&28,&2B,&FE,&02,&3E,&03,& [1846]
C2,&04,&8F
560 DATA &DD,&7E,&00,&FE,&03,&4F,&3E,&02,& [2438]
D2,&04,&8F
570 DATA &79,&37,&F5,&CD,&1C,&86,&21,&D1,& [1695]
82,&11,&21
580 DATA &00,&77,&23,&71,&19,&77,&23,&71,& [1303]
19,&77,&23
590 DATA &71,&18,&06,&B7,&3A,&C8,&B1,&18,& [2670]
E3,&CD,&0B
600 DATA &BC,&32,&98,&81,&CD,&51,&83,&F1,& [2188]
DC,&0E,&BC
610 DATA &AF,&32,&80,&80,&AF,&32,&9B,&81,& [1931]
CD,&9F,&BB
620 DATA &3E,&C9,&32,&8A,&80,&18,&5E,&11,& [2002]
1F,&81,&12
630 DATA &DD,&E5,&E1,&4F,&CB,&21,&06,&00,& [2473]
13,&ED,&B0
640 DATA &C9,&21,&1F,&81,&7E,&23,&E5,&DD,& [1837]
E1,&C3,&8A
650 DATA &80,&00,&00,&01,&C9,&3A,&80,&80,& [1801]
B7,&20,&26
660 DATA &CD,&AE,&83,&CD,&FC,&81,&AF,&CD,& [1635]
9F,&BB,&3A
670 DATA &98,&81,&CD,&08,&BC,&3A,&9A,&81,& [1848]
B7,&28,&06
680 DATA &3A,&99,&81,&CD,&0E,&BC,&3E,&C9,& [2310]
32,&9B,&81
690 DATA &AF,&32,&8A,&80,&C9,&CD,&68,&8D,& [2653]
18,&D8,&FE
700 DATA &01,&3E,&04,&C2,&04,&8F,&DD,&7E,& [901]
00,&32,&9A
710 DATA &81,&C9,&00,&CD,&0B,&BC,&32,&FB,& [2242]
81,&21,&FE
720 DATA &B1,&E5,&CD,&DD,&BC,&11,&30,&82,& [2235]
06,&81,&E1
730 DATA &3E,&C9,&32,&DB,&81,&AF,&32,&FC,& [1482]
81,&C3,&D7
740 DATA &BC,&00,&C9,&21,&FE,&B1,&E5,&CD,& [1649]
DD,&BC,&3A
750 DATA &FB,&81,&0F,&0F,&E6,&30,&6F,&01,& [960]
0C,&BC,&F3
760 DATA &ED,&49,&06,&BF,&ED,&78,&E6,&0F,& [2368]
06,&BD,&B5
770 DATA &ED,&79,&FB,&11,&5B,&0D,&06,&81,& [1497]
E1,&3E,&C9
780 DATA &32,&FC,&81,&AF,&32,&DB,&81,&C3,& [2140]
D7,&BC,&21
790 DATA &88,&80,&7E,&3C,&FE,&02,&38,&01,& [3131]
AF,&77,&FE
800 DATA &01,&3E,&C0,&11,&D1,&82,&38,&0B,& [1852]
3E,&40,&11
810 DATA &F3,&82,&28,&04,&AF,&11,&15,&83,& [1919]
0F,&0F,&6F
820 DATA &F3,&01,&0C,&BC,&ED,&49,&06,&BF,& [2846]
ED,&78,&E6
830 DATA &0F,&B5,&06,&BD,&ED,&79,&FB,&3A,& [2461]
80,&80,&B7
```

Listing "Screen Pack Plus"

```
840 DATA &28,&0F,&1A,&E6,&03,&F3,&D9,&CB,& [1830]
89,&CB,&81
850 DATA &B1,&4F,&ED,&49,&FB,&D9,&13,&1A,& [3385]
47,&13,&21
860 DATA &FD,&B1,&35,&20,&12,&CD,&C5,&82,& [1692]
32,&FD,&B1
870 DATA &21,&FB,&B1,&7E,&2F,&77,&AF,&32,& [1351]
FC,&B1,&18
880 DATA &0A,&2B,&7E,&B7,&28,&05,&CD,&C5,& [2317]
82,&18,&F0
890 DATA &3A,&FB,&B1,&B7,&21,&10,&00,&20,& [2126]
02,&19,&EB
900 DATA &68,&01,&00,&7F,&ED,&49,&1A,&F6,& [2118]
40,&ED,&79
910 DATA &13,&0C,&ED,&49,&1A,&F6,&40,&ED,& [2563]
79,&13,&0C
920 DATA &7D,&B9,&20,&EA,&C9,&3A,&FB,&B1,& [2523]
B7,&3A,&D8
930 DATA &B1,&C8,&3A,&D7,&B1,&C9,&02,&02,& [3190]
14,&16,&15
940 DATA &1C,&18,&1D,&0C,&05,&0D,&16,&06,& [2125]
17,&1E,&00
950 DATA &1F,&0E,&14,&16,&15,&1C,&18,&1D,& [2131]
0C,&05,&0D
960 DATA &16,&06,&17,&1E,&00,&1F,&0E,&02,& [2015]
02,&14,&1C
970 DATA &0F,&12,&02,&13,&1A,&19,&1B,&0A,& [2534]
03,&0B,&01
980 DATA &08,&09,&10,&14,&1C,&0F,&12,&02,& [2056]
13,&1A,&19
990 DATA &1B,&0A,&03,&0B,&01,&08,&09,&10,& [1987]
02,&02,&14
1000 DATA &04,&15,&1C,&18,&1D,&0C,&05,&0D, [1463]
&16,&06,&17
1010 DATA &1E,&00,&1F,&0E,&14,&04,&15,&1C, [2191]
&18,&1D,&0C
1020 DATA &05,&0D,&16,&06,&17,&1E,&00,&1F, [1780]
&0E,&FE,&01
1030 DATA &3E,&08,&C2,&04,&8F,&DD,&7E,&00, [2662]
&C3,&54,&85
1040 DATA &FE,&01,&3E,&08,&C2,&04,&8F,&DD, [1937]
&7E,&00,&C3
1050 DATA &5A,&85,&CD,&B1,&BB,&3A,&CB,&83, [3203]
&47,&E5,&21
1060 DATA &CC,&83,&5E,&16,&00,&23,&E3,&E5, [1644]
&19,&19,&19
1070 DATA &EB,&E1,&E3,&7E,&23,&4E,&23,&E5, [2444]
&6F,&61,&73
1080 DATA &23,&72,&E1,&10,&E7,&E1,&CD,&99, [2338]
&83,&78,&56
1090 DATA &2B,&5E,&D5,&2B,&56,&2B,&5E,&2B, [3203]
&E3,&E5,&01
1100 DATA &03,&00,&C5,&ED,&B0,&C1,&D1,&E1, [1690]
&2B,&2B,&E5
1110 DATA &ED,&B0,&E1,&2B,&47,&10,&E1,&C9, [2433]
&3A,&CA,&83
1120 DATA &5F,&47,&16,&00,&62,&6B,&19,&19, [1754]
&19,&19,&19
1130 DATA &19,&2B,&11,&FF,&83,&19,&C9,&CD, [1735]
&99,&83,&78
1140 DATA &56,&2B,&5E,&2B,&D5,&56,&2B,&5E, [1986]
&2B,&E3,&EB
1150 DATA &01,&03,&00,&ED,&B0,&E1,&2B,&2B, [1253]
&2B,&47,&10
1160 DATA &E8,&C9,&26,&11,&00,&97,&84,&01, [2237]
&9E,&84,&04
1170 DATA &A5,&84,&08,&AC,&84,&09,&B3,&84, [1737]
&0A,&BA,&84
1180 DATA &0B,&C1,&84,&0C,&C8,&84,&0D,&CF, [2966]
&84,&0F,&D6
1190 DATA &84,&10,&DD,&84,&11,&E4,&84,&12, [2112]
&EB,&84,&13
1200 DATA &F2,&84,&14,&F9,&84,&15,&00,&85, [2170]
&1C,&07,&85
1210 DATA &C3,&5A,&85,&5D,&85,&DE,&BB,&C3, [2432]
&54,&85,&57
1220 DATA &85,&90,&BB,&C3,&D5,&85,&FB,&85, [2311]
&EB,&BD,&C3
1230 DATA &6F,&86,&A1,&86,&E8,&BD,&C3,&A4, [1989]
&86,&00,&87
1240 DATA &E5,&BD,&C3,&91,&89,&CC,&89,&50, [2173]
&BC,&C3,&34
1250 DATA &86,&55,&86,&9F,&BB,&C3,&30,&86, [2110]
&51,&85,&A2
1260 DATA &BB,&C3,&FE,&85,&19,&86,&0E,&BC, [1810]
&C3,&D3,&89
1270 DATA &1E,&8A,&47,&BC,&C3,&CF,&89,&1B, [2857]
&8A,&44,&BC
```

Listing "Screen Pack Plus"

```
1280 DATA &C3,&43,&8A,&D8,&8A,&32,&BC,&C3, [1521]
&E5,&8B,&63
1290 DATA &8C,&4D,&BC,&C3,&66,&8C,&39,&85, [2616]
&D9,&BD,&C3
1300 DATA &85,&8C,&3C,&85,&5D,&BB,&C3,&9C, [2523]
&8C,&3F,&85
1310 DATA &CD,&BD,&C3,&9C,&8C,&42,&85,&D0, [2045]
&BD,&C3,&A5
1320 DATA &8C,&45,&85,&8A,&BB,&C3,&A5,&8C, [3316]
&48,&85,&8D
1330 DATA &BB,&C3,&33,&8B,&4B,&85,&6C,&BB, [2528]
&C3,&50,&8B
1340 DATA &4E,&85,&DB,&BB,&00,&0A,&8B,&09, [2125]
&85,&00,&00
1350 DATA &01,&85,&8C,&0C,&85,&00,&00,&01, [1922]
&FE,&85,&0F
1360 DATA &85,&00,&00,&00,&10,&8B,&12,&85, [1766]
&00,&00,&00
1370 DATA &15,&8B,&15,&85,&00,&00,&00,&1A, [2476]
&8B,&18,&85
1380 DATA &00,&00,&00,&1F,&8B,&1B,&85,&00, [2348]
&00,&00,&41
1390 DATA &8B,&1E,&85,&00,&00,&00,&DB,&8A, [1523]
&07,&8B,&00
1400 DATA &00,&01,&54,&85,&21,&85,&00,&00, [1689]
&00,&86,&8B
1410 DATA &24,&85,&00,&00,&00,&91,&8B,&27, [2236]
&85,&00,&00
1420 DATA &00,&9F,&8B,&2A,&85,&00,&00,&00, [1742]
&A9,&8B,&2D
1430 DATA &85,&00,&00,&00,&C0,&8B,&30,&85, [2014]
&00,&00,&01
1440 DATA &D7,&8B,&33,&85,&00,&00,&03,&DC, [2368]
&8B,&36,&85
1450 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1460 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1470 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1480 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1490 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1500 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1510 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
1520 DATA &CD,&60,&85,&00,&00,&00,&CD,&60, [1693]
&85,&00,&00
1530 DATA &00,&E5,&C5,&F5,&3A,&80,&80,&B7, [2070]
&20,&48,&3A
1540 DATA &C8,&B1,&FE,&01,&0E,&03,&28,&06, [2631]
&0E,&01,&30
1550 DATA &02,&0E,&0F,&3A,&36,&82,&6F,&67, [1616]
&3D,&47,&79
1560 DATA &28,&03,&81,&10,&FD,&47,&04,&F1, [2432]
&F5,&B8,&38
1570 DATA &02,&78,&3D,&B9,&38,&06,&28,&04, [1771]
&91,&2D,&18
1580 DATA &F7,&4F,&F1,&79,&F5,&7C,&95,&3C, [1359]
&32,&81,&80
1590 DATA &4F,&0D,&06,&00,&21,&82,&80,&09, [2677]
&7E,&CD,&08
1600 DATA &BC,&F1,&C1,&E1,&C9,&F1,&C1,&E1, [1848]
&C9,&FE,&01
1610 DATA &C0,&DD,&7E,&00,&21,&36,&82,&3D, [1736]
&BE,&D0,&21
1620 DATA &82,&80,&5F,&16,&00,&19,&56,&1E, [2127]
&01,&01,&FF
1630 DATA &3F,&62,&2E,&00,&75,&ED,&B0,&C9, [2479]
&CD,&FC,&81
1640 DATA &21,&00,&00,&CD,&05,&BC,&21,&82, [2300]
&80,&3A,&36
1650 DATA &82,&47,&C5,&E5,&56,&1E,&01,&62, [2044]
&2E,&00,&01
1660 DATA &FF,&3F,&75,&ED,&B0,&E1,&23,&C1, [1885]
&10,&ED,&C3
1670 DATA &DB,&81,&00,&00,&00,&F5,&CD,&19, [1965]
&86,&F1,&CD
1680 DATA &1C,&86,&21,&D1,&82,&11,&21,&00, [2971]
&06,&03,&77
1690 DATA &23,&71,&19,&10,&FA,&3A,&58,&86, [2837]
&18,&1B,&00
1700 DATA &00,&00,&FE,&01,&0E,&04,&C8,&0E, [1156]
&10,&D8,&0E
1710 DATA &02,&C9,&FE,&01,&C0,&DD,&7E,&00, [1928]
&C3,&9F,&BB
1720 DATA &3A,&58,&86,&C9,&21,&6F,&86,&22, [1793]
&68,&86,&21
```

Listing "Screen Pack Plus"

```
1730 DATA &59,&86,&B7,&32,&58,&86,&28,&03, [1685]
&21,&6A,&86
1740 DATA &22,&91,&B2,&C9,&B7,&21,&6F,&86, [1809]
&28,&E7,&21
1750 DATA &CF,&8D,&18,&E2,&00,&00,&00,&00, [1723]
&2A,&8F,&B2
1760 DATA &79,&2F,&A4,&47,&79,&A5,&B0,&0E, [1783]
&FF,&47,&EB
1770 DATA &C3,&6F,&86,&3A,&8F,&B2,&18,&F6, [2603]
&7E,&A8,&B1
1780 DATA &A9,&A8,&77,&3A,&36,&82,&FE,&01, [2780]
&C8,&D5,&11
1790 DATA &82,&80,&C5,&47,&C5,&7C,&E6,&C0, [2378]
&47,&1A,&B8
1800 DATA &28,&10,&E5,&47,&7C,&E6,&3F,&B0, [1398]
&67,&06,&00
1810 DATA &7E,&A8,&B1,&A9,&A8,&77,&E1,&13, [1340]
&C1,&10,&E3
1820 DATA &C1,&D1,&C9,&00,&00,&00,&3A,&36, [1901]
&82,&C5,&D5
1830 DATA &47,&11,&82,&80,&7C,&E6,&3F,&67, [1714]
&1A,&84,&67
1840 DATA &7E,&A1,&B7,&20,&03,&13,&10,&F1, [2146]
&D5,&C5,&11
1850 DATA &08,&00,&0F,&CB,&12,&CB,&09,&38, [2332]
&02,&CB,&1A
1860 DATA &1D,&20,&F4,&7A,&CD,&03,&87,&B7, [1727]
&C1,&D1,&28
1870 DATA &24,&E5,&21,&82,&80,&EB,&ED,&52, [2086]
&45,&04,&E1
1880 DATA &4F,&3A,&C8,&B1,&FE,&01,&3E,&03, [2131]
&28,&06,&3E
1890 DATA &01,&30,&02,&3E,&0F,&5F,&83,&10, [2367]
&FD,&93,&93
1900 DATA &81,&D1,&C1,&C9,&AF,&B8,&20,&AF, [1921]
&18,&F7,&00
1910 DATA &00,&00,&57,&3A,&C8,&B1,&FE,&01, [2539]
&7A,&D0,&0F
1920 DATA &0F,&CE,&00,&0F,&9F,&E6,&06,&AA, [2495]
&C9,&DD,&21
1930 DATA &53,&87,&3E,&03,&CD,&8A,&80,&21, [2055]
&92,&87,&CD
1940 DATA &5E,&87,&DD,&21,&59,&87,&3E,&01, [2190]
&CD,&6E,&8D
1950 DATA &3E,&01,&CD,&4A,&86,&21,&D3,&87, [2161]
&CD,&5E,&87
1960 DATA &CD,&18,&BB,&CD,&9B,&81,&DD,&21, [2191]
&5A,&87,&3E
1970 DATA &02,&CD,&8A,&80,&21,&D3,&88,&CD, [2458]
&5E,&87,&CD
1980 DATA &18,&BB,&C3,&9B,&81,&00,&00,&02, [2453]
&00,&02,&00
1990 DATA &02,&02,&00,&01,&00,&7E,&23,&B7, [2288]
&C8,&FE,&1B
2000 DATA &28,&09,&FE,&19,&28,&10,&CD,&5A, [1371]
&BB,&18,&EF
2010 DATA &46,&23,&7E,&23,&CD,&5A,&BB,&10, [1590]
&FB,&18,&E4
2020 DATA &46,&23,&0E,&00,&1E,&0F,&7B,&CD, [1920]
&5A,&BB,&79
2030 DATA &CD,&5A,&BB,&7E,&CD,&5A,&BB,&0C, [2320]
&10,&F1,&23
2040 DATA &18,&CC,&1C,&01,&18,&18,&0C,&2A, [2022]
&1F,&14,&01
2050 DATA &2A,&1F,&03,&03,&53,&43,&52,&45, [2487]
&45,&4E,&20
2060 DATA &50,&41,&43,&4B,&20,&50,&4C,&55, [2561]
&53,&1F,&03
2070 DATA &05,&19,&10,&8F,&0F,&01,&1F,&01, [1708]
&06,&2A,&20
2080 DATA &19,&10,&8F,&0F,&01,&20,&2A,&1F, [2651]
&03,&07,&19
2090 DATA &10,&8F,&1F,&05,&0C,&0F,&02,&50, [1781]
&4C,&55,&53
2100 DATA &00,&1C,&01,&1A,&1A,&0F,&01,&1F, [1598]
&05,&01,&1B
2110 DATA &48,&2A,&1F,&01,&02,&2A,&1F,&50, [1875]
&02,&2A,&1F
2120 DATA &01,&03,&2A,&1F,&50,&03,&2A,&1F, [2098]
&01,&04,&2A
2130 DATA &1F,&50,&04,&2A,&1F,&01,&05,&2A, [1736]
&1F,&50,&05
2140 DATA &2A,&1F,&08,&06,&2A,&2A,&2A,&2A, [2226]
&20,&49,&64
2150 DATA &65,&65,&2C,&20,&50,&72,&6F,&67, [2100]
&72,&61,&6D
2160 DATA &6D,&69,&65,&72,&75,&6E,&67,&20, [2712]
&26,&20,&43
```

Listing "Screen Pack Plus"


```
2170 DATA &4F,&50,&59,&52,&49,&47,&48,&54, [2297]
&20,&28,&63
2180 DATA &29,&20,&33,&2F,&31,&39,&38,&37, [1762]
&20,&62,&79
2190 DATA &20,&4B,&2E,&20,&4B,&72,&65,&6D, [1782]
&65,&72,&20
2200 DATA &20,&2A,&2A,&2A,&2A,&1F,&01,&0C, [1311]
&4D,&69,&74
2210 DATA &20,&53,&43,&52,&45,&45,&4E,&20, [2220]
&50,&41,&43
2220 DATA &4B,&1F,&22,&0C,&6B,&6F,&65,&6E, [1133]
&6E,&65,&6E
2230 DATA &20,&53,&69,&65,&20,&6A,&65,&74, [1327]
&7A,&74,&20
2240 DATA &32,&20,&41,&75,&66,&6C,&6F,&65, [1934]
&73,&75,&6E
2250 DATA &67,&65,&6E,&20,&67,&6C,&65,&69, [1956]
&63,&68,&7A
2260 DATA &65,&69,&74,&69,&67,&1F,&01,&0D, [2743]
&61,&75,&66
2270 DATA &20,&64,&65,&6D,&20,&67,&61,&6E, [1909]
&7A,&65,&6E
2280 DATA &20,&28,&21,&29,&20,&42,&69,&6C, [1218]
&64,&73,&63
2290 DATA &68,&69,&72,&6D,&20,&62,&65,&6E, [1440]
&75,&74,&7A
2300 DATA &65,&6E,&21,&1F,&02,&10,&42,&69, [2145]
&74,&74,&65
2310 DATA &20,&64,&72,&75,&65,&63,&6B,&65, [2244]
&6E,&20,&53
2320 DATA &69,&65,&20,&65,&69,&6E,&65,&20, [1473]
&54,&61,&73
2330 DATA &74,&65,&21,&00,&1C,&01,&15,&15, [1596]
&1C,&02,&06
2340 DATA &06,&53,&69,&65,&20,&6B,&6F,&65, [1218]
&6E,&6E,&65
2350 DATA &6E,&20,&6A,&65,&74,&7A,&74,&20, [2009]
&61,&75,&63
2360 DATA &68,&20,&62,&65,&69,&20,&0F,&02, [2253]
&38,&30,&20
2370 DATA &5A,&65,&69,&63,&68,&65,&6E,&2F, [1522]
&20,&5A,&65
2380 DATA &69,&6C,&65,&20,&0F,&01,&6D,&65, [2469]
&68,&72,&65
2390 DATA &72,&65,&20,&46,&61,&72,&62,&65, [1960]
&6E,&20,&61
2400 DATA &75,&66,&20,&64,&65,&6D,&20,&0F, [2955]
&02,&67,&61
2410 DATA &6E,&7A,&65,&6E,&20,&0F,&01,&42, [2052]
&69,&6C,&64
2420 DATA &2D,&73,&63,&68,&69,&72,&6D,&20, [1808]
&62,&65,&6E
2430 DATA &75,&74,&7A,&65,&6E,&21,&1F,&01, [2652]
&04,&44,&69
2440 DATA &65,&73,&20,&67,&69,&6C,&74,&20, [2005]
&66,&75,&65
2450 DATA &72,&20,&0F,&02,&6A,&65,&64,&65, [1792]
&6E,&20,&0F
2460 DATA &01,&65,&69,&6E,&7A,&65,&6C,&6E, [1906]
&65,&6E,&20
2470 DATA &50,&69,&78,&65,&6C,&21,&1F,&01, [1860]
&08,&42,&69
2480 DATA &74,&74,&65,&20,&64,&72,&75,&65, [1430]
&63,&6B,&65
2490 DATA &6E,&20,&53,&69,&65,&20,&65,&69, [2312]
&6E,&65,&20
2500 DATA &54,&61,&73,&74,&65,&21,&00,&32, [2114]
&B4,&89,&3A
2510 DATA &CB,&B1,&ED,&53,&B6,&89,&57,&3A, [1898]
&36,&82,&5F
2520 DATA &1C,&E5,&21,&82,&80,&18,&08,&7E, [1861]
&23,&BA,&28
2530 DATA &17,&32,&CB,&B1,&E3,&E5,&D5,&C5, [1650]
&3E,&00,&11
2540 DATA &00,&00,&CD,&CC,&89,&C1,&D1,&E1, [2904]
&E3,&AF,&32
2550 DATA &B4,&89,&1D,&20,&E1,&7A,&32,&CB, [2140]
&B1,&E1,&C9
2560 DATA &00,&00,&00,&4F,&DF,&18,&8A,&D5, [2069]
&C5,&E5,&CD
2570 DATA &1E,&8A,&E1,&C1,&7C,&E6,&C0,&47, [2332]
&7C,&E6,&3F
2580 DATA &67,&D1,&C5,&D5,&E5,&21,&82,&80, [1884]
&3A,&36,&82
2590 DATA &4F,&7E,&23,&22,&11,&8A,&B8,&28, [2752]
&1F,&E1,&E6
2600 DATA &C0,&B4,&67,&79,&D1,&C1,&C5,&D5, [2900]
&E5,&F5,&0E
2610 DATA &00,&CD,&1E,&8A,&F1,&E1,&D1,&C1, [2764]
&3D,&C8,&C5
```

Listing "Screen Pack Plus"

```
2620 DATA &D5,&E5,&21,&00,&00,&18,&D8,&79, [2593]
&18,&F0,&95
2630 DATA &0B,&FE,&00,&00,&00,&00,&00,&00, [2085]
&FE,&03,&28
2640 DATA &09,&FE,&02,&28,&11,&3E,&05,&C3, [1498]
&04,&8F,&DD
2650 DATA &4E,&00,&DD,&46,&02,&DD,&7E,&04, [2165]
&C3,&32,&BC
2660 DATA &DD,&46,&00,&DD,&7E,&02,&48,&18, [2959]
&F4,&F5,&3A
2670 DATA &C8,&B1,&FE,&01,&26,&09,&2E,&03, [2137]
&28,&0A,&26
2680 DATA &2D,&2E,&0F,&38,&04,&26,&03,&2E, [2506]
&01,&F1,&BC
2690 DATA &38,&01,&7C,&B7,&28,&42,&5D,&1C, [1861]
&26,&00,&BB
2700 DATA &38,&07,&24,&95,&BB,&38,&02,&95, [2088]
&24,&5C,&57
2710 DATA &7C,&21,&D1,&82,&B7,&28,&0B,&C5, [2805]
&D5,&43,&11
2720 DATA &22,&00,&19,&10,&FD,&D1,&C1,&23, [2237]
&23,&5A,&16
2730 DATA &00,&19,&CD,&06,&B9,&F5,&E5,&78, [1666]
&CD,&0A,&0D
2740 DATA &46,&79,&CD,&0A,&0D,&4E,&E1,&70, [2272]
&11,&10,&00
2750 DATA &19,&71,&F1,&C3,&0C,&B9,&21,&D3, [1076]
&82,&E5,&CD
2760 DATA &89,&8A,&E1,&11,&22,&00,&19,&70, [2142]
&11,&10,&00
2770 DATA &D5,&19,&71,&11,&12,&00,&19,&70, [1905]
&D1,&19,&71
2780 DATA &C9,&FE,&10,&D0,&C5,&4F,&3A,&89, [2504]
&80,&47,&21
2790 DATA &D3,&82,&11,&22,&00,&19,&10,&FD, [1946]
&B7,&ED,&52
2800 DATA &59,&C1,&18,&B0,&00,&00,&00,&CD, [2019]
&E1,&8A,&C3
2810 DATA &33,&15,&CD,&D0,&BD,&2A,&85,&B2, [1790]
&CD,&DA,&11
2820 DATA &22,&85,&B2,&D8,&E5,&21,&8C,&B2, [1873]
&78,&87,&3C
2830 DATA &86,&77,&CD,&56,&12,&3A,&90,&B2, [2281]
&F5,&DC,&91
2840 DATA &89,&F1,&D4,&E5,&8B,&E1,&C9,&00, [2782]
&00,&00,&CD
2850 DATA &E1,&8A,&C3,&CD,&BD,&11,&00,&FF, [2244]
&18,&0D,&11
2860 DATA &00,&01,&18,&08,&11,&01,&00,&18, [2558]
&03,&11,&FF
2870 DATA &00,&D5,&CD,&E1,&8A,&D1,&7D,&83, [1359]
&6F,&7C,&82
2880 DATA &67,&22,&85,&B2,&C3,&CD,&BD,&CD, [2649]
&06,&B9,&F5
2890 DATA &CD,&03,&B9,&CD,&41,&8B,&F1,&C3, [2381]
&0C,&B9,&CD
2900 DATA &D0,&BD,&2A,&88,&B2,&22,&85,&B2, [1954]
&ED,&5B,&8A
2910 DATA &B2,&18,&4A,&CD,&D5,&BB,&B7,&ED, [2847]
&52,&23,&CD
2920 DATA &81,&8B,&CD,&81,&8B,&CB,&3D,&45, [1839]
&ED,&5B,&36
2930 DATA &B3,&2A,&34,&B3,&E5,&B7,&ED,&52, [2181]
&23,&4D,&ED
2940 DATA &5B,&30,&B3,&E1,&C5,&CD,&1D,&BC, [3166]
&D1,&3A,&39
2950 DATA &B3,&4F,&CD,&D3,&89,&CF,&0B,&96, [2410]
&CB,&2C,&CB
2960 DATA &1D,&C9,&CD,&E1,&8A,&54,&5D,&CD, [2412]
&CF,&89,&C3
2970 DATA &CD,&BD,&CD,&E1,&8A,&EB,&6B,&3A, [2250]
&89,&B2,&67
2980 DATA &3A,&90,&B2,&18,&EC,&CD,&E1,&8A, [2105]
&5D,&3A,&8B
2990 DATA &B2,&57,&18,&F1,&CD,&91,&8B,&2A, [3017]
&88,&B2,&3A
3000 DATA &8B,&B2,&57,&3A,&85,&B2,&3D,&5F, [2674]
&BD,&3A,&90
3010 DATA &B2,&D2,&CF,&89,&C9,&CD,&9F,&8B, [2469]
&2A,&88,&B2
3020 DATA &ED,&5B,&8A,&B2,&3A,&85,&B2,&6F, [3011]
&2C,&BB,&3A
3030 DATA &90,&B2,&DA,&CF,&89,&C9,&0F,&9F, [1186]
&C3,&9F,&BB
3040 DATA &23,&7E,&23,&46,&23,&4E,&C3,&32, [2495]
&BC,&4F,&3A
3050 DATA &CB,&B1,&2A,&C9,&B1,&22,&0C,&8C, [2781]
&57,&D5,&C5
3060 DATA &79,&CD,&1F,&8C,&C1,&D1,&3A,&36, [2224]
&82,&21,&82
```

Listing "Screen Pack Plus"

```
3070 DATA &80,&5F,&7E,&23,&BA,&28,&13,&32, [1676]
&CB,&B1,&D5
3080 DATA &E5,&C5,&AF,&21,&00,&00,&22,&C9, [2553]
&B1,&CD,&1F
3090 DATA &8C,&C1,&E1,&D1,&1D,&20,&E5,&7A, [1837]
&32,&CB,&B1
3100 DATA &C9,&4F,&C5,&11,&D0,&FF,&06,&30, [2433]
&CD,&49,&8C
3110 DATA &C1,&CD,&C4,&8C,&78,&B7,&20,&0D, [1717]
&11,&B0,&FF
3120 DATA &CD,&5C,&8C,&11,&00,&00,&06,&20, [2304]
&18,&0B,&11
3130 DATA &50,&00,&CD,&5C,&8C,&11,&B0,&FF, [2470]
&06,&20,&2A
3140 DATA &C9,&B1,&19,&7C,&E6,&07,&67,&3A, [1866]
&CB,&B1,&84
3150 DATA &67,&50,&1E,&08,&C3,&D3,&89,&2A, [2654]
&C9,&B1,&19
3160 DATA &C3,&05,&BC,&00,&00,&00,&4F,&3A, [2108]
&93,&B2,&B7
3170 DATA &79,&C2,&45,&19,&21,&B8,&B2,&46, [2435]
&78,&FE,&0A
3180 DATA &D2,&46,&14,&B7,&C2,&27,&14,&79, [2661]
&FE,&20,&30
3190 DATA &03,&C3,&27,&14,&47,&3A,&8E,&B2, [2231]
&CB,&47,&C8
3200 DATA &C5,&CD,&E1,&8A,&24,&22,&85,&B2, [2775]
&25,&F1,&CD
3210 DATA &D3,&BD,&C3,&CD,&BD,&3A,&8D,&B2, [2448]
&E6,&7F,&B7
3220 DATA &C0,&18,&0E,&CD,&06,&B9,&F5,&CD, [3031]
&03,&B9,&CD
3230 DATA &B3,&8C,&F1,&C3,&0C,&B9,&C5,&D5, [1930]
&E5,&CD,&E4
3240 DATA &8A,&ED,&4B,&8F,&B2,&CD,&DF,&0D, [3612]
&E1,&D1,&C1
3250 DATA &C9,&F5,&C5,&06,&F5,&ED,&78,&1F, [2498]
&30,&FB,&C1
3260 DATA &F1,&C9,&4F,&3E,&01,&32,&27,&8D, [2175]
&79,&4F,&3A
3270 DATA &CB,&B1,&F5,&21,&EB,&BD,&7E,&F5, [1781]
&E5,&3E,&C9
3280 DATA &77,&C5,&CD,&FC,&81,&3A,&27,&8D, [1819]
&B7,&3A,&CB
3290 DATA &B1,&57,&C4,&C9,&85,&C1,&3A,&89, [2413]
&80,&3D,&5F
3300 DATA &16,&00,&21,&85,&80,&19,&71,&47, [1456]
&1E,&22,&21
3310 DATA &D1,&82,&B7,&28,&03,&19,&10,&FD, [2837]
&79,&CD,&1C
3320 DATA &86,&77,&23,&71,&CD,&19,&86,&CD, [1938]
&DB,&81,&E1
3330 DATA &F1,&77,&F1,&CD,&08,&BC,&3A,&58, [2708]
&86,&C3,&4A
3340 DATA &86,&00,&CD,&B1,&BB,&3A,&9E,&90, [1715]
&47,&E5,&21
3350 DATA &9F,&90,&5E,&16,&00,&23,&E3,&E5, [2631]
&19,&19,&19
3360 DATA &EB,&E1,&E3,&7E,&23,&4E,&23,&E5, [2444]
&6F,&61,&73
3370 DATA &23,&72,&E1,&10,&E7,&E1,&CD,&53, [2768]
&8D,&C3,&79
3380 DATA &83,&3A,&9D,&90,&47,&5F,&16,&00, [2261]
&62,&6B,&19
3390 DATA &19,&19,&19,&19,&19,&2B,&11,&CF, [1927]
&90,&19,&C9
3400 DATA &CD,&53,&8D,&C3,&B1,&83,&FE,&01, [2565]
&20,&15,&3A
3410 DATA &8A,&80,&B7,&C8,&3A,&80,&80,&B7, [2741]
&C8,&DD,&7E
3420 DATA &00,&21,&36,&82,&BE,&28,&07,&38, [2114]
&05,&3E,&09
3430 DATA &C3,&04,&8F,&5F,&32,&89,&80,&1D, [2688]
&16,&00,&21
3440 DATA &85,&80,&19,&7E,&F5,&21,&82,&80, [2178]
&19,&7E,&CD
3450 DATA &08,&BC,&AF,&32,&27,&8D,&F1,&C3, [1630]
&D7,&8C,&4F
3460 DATA &3A,&80,&80,&B7,&C8,&3A,&8A,&80, [1870]
&B7,&C8,&79
3470 DATA &FE,&01,&C0,&DD,&7E,&00,&32,&CE, [3066]
&8D,&B7,&11
3480 DATA &A1,&86,&28,&03,&11,&CF,&8D,&21, [1648]
&E9,&BD,&73
3490 DATA &23,&72,&C9,&00,&7E,&A8,&B1,&A9, [868]
&A8,&77,&3A
3500 DATA &36,&82,&FE,&01,&C8,&D5,&11,&82, [1855]
&80,&C5,&47
3510 DATA &C5,&7C,&E6,&C0,&47,&1A,&13,&B8, [1828]
&28,&21,&E5
```

Listing "Screen Pack Plus"

```
3520 DATA &47,&7C,&E6,&3F,&B0,&67,&E5,&D5, [2560]
&13,&13,&3A
3530 DATA &C8,&B1,&07,&07,&EB,&86,&CD,&12, [2774]
&8E,&D1,&E1
3540 DATA &4F,&06,&00,&7E,&A8,&B1,&A9,&A8, [2723]
&77,&E1,&C1
3550 DATA &10,&D2,&C1,&D1,&C9,&21,&23,&8E, [2164]
&16,&00,&E6
3560 DATA &0F,&5F,&19,&19,&5E,&23,&56,&D5, [2088]
&C9,&79,&C9
3570 DATA &21,&8E,&43,&8E,&59,&8E,&21,&8E, [2536]
&83,&8E,&21
3580 DATA &8E,&68,&8E,&21,&8E,&CA,&8E,&A3, [2698]
&8E,&21,&8E
3590 DATA &21,&8E,&21,&8E,&21,&8E,&21,&8E, [2557]
&21,&8E,&79
3600 DATA &E6,&01,&5F,&07,&B3,&5F,&79,&E6, [2333]
&02,&07,&47
3610 DATA &07,&B0,&B3,&5F,&07,&07,&07,&07, [2710]
&B3,&C9,&79
3620 DATA &E6,&A5,&5F,&E6,&A0,&0F,&B3,&5F, [1768]
&79,&E6,&05
3630 DATA &07,&B3,&C9,&79,&E6,&01,&5F,&79, [2803]
&E6,&02,&07
3640 DATA &B3,&5F,&07,&B3,&5F,&79,&E6,&80, [2820]
&47,&79,&E6
3650 DATA &40,&0F,&B0,&47,&0F,&B0,&B3,&C9, [1081]
&79,&E6,&01
3660 DATA &5F,&79,&E6,&02,&0F,&B3,&5F,&79, [2011]
&E6,&04,&47
3670 DATA &79,&E6,&08,&0F,&B0,&0F,&B3,&5F, [2467]
&07,&07,&B3
3680 DATA &5F,&07,&07,&07,&07,&B3,&C9,&79, [2276]
&E6,&55,&5F
3690 DATA &79,&E6,&AA,&0F,&B3,&5F,&E6,&01, [2295]
&47,&7B,&E6
3700 DATA &54,&0F,&B0,&5F,&E6,&28,&0F,&B3, [1863]
&47,&E6,&07
3710 DATA &5F,&78,&E6,&10,&0F,&B3,&5F,&07, [1683]
&07,&07,&07
3720 DATA &B3,&C9,&79,&E6,&11,&5F,&79,&E6, [2746]
&22,&0F,&B3
3730 DATA &5F,&79,&E6,&44,&0F,&0F,&B3,&5F, [1559]
&79,&E6,&88
3740 DATA &0F,&0F,&0F,&B3,&5F,&E6,&01,&47, [1589]
&07,&07,&B0
3750 DATA &47,&7B,&E6,&10,&0F,&6F,&0F,&0F, [2496]
&B5,&B0,&47
3760 DATA &07,&07,&07,&07,&B0,&C9,&FE,&01, [2030]
&C0,&DD,&7E
3770 DATA &00,&32,&9C,&90,&C9,&E5,&D5,&C5, [1452]
&F5,&FE,&0A
3780 DATA &38,&01,&AF,&5F,&3A,&9C,&90,&B7, [2094]
&28,&0E,&21
3790 DATA &27,&8F,&16,&00,&19,&19,&5E,&23, [1391]
&56,&EB,&CD
3800 DATA &5E,&87,&F1,&C1,&D1,&E1,&C9,&3B, [2237]
&8F,&72,&8F
3810 DATA &A5,&8F,&CF,&8F,&FA,&8F,&27,&90, [2121]
&50,&90,&51
3820 DATA &90,&52,&90,&75,&90,&2A,&2A,&20, [1409]
&55,&6E,&62
3830 DATA &65,&6B,&61,&6E,&6E,&74,&65,&72, [2874]
&20,&46,&65
3840 DATA &68,&6C,&65,&72,&20,&62,&65,&69, [1742]
&20,&41,&75
3850 DATA &66,&72,&75,&66,&20,&76,&6F,&6E, [1532]
&20,&53,&43
3860 DATA &52,&45,&45,&4E,&20,&50,&41,&43, [1653]
&4B,&20,&2B
3870 DATA &21,&20,&2A,&2A,&00,&2A,&2A,&20, [1935]
&46,&61,&6C
3880 DATA &73,&63,&68,&65,&20,&56,&65,&72, [2469]
&73,&69,&6F
3890 DATA &6E,&20,&62,&65,&69,&20,&42,&65, [2534]
&66,&65,&68
3900 DATA &6C,&20,&7C,&42,&56,&2E,&4F,&4E, [2276]
&20,&61,&6E
3910 DATA &67,&65,&67,&65,&62,&65,&6E,&21, [2866]
&20,&2A,&2A
3920 DATA &00,&2A,&2A,&20,&46,&61,&6C,&73, [2112]
&63,&68,&65
3930 DATA &72,&20,&4D,&6F,&64,&65,&20,&61, [2233]
&6E,&67,&65
3940 DATA &67,&65,&62,&65,&6E,&20,&62,&65, [1906]
&69,&20,&7C
3950 DATA &42,&56,&2E,&4F,&4E,&21,&20,&2A, [2217]
&2A,&00,&2A
3960 DATA &2A,&20,&46,&61,&6C,&73,&63,&68, [2347]
&65,&20,&50
```

Listing "Screen Pack Plus"


```
3970 DATA &61,&72,&61,&6D,&65,&74,&65,&72, [2052]
&2D,&41,&6E
3980 DATA &7A,&61,&68,&6C,&20,&62,&65,&69, [2674]
&20,&7C,&42
3990 DATA &56,&2E,&4F,&4E,&21,&20,&2A,&2A, [2344]
&00,&2A,&2A
4000 DATA &20,&46,&61,&6C,&73,&63,&68,&65, [1742]
&20,&50,&61
4010 DATA &72,&61,&6D,&65,&74,&65,&72,&2D, [2792]
&41,&6E,&7A
4020 DATA &61,&68,&6C,&20,&62,&65,&69,&20, [2135]
&7C,&42,&56
4030 DATA &2E,&46,&4C,&41,&47,&21,&20,&2A, [2027]
&2A,&00,&2A
4040 DATA &2A,&20,&46,&61,&6C,&73,&63,&68, [2347]
&65,&20,&50
4050 DATA &61,&72,&61,&6D,&65,&74,&65,&72, [2052]
&2D,&41,&6E
4060 DATA &7A,&61,&68,&6C,&20,&62,&65,&69, [2703]
&20,&7C,&49
4070 DATA &4E,&4B,&21,&20,&2A,&2A,&00,&00, [1532]
&00,&2A,&2A
4080 DATA &20,&46,&61,&6C,&73,&63,&68,&65, [1968]
&20,&53,&74
4090 DATA &69,&66,&74,&2D,&4E,&72,&2E,&20, [2119]
&61,&6E,&67
4100 DATA &65,&67,&65,&62,&65,&6E,&21,&20, [2176]
&2A,&2A,&00
4110 DATA &2A,&2A,&20,&46,&61,&6C,&73,&63, [2110]
&68,&65,&72
4120 DATA &20,&42,&69,&6C,&64,&73,&63,&68, [1954]
&69,&72,&6D
4130 DATA &20,&61,&75,&73,&67,&65,&77,&61, [2146]
&65,&68,&6C
4140 DATA &74,&21,&20,&2A,&2A,&00,&01,&21, [2201]
&10,&00,&4B
4150 DATA &91,&01,&52,&91,&04,&59,&91,&08, [2063]
&60,&91,&09
4160 DATA &67,&91,&0A,&6E,&91,&0B,&75,&91, [1852]
&0C,&7C,&91
4170 DATA &0D,&83,&91,&10,&8A,&91,&11,&91, [1373]
&91,&12,&98
4180 DATA &91,&13,&9F,&91,&14,&A6,&91,&16, [3258]
&AD,&91,&1C
4190 DATA &B4,&91,&C3,&CF,&8D,&A1,&86,&E8, [2311]
&BD,&C3,&91
4200 DATA &89,&CC,&89,&50,&BC,&C3,&4A,&86, [2360]
&55,&86,&9F
4210 DATA &BB,&C3,&30,&86,&51,&85,&A2,&BB, [2268]
&C3,&D0,&8C
4220 DATA &19,&86,&0E,&BC,&C3,&D3,&89,&1E, [2388]
&8A,&47,&BC
4230 DATA &C3,&CF,&89,&1B,&8A,&44,&BC,&C3, [2386]
&BF,&8A,&D8
4240 DATA &8A,&32,&BC,&C3,&E5,&8B,&63,&8C, [1567]
&4D,&BC,&C3
4250 DATA &66,&8C,&39,&85,&D9,&BD,&C3,&85, [2373]
&8C,&3C,&85
4260 DATA &5D,&BB,&C3,&9C,&8C,&3F,&85,&CD, [1726]
&BD,&C3,&9C
4270 DATA &8C,&42,&85,&D0,&BD,&C3,&A5,&8C, [3050]
&45,&85,&8A
4280 DATA &BB,&C3,&A5,&8C,&48,&85,&8D,&BB, [2438]
&C3,&33,&8B
4290 DATA &4B,&85,&6C,&BB,&C3,&50,&8B,&4E, [2359]
&85,&DB,&BB
4300 DATA &00,&0A,&8B,&09,&85,&00,&00,&01, [1905]
&85,&8C,&0C
4310 DATA &85,&00,&00,&01,&D0,&8C,&0F,&85, [1463]
&00,&00,&00
4320 DATA &10,&8B,&12,&85,&00,&00,&00,&15, [1603]
&8B,&15,&85
4330 DATA &00,&00,&00,&1A,&8B,&18,&85,&00, [1350]
&00,&00,&1F
4340 DATA &8B,&1B,&85,&00,&00,&00,&41,&8B, [2399]
&1E,&85,&00
4350 DATA &00,&00,&DB,&8A,&07,&8B,&00,&00, [1129]
&00,&86,&8B
4360 DATA &24,&85,&00,&00,&00,&91,&8B,&27, [2236]
&85,&00,&00
4370 DATA &00,&9F,&8B,&2A,&85,&00,&00,&00, [1742]
&A9,&8B,&2D
4380 DATA &85,&00,&00,&00,&C0,&8B,&30,&85, [2014]
&00,&00,&01
4390 DATA &D7,&8B,&33,&85,&00,&00,&03,&DC, [2368]
&8B,&36,&85
4400 DATA &00,&00,&00,&00,&00,&00,&00,&00, [1795]
&00,&00,&00
```

Listing "Screen Pack Plus"

```
50 'Version 1.2 (fuer Schneider/ Amstrad C [2732]
PC 664!) ------
240 SAVE"spp-664 ",b,32768,4534           [1443]
390 DATA &DF,&7D,&80,&C9,&77,&DD,&FD,&00,& [1526]
01,&C0,&40
410 DATA &81,&B7,&CA,&8D,&81,&F5,&3A,&C3,& [1621]
B7,&32,&99
590 DATA &71,&18,&06,&B7,&3A,&C3,&B7,&18,& [2689]
E3,&CD,&0B
710 DATA &81,&C9,&00,&CD,&0B,&BC,&32,&FB,& [2238]
81,&21,&F9
720 DATA &B7,&E5,&CD,&DD,&BC,&11,&30,&82,& [2600]
06,&81,&E1
740 DATA &BC,&00,&C9,&21,&F9,&B7,&E5,&CD,& [2762]
DD,&BC,&3A
770 DATA &ED,&79,&FB,&11,&5D,&0D,&06,&81,& [1391]
E1,&3E,&C9
860 DATA &F8,&B7,&35,&20,&12,&CD,&C5,&82,& [2362]
32,&F8,&B7
870 DATA &21,&F6,&B7,&7E,&2F,&77,&AF,&32,& [2220]
F7,&B7,&18
890 DATA &3a,&f6,&b7,&b7,&21,&10,&00,&20,& [1485]
02,&19,&eb
920 DATA &7D,&B9,&20,&EA,&C9,&3A,&F6,&B7,& [2796]
B7,&3A,&D3
930 DATA &B7,&C8,&3A,&D2,&B7,&C9,&02,&02,& [2358]
14,&16,&15
1540 DATA &C3,&B7,&FE,&01,&0E,&03,&28,&06, [1744]
&0E,&01,&30
1740 DATA &22,&31,&B7,&C9,&B7,&21,&6F,&86, [2439]
&28,&E7,&21
1750 DATA &CF,&8D,&18,&E2,&00,&00,&00,&00, [1838]
&2A,&2F,&B7
1770 DATA &C3,&6F,&86,&3A,&2F,&B7,&18,&F6, [2472]
&7E,&A8,&B1
1880 DATA &4F,&3A,&C3,&B7,&FE,&01,&3E,&03, [1960]
&28,&06,&3E
1910 DATA &00,&00,&57,&3A,&C3,&B7,&FE,&01, [1353]
&7A,&D0,&0F
2510 DATA &C6,&B7,&ED,&53,&B6,&89,&57,&3A, [1954]
&36,&82,&5F
2530 DATA &17,&32,&C6,&B7,&E3,&E5,&D5,&C5, [2033]
&3E,&00,&11
2550 DATA &B4,&89,&1D,&20,&E1,&7A,&32,&C6, [1948]
&B7,&E1,&C9
2620 DATA &D5,&E5,&21,&00,&00,&18,&D8,&79, [2591]
&18,&F0,&97
2670 DATA &C3,&B7,&FE,&01,&26,&09,&2E,&03, [1570]
&28,&0A,&26
2730 DATA &00,&19,&CD,&06,&B9,&F5,&E5,&78, [1695]
&CD,&0C,&0D
2740 DATA &46,&79,&CD,&0C,&0D,&4E,&E1,&70, [2118]
&11,&10,&00
2810 DATA &3E,&15,&CD,&D0,&BD,&2A,&26,&B7, [3001]
&CD,&D2,&11
2820 DATA &22,&26,&B7,&D8,&E5,&21,&2D,&B7, [2420]
&78,&87,&3C
2830 DATA &86,&77,&CD,&4E,&12,&3A,&30,&B7, [1545]
&F5,&DC,&91
2880 DATA &67,&22,&26,&B7,&C3,&CD,&BD,&CD, [2666]
&06,&B9,&F5
2900 DATA &D0,&BD,&2A,&29,&B7,&22,&26,&B7, [2082]
&ED,&5B,&2B
2910 DATA &B7,&18,&4A,&CD,&D5,&BB,&B7,&ED, [3008]
&52,&23,&CD
2920 DATA &81,&8B,&CD,&81,&8B,&CB,&3D,&45, [1596]
&ED,&5B,&A1
2930 DATA &B6,&2A,&9F,&B6,&E5,&B7,&ED,&52, [2483]
&23,&4D,&ED
2940 DATA &5B,&9B,&B6,&E1,&C5,&CD,&1D,&BC, [1611]
&D1,&3A,&A4
2950 DATA &B6,&4F,&CD,&D3,&89,&CF,&11,&96, [2736]
&CB,&2C,&CB
2970 DATA &CD,&BD,&CD,&E1,&8A,&EB,&6B,&3A, [2469]
&2A,&B7,&67
2980 DATA &3A,&30,&B7,&18,&EC,&CD,&E1,&8A, [2000]
&5D,&3A,&2C
2990 DATA &B7,&57,&18,&F1,&CD,&91,&8B,&2A, [3030]
&29,&B7,&3A
3000 DATA &2C,&B7,&57,&3A,&26,&B7,&3D,&5F, [2744]
&BD,&3A,&30
3010 DATA &B7,&D2,&CF,&89,&C9,&CD,&9F,&8B, [3041]
&2A,&29,&B7
3020 DATA &ED,&5B,&2B,&B7,&3A,&26,&B7,&6F, [2522]
&2C,&BB,&3A
3030 DATA &30,&B7,&DA,&CF,&89,&C9,&0F,&9F, [1199]
&C3,&9F,&BB
3050 DATA &C6,&B7,&2A,&C4,&B7,&22,&0C,&8C, [2557]
&57,&D5,&C5
```

Listing "Screen Pack Plus"

```
3070 DATA &80,&5F,&7E,&23,&BA,&28,&13,&32, [1827]
&C6,&B7,&D5
3080 DATA &E5,&C5,&AF,&21,&00,&00,&22,&C4, [2225]
&B7,&CD,&1F
3090 DATA &8C,&C1,&E1,&D1,&1D,&20,&E5,&7A, [1731]
&32,&C6,&B7
3140 DATA &C4,&B7,&19,&7C,&E6,&07,&67,&3A, [2383]
&C6,&B7,&84
3150 DATA &67,&50,&1E,&08,&C3,&D3,&89,&2A, [2329]
&C4,&B7,&19
3160 DATA &C3,&05,&BC,&00,&00,&00,&4F,&3A, [1950]
&33,&B7,&B7
3170 DATA &79,&C2,&3C,&19,&21,&58,&B7,&46, [1942]
&78,&FE,&0A
3180 DATA &D2,&40,&14,&B7,&C2,&21,&14,&79, [2572]
&FE,&20,&30
3190 DATA &03,&C3,&21,&14,&47,&3A,&2E,&B7, [1637]
&CB,&47,&C0
3200 DATA &C5,&CD,&E1,&8A,&24,&22,&26,&B7, [2329]
&25,&F1,&CD
3210 DATA &D3,&BD,&C3,&CD,&BD,&3A,&2E,&B7, [2836]
&E6,&7F,&B7
3240 DATA &8A,&ED,&4B,&2F,&B7,&CD,&E1,&0D, [2786]
&E1,&D1,&C1
3270 DATA &C6,&B7,&F5,&21,&EB,&BD,&7E,&F5, [1934]
&E5,&3E,&C9
3280 DATA &77,&C5,&CD,&FC,&81,&3A,&27,&8D, [1639]
&B7,&3A,&C6
3290 DATA &B7,&57,&C4,&C9,&85,&C1,&3A,&89, [1905]
&80,&3D,&5F
3530 DATA &C3,&B7,&07,&07,&EB,&86,&CD,&12, [2420]
&8E,&D1,&E1
```

```
50 'Version 1.3 (fuer Schneider/ Amstrad C [3417]
PC 6128!) -----
240 SAVE"spp-6128",b,32768,4534            [1168]
390 DATA &DF,&7D,&80,&C9,&72,&DD,&FD,&00,& [2225]
01,&C0,&40
410 DATA &81,&B7,&CA,&8D,&81,&F5,&3A,&C3,& [1621]
B7,&32,&99
590 DATA &71,&18,&06,&B7,&3A,&C3,&B7,&18,& [2689]
E3,&CD,&0B
710 DATA &81,&C9,&00,&CD,&0B,&BC,&32,&FB,& [2238]
81,&21,&F9
720 DATA &B7,&E5,&CD,&DD,&BC,&11,&30,&82,& [2600]
06,&81,&E1
740 DATA &BC,&00,&C9,&21,&F9,&B7,&E5,&CD,& [2762]
DD,&BC,&3A
770 DATA &ED,&79,&FB,&11,&61,&0D,&06,&81,& [1555]
E1,&3E,&C9
860 DATA &F8,&B7,&35,&20,&12,&CD,&C5,&82,& [2362]
32,&F8,&B7
870 DATA &21,&F6,&B7,&7E,&2F,&77,&AF,&32,& [2220]
F7,&B7,&18
890 DATA &3A,&F6,&B7,&B7,&21,&10,&00,&20,& [1485]
02,&19,&EB
920 DATA &7D,&B9,&20,&EA,&C9,&3A,&F6,&B7,& [2796]
B7,&3A,&D3
930 DATA &B7,&C8,&3A,&D2,&B7,&C9,&02,&02,& [2358]
14,&16,&15
1540 DATA &C3,&B7,&FE,&01,&0E,&03,&28,&06, [1744]
&0E,&01,&30
1740 DATA &22,&31,&B7,&C9,&B7,&21,&6F,&86, [2439]
&28,&E7,&21
1750 DATA &CF,&8D,&18,&E2,&00,&00,&00,&00, [1838]
&2A,&2F,&B7
1770 DATA &C3,&6F,&86,&3A,&2F,&B7,&18,&F6, [2472]
&7E,&A8,&B1
1880 DATA &4F,&3A,&C3,&B7,&FE,&01,&3E,&03, [1960]
&28,&06,&3E
1910 DATA &00,&00,&57,&3A,&C3,&B7,&FE,&01, [1353]
&7A,&D0,&0F
2510 DATA &C6,&B7,&ED,&53,&B6,&89,&57,&3A, [1954]
&36,&82,&5F
2530 DATA &17,&32,&C6,&B7,&E3,&E5,&D5,&C5, [2033]
&3E,&00,&11
2550 DATA &B4,&89,&1D,&20,&E1,&7A,&32,&C6, [1948]
&B7,&E1,&C9
2620 DATA &D5,&E5,&21,&00,&00,&18,&D8,&79, [2372]
&18,&F0,&9B
2670 DATA &C3,&B7,&FE,&01,&26,&09,&2E,&03, [1570]
&28,&0A,&26
2730 DATA &00,&19,&CD,&06,&B9,&F5,&E5,&78, [1678]
&CD,&10,&0D
2740 DATA &46,&79,&CD,&10,&0D,&4E,&E1,&70, [2284]
&11,&10,&00
```

Listing "Screen Pack Plus"

```
2810 DATA &42,&15,&CD,&D0,&BD,&2A,&26,&B7, [2583]
&CD,&D6,&11
2820 DATA &22,&26,&B7,&D8,&E5,&21,&2D,&B7, [2420]
&78,&87,&3C
2830 DATA &86,&77,&CD,&52,&12,&3A,&30,&B7, [1090]
&F5,&DC,&91
2880 DATA &67,&22,&26,&B7,&C3,&CD,&BD,&CD, [2666]
&06,&B9,&F5
2900 DATA &D0,&BD,&2A,&29,&B7,&22,&26,&B7, [2082]
&ED,&5B,&2B
2910 DATA &B7,&18,&4A,&CD,&D5,&BB,&B7,&ED, [3008]
&52,&23,&CD
2920 DATA &81,&8B,&CD,&81,&8B,&CB,&3D,&45, [1596]
&ED,&5B,&A1
2930 DATA &B6,&2A,&9F,&B6,&E5,&B7,&ED,&52, [2483]
&23,&4D,&ED
2940 DATA &5B,&9B,&B6,&E1,&C5,&CD,&1D,&BC, [1611]
&D1,&3A,&A4
2950 DATA &B6,&4F,&CD,&D3,&89,&CF,&15,&96, [2724]
&CB,&2C,&CB
2970 DATA &CD,&BD,&CD,&E1,&8A,&EB,&6B,&3A, [2469]
&2A,&B7,&67
2980 DATA &3A,&30,&B7,&18,&EC,&CD,&E1,&8A, [2000]
&5D,&3A,&2C
2990 DATA &B7,&57,&18,&F1,&CD,&91,&8B,&2A, [3030]
&29,&B7,&3A
3000 DATA &2C,&B7,&57,&3A,&26,&B7,&3D,&5F, [2744]
&BD,&3A,&30
3010 DATA &B7,&D2,&CF,&89,&C9,&CD,&9F,&8B, [3041]
&2A,&29,&B7
3020 DATA &ED,&5B,&2B,&B7,&3A,&26,&B7,&6F, [2522]
&2C,&BB,&3A
3030 DATA &30,&B7,&DA,&CF,&89,&C9,&0F,&9F, [1199]
&C3,&9F,&BB
3050 DATA &C6,&B7,&2A,&C4,&B7,&22,&0C,&8C, [2557]
&57,&D5,&C5
3070 DATA &80,&5F,&7E,&23,&BA,&28,&13,&32, [1827]
&C6,&B7,&D5
3080 DATA &E5,&C5,&AF,&21,&00,&00,&22,&C4, [2225]
&B7,&CD,&1F
3090 DATA &8C,&C1,&E1,&D1,&1D,&20,&E5,&7A, [1731]
&32,&C6,&B7
3140 DATA &C4,&B7,&19,&7C,&E6,&07,&67,&3A, [2383]
&C6,&B7,&84
3150 DATA &67,&50,&1E,&08,&C3,&D3,&89,&2A, [2329]
&C4,&B7,&19
3160 DATA &C3,&05,&BC,&00,&00,&00,&4F,&3A, [1950]
&33,&B7,&B7
3170 DATA &79,&C2,&40,&19,&21,&58,&B7,&46, [2533]
&78,&FE,&0A
3180 DATA &D2,&44,&14,&B7,&C2,&25,&14,&79, [2077]
&FE,&20,&30
3190 DATA &03,&C3,&25,&14,&47,&3A,&2E,&B7, [2590]
&CB,&47,&C0
3200 DATA &C5,&CD,&E1,&8A,&24,&22,&26,&B7, [2329]
&25,&F1,&CD
3210 DATA &D3,&BD,&C3,&CD,&BD,&3A,&2E,&B7, [2836]
&E6,&7F,&B7
3240 DATA &8A,&ED,&4B,&2F,&B7,&CD,&E5,&0D, [2306]
&E1,&D1,&C1
3270 DATA &C6,&B7,&F5,&21,&EB,&BD,&7E,&F5, [1934]
&E5,&3E,&C9
3280 DATA &77,&C5,&CD,&FC,&81,&3A,&27,&8D, [1639]
&B7,&3A,&C6
3290 DATA &B7,&57,&C4,&C9,&85,&C1,&3A,&89, [1905]
&80,&3D,&5F
3530 DATA &C3,&B7,&07,&07,&EB,&86,&CD,&12, [2420]
&8E,&D1,&E1
```

```
10 '////////////////////////////////////// [2260]
//////////
20 '/                                      [200]
      /
30 '/ Demo-Programm zur Betriebssystem-Erw [3259]
eiterung /
40 '/                                      [200]
      /
50 '/      S C R E E N   P A C K   P L U  [2168]
S     /
60 '/                                      [200]
      /
70 '/                    8/1987 by Klaus Kr [3153]
emer  /
80 '/                                      [200]
      /
90 '////////////////////////////////////// [2260]
//////////
```

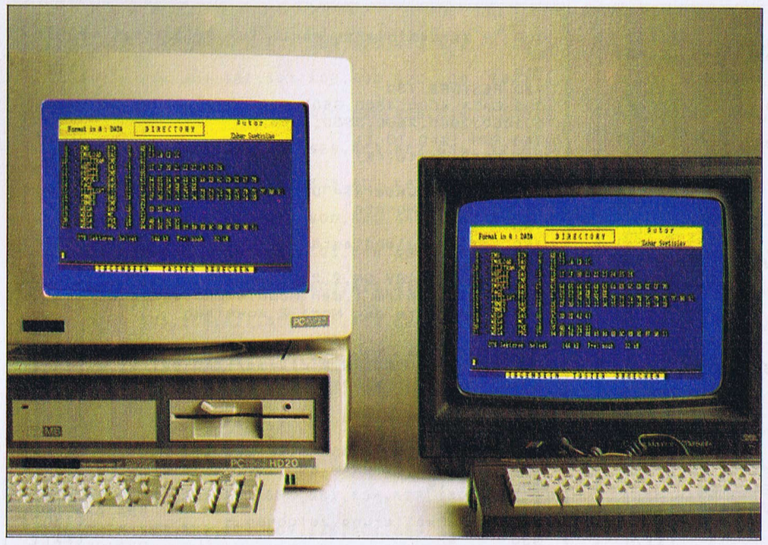
Listing "Screen Pack Plus"


```
100 '                                      [117]
110 'Screen Pack +  muss geladen und inita [2448]
llisiert sein!
120 '                                      [117]
130 MODE 1:INK 0,0:BORDER 0:INK 1,9        [2246]
140 '                                      [117]
150 INPUT "Welche Version soll vorgefuehrt [4316]
 werden  (1 oder 2)";ant$
160 IF ant$="2" THEN 440                   [87]
170 '                                      [117]
180 '/// Demo fuer Version 1 ///           [1245]
190 '                                      [117]
200 '- Demo fuer Mode 2 -                  [1204]
210 ¦BV.OFF:¦BV.ON,1,2:INK 0,0,0:INK 1,2,2 [3419]
:INK 2,24,24
220 FOR i%=1 TO 23 STEP 2                  [1372]
230 LOCATE 1,i%:PEN 1:PRINT"zweite Farbe": [4894]
PEN 2:PRINT"dritte Farbe";
240 NEXT                                   [350]
250 fmax%=2:d%=0:GOSUB 400:d%=1:GOSUB 400: [2865]
FOR i%=0 TO 5000:NEXT
260 '                                      [117]
270 '- Demo fuer Mode 1 -                  [1210]
280 ¦BV.OFF:¦BV.ON,1,1:FOR i%=0 TO 7:INK i [3548]
%,i%*2,i%*2:NEXT
290 FOR i2%=1 TO 4:FOR i%=1 TO 6:PEN i%:PR [3584]
INT STR$(i%+1);". Farbe":NEXT i%:NEXT i2%
300 fmax%=6:d%=0:GOSUB 400:FOR i%=0 TO 500 [2134]
0:NEXT
310 '                                      [117]
320 '- Demo fuer Mode 0 -                  [1208]
330 ¦BV.OFF:¦BV.ON,1,0:FOR i%=0 TO 26:¦INK [3247]
,i%,i%,i%:NEXT:FOR i%=27 TO 30:¦INK,i%,i%,
i%-16:NEXT
340 FOR i%=1 TO 9 STEP 2:¦PEN,i%:PRINT STR [5347]
$(i%+1);".+";:¦PEN,i%+1:PRINT STR$(i%+2);"
.F.":NEXT i%
350 FOR i%=11 TO 30:¦PEN,i%:PRINT STR$(i%+ [3112]
1);".Farbe":NEXT i%:LOCATE 1,1
360 fmax%=30:d%=0:GOSUB 400:FOR i%=0 TO 50 [2836]
00:NEXT
370 ¦BV.OFF:RUN                            [1230]
380 '                                      [117]
390 '- Linien zeichnen -                   [306]
400 fr%=0:FOR i%=1 TO 31:IF fr%>fmax%-1 TH [3303]
EN fr%=0
410 fr%=fr%+1:¦GPEN,fr%:MOVE 300+i%*10+d%, [3895]
0:DRAW 300+i%*10+d%,399:NEXT
420 RETURN                                 [555]
430 '                                      [117]
440 '/// Demo fuer Version 2 ///           [1320]
450 '                                      [117]
460 MODE 1                                 [506]
470 DIM werte(11)                          [1051]
480 ueb$="Saeulendiagramm":LOCATE 20-(LEN( [3990]
ueb$)/2),2:PRINT ueb$
490 LOCATE 1,6:INPUT "Moechten Sie eigene  [4161]
Werte eingeben      (j/n)";ant$
500 IF LOWER$(ant$)="j" THEN 520 ELSE 700  [1264]
510 '                                      [117]
520 '- Werte ueber den Bildschirm abfragen [1273]
 -
530 '                                      [117]
540 MODE 1:LOCATE 20-(LEN(ueb$)/2),2:PRINT [3766]
 ueb$
550 LOCATE 1,4:PRINT"Bitte geben Sie die W [6067]
erte fuer die      folgenden Monate ein"
560 DATA Januar,Februar,Maerz,April,Mai,Ju [5834]
ni,Juli,August,September,Oktober,November,
Dezember
570 PRINT:INPUT "Einheit (z.B. 'DM')";skx$ [6307]
:PRINT:PRINT "maximal 5 Vorkomma-Stellen!"
:PRINT
580 '                                      [117]
590 RESTORE 560:FOR i%=1 TO 12             [1461]
600 READ mo$                               [440]
610 PRINT USING "\          \";mo$;:INPUT  [3571]
werte(i%-1)
620 IF werte(i%-1)=0 THEN PRINT CHR$(11);C [5651]
HR$(7);STRING$(39," ");CHR$(13);CHR$(11):G
OTO 610
630 IF werte(i%-1)>x THEN x=werte(i%-1)    [2698]
640 NEXT                                   [350]
650 '                                      [117]
660 PRINT:PRINT"Diagramm-Ueberschrift (max [4718]
. 16 Z.)":INPUT "";uebers$
670 ma=INT(x):sky$="Monate"                [2565]
680 GOTO 790                               [415]
690 '                                      [117]
```

Listing "Screen Pack Plus"

```
700 '- Vordefinierte Werte zur Demonstrati [3036]
on -
710 '                                      [117]
720 RESTORE 730                            [729]
730 DATA 4000,4500,3600,3300,3000,1500,100 [2909]
0,2000,1000,2500,3500,3800
740 FOR i%=0 TO 11                         [804]
750 READ werte(i%)                         [1013]
760 NEXT i%                                [457]
770 ma=6000:uebers$="Umsaetze in '86":skx$ [4571]
="DM":sky$="Monate"
780 '                                      [117]
790 '/// Saeulendiagramm erstellen ///     [2222]
800 '                                      [117]
810 ¦BV.OFF:¦BV.ON,2,2,0'Version 2 einscha [6267]
lten; Bildschirm 1 den Mode 0 und dem zwei
ten Bildschirm den Mode 2 zuweisen
820 ¦TRANSP.,0                             [741]
830 '                                      [117]
840 CLS:¦BILDSCHIRM,2:div=ma/(350-80):'- S [3130]
kala in MODE 2 erstellen -
850 '                                      [117]
860 '- Y-Skalierung zeichnen -             [1400]
870 '                                      [117]
880 FOR i%=107 TO 350 STEP 27              [1729]
890 MOVE 102,i%:DRAW 600,i%,1              [1332]
900 NEXT i%                                [457]
910 '                                      [117]
920 '- X-Skalierung zeichnen -             [1961]
930 '                                      [117]
940 FOR i%=120 TO 560 STEP 40              [927]
950 MOVE i%,80                             [605]
960 DRAW i%,350                            [311]
970 NEXT i%                                [457]
980 '                                      [117]
990 MOVE 120,80:DRAW 600,80,1              [1701]
1000 '                                     [117]
1010 '- X-Achse mit Monatsnamen beschrifte [2886]
n -
1020 '                                     [117]
1030 LOCATE 14,22:PRINT"Jan. Feb. Maerz Ap [4745]
ril Mai  Juni Juli Aug. Sep. Okt. Nov. Dez
."
1040 '                                     [117]
1050 '- Sauelen einzeichnen -              [2080]
1060 '                                     [117]
1070 ¦BILDSCHIRM,1                         [1113]
1080 FOR i%=0 TO 11                        [804]
1090 yw=(werte(i%)/div)-2:INK i%+1,i%*2+1  [3352]
1100 i2%=120+i%*40:ORIGIN 0,0,i2%,i2%+30,9 [6593]
0+yw,82:CLG 0:ORIGIN 0,0,i2%,i2%+20,80+yw,
82:CLG i%+1:NEXT
1110 '                                     [117]
1120 ¦BILDSCHIRM,2:INK 1,6                 [1131]
1130 FOR i%=0 TO 11                        [804]
1140 yw=(werte(i%)/div)-2                  [2106]
1150 i2%=120+i%*40                         [1033]
1160 MOVE i2%,82:DRAW i2%+20,82,1:DRAW i2% [7097]
+30,90:DRAW i2%+30,90+yw:DRAW i2%+20,82+yw
:DRAW i2%,82+yw:DRAW i2%+10,90+yw:DRAW i2%
+30,90+yw:MOVE i2%+20,82+yw:DRAW i2%+20,82
1170 NEXT                                  [350]
1180 '                                     [117]
1190 '- Diagramm beschriften -             [2313]
1200 '                                     [117]
1210 l1=LEN(uebers$)                       [1219]
1220 l2=LEN(skx$)                          [1004]
1230 l3=LEN(sky$)                          [891]
1240 pu%=(10-INT(l1/2))+1                  [1480]
1250 pu2%=5-l2                             [866]
1260 pu3%=(40-INT(l3/2))+7                 [1458]
1270 LOCATE pu3%,24:PRINT sky$ '- Y-Einhei [2537]
t -
1280 LOCATE pu2%,3:PRINT skx$ '- X-Einheit [1979]
 -
1290 z%=0:TAG:FOR i%=88 TO 358 STEP 27     [1823]
1300 MOVE 40,i%:PRINT USING"#####.#";ma/10 [6122]
*z%:z%=z%+1'- X-Skala mit Zahlenwerte bes
chriften
1310 NEXT i%:TAGOFF                        [2114]
1320 '                                     [117]
1330 t$="Screen Pack Plus Demo - weiter mi [6255]
t einer beliebigen Taste":LOCATE (40-LEN(t
$)/2),25:PRINT t$
1340 ¦BILDSCHIRM,1:LOCATE pu%,2:PEN 13:INK [4027]
 13,24:PRINT uebers$
1350 '                                     [117]
1360 IF INKEY$="" THEN 1360 ELSE ¦BV.OFF:R [1730]
UN
```

Listing "Screen Pack Plus"



Von Rechner zu Rechner

FILE TRANSFER – Datentransport... unter anderem

Die Eingabe eines BASIC-Programmes beansprucht gewöhnlich viel Zeit, besonders wenn es sich um ein Programm handelt, das noch in der Entwicklungsphase steckt. Noch mehr Zeit wird in Anspruch genommen, wenn man den zugehörigen Text mit einer Textverarbeitung oder Daten in eine Datenbank eingibt. Falls man dies auf einem PC-Rechner am Arbeitsplatz macht, stellen oft die Arbeitszeit und die Inanspruchnahme des Rechners den "Engpass" dar, so daß es sehr günstig wäre, einen Teil dieser Arbeiten zuhause erledigen zu können und daß der Rechner am Arbeitsplatz nur für die Verarbeitung oder Anwendung dieser Files benutzt wird.

Für diejenigen, die glücklicherweise auch einen der PC- Rechner ihr Eigen nennen, gibt es da keine Schwierigkeit. Man kann aber mit den geeigneten Programmen auch den CPC 6128 für diesen Zweck benutzen. Die Voraussetzung dafür ist die Möglichkeit, daß der Rechner das PC-(IBM-)Format erkennt und daß man über ein zusätzliches 5.25"-Laufwerk verfügt. Diese Laufwerke sind seit geraumer Zeit immer öfter bei CPC-Besitzern anzutreffen, da die Preise dieser Laufwerke ziemlich heruntergesetzt (unter 400,– DM) wurden.

Das Programm, das hier präsentiert wird, gibt diese Möglichkeit des File-Austausches zwischen den Schneider CPC 6128-Rechnern oder PCW 8256/8256 (JOYCE) und IBM- kompatiblen Rechnern. Dabei wird die vollständige Kompatibilität nur bei den ASCII-Dateien erreicht, BASIC-Programme können also nur dann übertragen werden, wenn sie mit der Option "A" (auch im ASCII-Code) gespeichert worden sind. Dies bedeutet, daß diese Programme an einem Rechner eingelesen und an einem anderen getestet und gestartet werden können.

Das Programm wurde größtenteils in BASIC programmiert, bis auf einen kleinen Maschinenspracheteil, der zur Kontrolle des Laufwerkes benötigt wird. Alle Optionen in den benutzerorientierten Menüs können hauptsächlich über die Cursortasten, die Copy- und die Return-Taste aufgerufen werden. Das Hauptmenü setzt sich aus folgenden Menüpunkten zusammen:

1. – Ausdruck des Directory
2. – Ausdruck des Inhalts-Verzeichnisses eines Sektors auf der Diskette
3. – Kopieren der Files von einer Diskette auf die andere
4. – Korrigieren des Inhalts Verzeichnisses eines Sektors
5. – Löschen eines Eintrags vom Directory einer PC-Diskette.

Dabei werden die ersten vier Optionen auf eines der Formate bezogen, welches die CPC-Rechner erkennen können: DATA, CP/M, JOYCE oder IBM (Abb. 1).

Die Menüpunkte

Menüpunkt 1: DIRECTORY
Dieser Programmteil ermöglicht den Ausdruck des Directories von Diskette A oder B auf Bildschirm oder Drucker. Dabei kann man ein Grundinhaltsverzeichnis anzeigen lassen, in dem nur die Dateinamen und die Dateilänge enthalten sind, oder ein vollständiges Inhaltsverzeichnis, in dem neben den obigen Angaben noch die Sektoren auf der Diskette, wo die Dateien gespeichert worden sind, angegeben werden. Dies ist von besonderer Bedeutung bei den CPC-Disketten, bei denen eine Datei in mehreren Teilen auf den Disketten gespeichert werden kann. Falls der Diskettenname auf der Diskette vorhanden ist, wird dieser Name auch beim Druck angezeigt, ansonsten fordert das Programm vom Benutzer, daß er den Namen der Diskette einträgt.

2.- HEXDUMP
Nach der Auswahl dieser Option fordert das Programm vom Benutzer, daß er definiert, welches Laufwerk benutzt wird, welche Datei ausgewählt und welcher Dateiteil gedruckt werden soll und in welcher Form man sich die Ausgabe wünscht; nur hexadezimal, nur ASCII oder beides. Wenn man sich ein vollständiges Hexdump wünscht, stehen zwei Möglichkeiten zur Verfügung: Druck von ASCII-Code unter dem hexadezimalen Code, oder seitlich. Am Anfang jeder Zeile wird eine relative Adresse vom Anfang dieser Datei ausgegeben.
Auch hier, wie bei der vorherigen Option, kann man mit einem der vorhin aufgezeigten Formate arbeiten.

3.- COPY CPC - JOYCE - PC
Dieser Programmteil ist wohl der interessanteste, wie auch schon am Anfang erwähnt, denn er ermöglicht das Dateikopieren von einer Diskette auf die andere, ohne Rücksicht auf das Format. Hierbei sei allerdings gesagt, ohne näher auf Erklärungen zur Formatstruktur eingehen zu wollen, daß man mit dem Programm die Arbeit vom PC nur simuliert; alle Angaben, die für die Erkennung der Datei seitens des IBM PC- oder eines kompatiblen Rechners nötig sind, werden in die entsprechenden Sektoren eingetragen (es genügt also nicht, nur den Dateinamen ins Directory einzutragen). Wie schon erwähnt, kann man neben der Übertragung von ASCII-Dateien auch im ASCII-Code vorher gespeicherte BASIC-Programme übertragen. Es wäre natürlich auch möglich, ein besonderes Programm zu erstellen, welches ein in einer BASIC-Version geschriebenes Programm in ein anderes transformieren könnte, wobei wir uns allerdings auf die Befehle, die in beiden Versionen vorhanden sind, beschränken müßten. Aber das ist ein anderes Thema.

4.- EDITIEREN
Nicht weniger nützlich (für die meisten vielleicht noch nützlicher) ist die Möglichkeit, auf die Inhalte der Sektoren einer Diskette zum Zweck der Überprüfung und Änderung zurückzugreifen, ganz egal, ob es sich um CPC-, JOYCE- oder PC- Disketten handelt. Wenn dieser Programmteil angesprochen wird, fordert das Programm, daß man die Diskette bestimmt und danach auch die Dateien und Sektoren, die kontrolliert werden sollen. Danach wird der Inhalt der ersten Hälfte des ausgewählten Sektors auf dem Bildschirm in zwei unabhängige Fenster ausgegeben:
links hexadezimale Codes,
rechts ASCII-Codes.
Der Cursor läßt sich in alle vier Richtungen, links, rechts, aufwärts, abwärts, mit den Cursortasten bewegen. Mit dem gleichzeitigen Drücken der SHIFT-Taste und einer der Cursortasten bewegt sich der Cursor zum Anfang oder zum Ende der Zeile oder des Fensters, und mit dem Drücken der CTRL-Taste und der Cursor-rechts- oder Cursor-links-Taste bewegt man sich von einem Fenster zum anderen. Um die andere Hälfte des Sektors einzulesen, drücken Sie die Cursor-auf- oder Cursor1/2Taste zusammen mit der CTRL-Taste. Mit RETURN wird der EDIT-Modus verlassen und das Programm fragt, ob man diesen veränderten Sektor auf Diskette speichern will (natürlich an derselben Stelle, wo er vorher war) um danach den nächsten ausgewählten Sektor der gleichen oder nächsten Datei einzulesen.

5.- LÖSCHEN DER PC-DISKETTE
Dieser Programmteil wird benutzt, wenn man den Inhalt der PC-Diskette löschen will. Man kann den auf dem Originalrechner eingegebenen Inhalt löschen, oder auch vorher auf dem CPC-Rechner erstellte und von dort kopierte Dateien. Falls keine Diskette vom IBM-Format in Laufwerk B vorhanden ist, meldet das Programm, daß es sich um ein unkorrektes Format handelt.
Hierzu gehört auch, daß die PC- Diskette, die man für den Transfer oder für die Änderung des Dateiinhaltes benutzen will, vorher auf einem PC-Rechner formatiert werden muß. Es reicht eine einseitige Formatierung aus, da die meisten 5 1/4 "-Laufwerke für die CPCs ebenfalls nur eine Seite der Diskette beschreiben können. Diese Option ist schon bei den IBM- und kompatiblen PC-Rechnern vorgesehen, indem man die Option FORMAT/1 benutzt.

Das Programm besteht, wie schon gesagt, aus zwei Teilen: BASIC- und Maschinenspracheteil. Zuerst geben Sie das BASIC-Programm ein und speichern es auf Diskette ab, danach den DATA-Lader, den Sie ebenfalls aus Sicherheitsgründen erst einmal abspeichern. Danach starten Sie den Lader, er erzeugt den Maschinenspracheteil, der automatisch auf Diskette abgelegt wird.

(Milorad Zahar/jb)

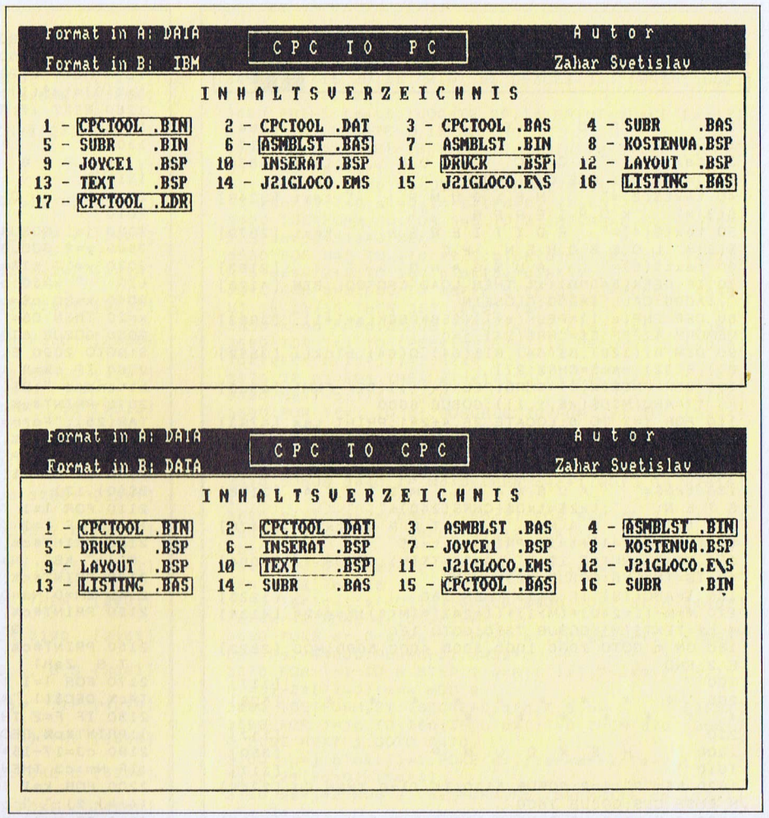


Abb. 1: Ob IBM-(PC-)Format oder CPC-Dataformat, alle Files werden übersichtlich aufgezeigt und gekennzeichnet.

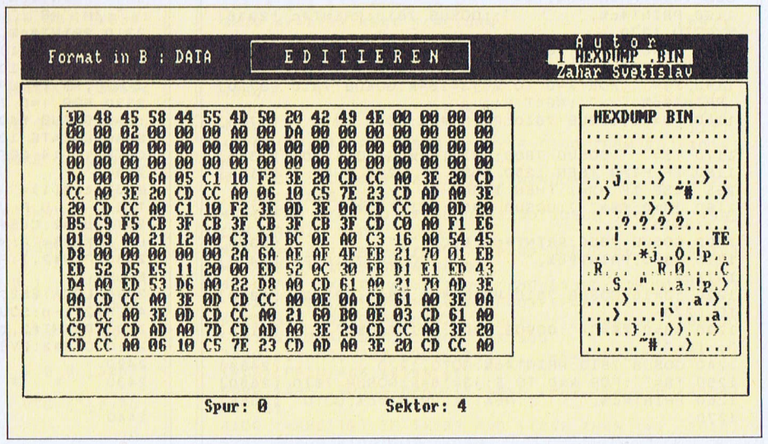
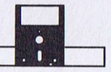


Abb. 2: Anhand eines Hexdumps eines jeden Sektors lassen sich direkte Manipulationen an den abgespeicherten Bytes vornehmen.

für 6128



```
10 '    C P C      T O O L                [364]
20 '                                       [117]
30 text$(0)="   C P C    T O O L    ":text [3636]
$(1)="   D I R E C T O R Y    "
40 text$(2)="      H E X D U M P      ":text [3349]
$(3)="    K O P I E R E N     "
50 text$(4)="   E D I T I E R E N    ":text [2879]
$(5)=" L O E S C H E N   P C "
60 text$(6)="     A U S G A N G      "      [2163]
70 IF PEEK(&A000)<>1 THEN LOAD"CPCTOOL.BIN [4128]
",&A000:CALL &A000:CLOSEIN
80 DEF FNP(a,l)=PEEK(a+l)+256*PEEK(a+l+1): [3983]
MEMORY &7FFF:E$=CHR$(24):j=0
90 DIM B1(128),B$(64),B1$(64),Q(64),blok(1 [3552]
80),F(32):esc$=CHR$(27)
100 DEF FNS$(x,i)=CHR$(PEEK(x+i)):DEF FNM( [4533]
x$,i)=ASC(MID$(x$,i,1)):GOSUB 6000
110 FOR j=1 TO 6:LOCATE 25,4+2*j:PRINT tex [4793]
t$(j):MOVE 188,269-J*32:DRAWR 190,0,1
120 DRAWR 0,22:DRAWR -190,0,1:DRAWR 0,-22: [3251]
NEXT J:M=1:X=25:Y=6:LOCATE#2,1,1
130 tx0$="    A U S W A H L   M I T   T A [3198]
S T E N:    ":tx2$=tx0$+CHR$(240)+"  "
140 tx1$=",  D A N N    R E T U R N":tx2$= [3470]
tx2$+CHR$(241)+tx1$:PRINT#2,tx2$
150 I$="":WHILE I$="":p$=TEXT$(M):GOSUB 72 [4064]
40:I$=INKEY$:GOSUB 7250:WEND
160 I=ASC(I$):IF I=13 THEN 180             [1225]
170 M=M-(I=240)*(M>1)+(I=241)*(M<6):Y=4+2* [4134]
M:p$=TEXT$(M):GOSUB 7250:GOTO 150
180 ON m GOTO 2000,1000,3000,4000,5000:MOD [2822]
E 2:END
190 '                                      [117]
200 ' *   *   *   *   *   *   *   *   *    [528]
*   *   *   *   *   *   *
210 '                                      [117]
1000 '    H E X D U M P                    [350]
1010 '                                     [117]
1020 S$="P":j=2:GOSUB 5110:IF f1=0 THEN RU [2746]
N ELSE CLS:GOSUB 7600
1030 CLS:H1=0:H2=0:H3=0:h1$=" N E I N ":h2 [2520]
$="   J A   "
1040 GOSUB 7920:IF H2*H1=0 THEN 1060 ELSE  [3424]
H2=0:h1$="  UNTEN  ":h2$=" RECHTS  "
1050 y=15:GOSUB 7950:H3=H2:H2=1            [1240]
1060 IF c%+H1=0 THEN G=64 ELSE IF c%+H1=8  [4066]
THEN G=96 ELSE IF c%+H3=9 THEN G=16
1070 FOR lf=1 TO pf:GOSUB 5130:GOSUB 7500  [1845]
1080 CLS:FOR ls=1 TO pl:l=B1(Q(ls)):T=l\9- [2449]
2*(F=0)-(F=3)
1082 S=l MOD 9:ad=(Q(ls)-1)*512            [1339]
1090 ¦SECREAD,d,T,S,&A300:PRINT#c1%,TAB(29 [4709]
);"Spur:";T;" Sektor:";S;:PRINT#c1%
1100 FOR a=&A300 TO &A4FF STEP G:PRINT#c%, [3657]
HEX$(ad,4);": ";:ad=ad+G:j3=0
1110 IF c%=8 THEN 1180 ELSE IF H1=0 THEN 1 [3621]
150 ELSE IF H2=0 THEN 1170
1120 IF H3=1 THEN 1140 ELSE j3=0:f$=" ":GO [3866]
SUB 7800:PRINT#c%:f$=" "
1130 PRINT#c%,"      ";:GOSUB 7810:PRINT#c [2938]
%:GOTO 1270
1140 j3=0:f$=" ":GOSUB 7800:f$="":PRINT#c% [5267]
,"   ";:GOSUB 7810:PRINT#c%:GOTO 1270
1150 f$="":FOR k=0 TO 2:j3=16*k:GOSUB 7810 [3730]
:PRINT#c%,"   ";:NEXT k:
1160 j3=48:GOSUB 7810:PRINT#c%:GOTO 1270   [2728]

1170 f$=" ":GOSUB 7800:PRINT#c%:GOTO 1270  [1882]
1180 IF H1=0 THEN 1250 ELSE IF H2=0 THEN 1 [3155]
265 ELSE IF H3=1 THEN 1230
1190 j3=0:f$=" ":GOSUB 7800:PRINT#c%,"     [3157]
";:j3=16:
1200 GOSUB 7800:PRINT#c%,ESC$;"0"          [2308]
1210 j3=0:PRINT#c%,"      ";:GOSUB 7810:PR [3470]
INT#c%,"   ";
1220 j3=16:GOSUB 7810:PRINT#c%,ESC$;"2":GO [2843]
TO 1270
1230 j3=0:f$=" ":GOSUB 7800:PRINT#c%,"     [3124]
";:f$=""
1240 GOSUB 7810:PRINT#c%:GOTO 1270         [2484]
1250 f$="":FOR k=0 TO 5:j3=16*k:GOSUB 7810 [2650]
1260 PRINT#c%,"   ";:NEXT k:PRINT#c%:GOTO  [2368]
1270
```

Listing "FILE TRANSFER"

```
1265 f$=" ":GOSUB 7800:PRINT#c%,"    ";:j3 [2987]
=16:GOSUB 7800:PRINT#c%
1270 NEXT a:PRINT#c%:NEXT ls:IF c%=0 THEN  [3219]
tx$(0)=tx$(1):GOSUB 2360
1280 NEXT lf:RUN                           [443]
1290 '                                     [117]
1300 '   *   *   *   *   *   *   *   *   * [528]
   *   *   *   *   *   *   *
1310 '                                     [117]
2000 '      D I R E C T O R Y              [889]
2010 '                                     [117]
2020 j=1:GOSUB 6000:GOSUB 6100:ZM=0:GOSUB  [3445]
7600:y=2:GOSUB 6400
2030 y=12:p1$="  GRUNDFORM  ":p2$="   VOEL [4259]
LIG   ":x=50:p$=p2$:GOSUB 6210
2040 x=20:p$=p1$:GOSUB 6210:GOSUB 7620:IF  [3772]
x=20 THEN S$="O" ELSE S$="P"
2050 GOSUB 6700:GOSUB 6900:IF c%=0 THEN CL [2605]
S:GOTO 2090 ELSE LOCATE 20,14
2060 IF c%=8 AND B$(0)="" THEN INPUT "Disk [2684]
ettename ";B$(0)
2070 PRINT#c%,q$;"Name :";LEFT$(B$(0),12); [4619]
TAB(25);"Format: ";form$(F)
2080 PRINT#c%,q$;STRING$(20,"="):PRINT#c%  [2067]
2090 IF S$="P" THEN 2140 ELSE ZONE 24      [1099]
2100 IF LEFT$(B$(0),12)>" " THEN n$=LEFT$( [1485]
B$(0),12)
2110 FOR l=1 TO fl:Nb=FNM(B$(l),14):Nm=(Nb [2864]
+1)\2:IF F=2 THEN Nm=0.5*Nb
2120 PRINT#c%,LEFT$(B$(l),12)+DEC$(Nm,"### [5412]
.#")+" kB",:Na=Na+Nb:ZM=ZM+Nm:NEXT l
2130 PRINT#c%:PRINT#c%:GOTO 2300           [1906]
2140 ZM=0:Na=0:jp=1:jf=0: IF F=2 THEN 2160 [2934]
2150 PRINT#c%,"    Dateiname      Sektoren [3043]
          Gruppen":GOTO 2170
2160 PRINT#c%,"    Dateiname      Laenge   [4302]
  T.S  Zahl      Datum":PRINT#c%
2170 FOR l=1 TO fl:F$=LEFT$(B$(l),12):PRIN [3610]
T#c%,DEC$(l,"##.");" - ";F$;TAB(20);
2180 IF F=2 THEN 2230 ELSE Nb=FNM(B$(l),14 [3992]
):PRINT#c%,DEC$(Nb,"####");"    ";
2190 c3=17-15*(c%=8):Ds=512*Nb:Nm=(Nb+1)\2 [3165]
:IF Nm>c3 THEN N=c3 ELSE N=Nm
2200 FOR k=1 TO N:PRINT#c%,HEX$(FNM(B$(l), [4397]
14+k),2);" ";:NEXT k
2210 IF Nm<=c3 THEN 2290 ELSE PRINT#c%:PRI [3159]
NT#c%,TAB(27);:FOR k=c3+1 TO Nm
2220 PRINT#c%,HEX$(FNM(B$(l),14+k),2);" "; [2074]
:NEXT k:GOTO 2290
2230 Ds=64*FNM(B$(l),21)+(256*FNM(B$(l),20 [2538]
)+FNM(B$(l),19))/1024
2240 Dt=FNM(B$(l),15)+256*FNM(B$(l),16):St [5126]
=256*FNM(B$(l),18)+FNM(B$(l),17)
2250 Nb=FNM(B$(l),14):Nm=0.5*Nb:TS=(St+7)\ [4236]
9+((St+7) MOD 9)/10:G=Dt\512
2260 M=(Dt-512*G)\32:Dn=Dt-512*G-32*M:PRIN [2328]
T#c%,DEC$(Ds,"####.#");
2270 PRINT#c%,DEC$(TS,"#####.#");DEC$(Nb," [1911]
#####");
2280 PRINT#c%,DEC$(Dn,"######");"-";DEC$(M [2598]
,"##");"-";DEC$(G+1980,"####");
2290 Na=Na+Nb:ZM=ZM+Nm:PRINT#c%:NEXT l:PRI [2742]
NT#c%
2300 PRINT#c%,"    ";Na"Sektoren  belegt   [4114]
  ";ZM;"kB";"   Frei noch   ";
2310 PRINT#c%,UM-ZM;"kB":PRINT#c%:PRINT#c% [4656]
:tx$(0)=tx$(1):GOSUB 2360:CLS
2320 LOCATE 10,10:INPUT "Andere Diskette ( [4752]
J/N)";x$:IF UPPER$(x$)<>"J" THEN RUN
2330 FOR l=0 TO 64:B$(l)="":NEXT:Na=0:ZM=0 [4014]
:y=2:GOSUB 6400:¦SETFORM,F
2340 LOCATE 10,14:PRINT STRING$(30,32);:LO [3153]
CATE 10,14:GOTO 2050
2350 '                                     [117]
2360 tx$(1)=" I R G E N D E I N E   T A S  [3356]
T E    D R U E C K E N  "
2370 p1=2:CLS#2:EVERY  30,1 GOSUB 2390:WHI [2062]
LE INKEY$="":WEND
2380 PEN#2,1:PAPER#2,0:CLS#2:a=REMAIN(1):R [3302]
ETURN
2390 LOCATE#2,13,1:p=(p+1) AND 1:PEN#2,p:P [2999]
APER#2,1-p:SOUND p1,60,2
2400 PRINT#2,tx$(p);:RETURN                [1830]
2410 CLS#2:EVERY  40,1 GOSUB 2390:RETURN   [2080]
2420 '                                     [117]
2430 '  *   *   *   *   *   *   *   *   *  [528]
   *   *   *   *   *   *   *
2440 '                                     [117]
```

Listing "FILE TRANSFER"


```
3000 '        C O P Y   C P C - P C        [966]
3010 '                                     [117]
3020 j=3:S$="P":GOSUB 6000:GOSUB 6320      [1977]
3030 CLS:y=1:d$="A":GOSUB 6400:FA=F:sr0=sr [4494]
:ip0=ip:y=3:d$="B":GOSUB 6400
3040 FB=F:sr1=sr:ip1=ip:tx3$=tx0$+CHR$(242 [4227]
)+", "+CHR$(243)+tx1$
3050 y=10:p2$="   A  "+CHR$(243)+"  B   ": [3204]
p1$="   B  "+CHR$(243)+"  A   "
3060 p$=p2$:x=50:GOSUB 6210:x=20:p$=p1$:GO [3974]
SUB 6210:GOSUB 7620:txt$=" C P C "
3070 IF Fb=2 THEN txt$="  P C  " ELSE IF F [2981]
b=3 THEN txt$=" JOYCE "
3075 IF Fa=3 THEN tx$=" JOYCE " ELSE tx$=" [1954]
 C P C "
3080 Fi=Fb:Fo=Fa:din=1:do=0:t0$=txt$:t1$=t [3202]
x$:sri=sr1:sro=sr0
3090 IF x=20 THEN 3100 ELSE Fi=Fa:Fo=Fb:di [2635]
n=0:do=1:t0$=tx$:t1$=txt$
3094 sri=sr0:sro=sr1: IF F=3 THEN t0$=" JO [4247]
YCE ":t1$=txt$
3100 LOCATE#1,28,2:PRINT#1,t0$;"  T O   ";t [2313]
1$
3110 ¦ERRORON:IF Fo<>2 THEN 3500           [3180]
3130 ¦SETFORM,Fi:T=-2*(Fi=0)-(Fi=3):u1=0:s [3570]
r=sri:F=Fi:GOSUB 6700:GOSUB 7000
3140 FOR lf=1 TO pf:GOSUB 5130:GOSUB 3780: [2367]
Ds=512*Nb
3150 ¦SETFORM,Fo:TB=0:H=&A400:¦SECREAD,1,0 [3995]
,1,H:k=0:M=3:c=&9000
3160 FOR i=3 TO 540 STEP 3:FOR j=0 TO 2:p( [2321]
j)=PEEK(H+i+j):NEXT j:M=i
3170 B=p(0)+256*(p(1) MOD 16):IF b=0 THEN  [2901]
3190 ELSE k=k+1:M=M+1
3180 B=16*p(2)+(p(1)\16):IF b=0 THEN 3190  [2569]
ELSE k=k+1:NEXT i:STOP
3190 St1=k+8:FOR SB=5 TO 8:¦SECREAD,1,TB,S [3301]
B,&8800+512*(SB-5):NEXT SB
3200 FOR b=&8800 TO &8FFF STEP 32:f$="":IF [4298]
 PEEK(b)=0 THEN 3230
3210 x0=b:GOSUB 7300:IF fl$=UPPER$(f$) THE [3888]
N 3450 ELSE NEXT b:GOTO 3460
3230 IF b>&8800 THEN 3250 ELSE St=8:POKE c [5234]
+11,32:FOR i=12 TO 21:POKE c+i,0
3240 NEXT:POKE c+22,64:POKE c+23,80:POKE c [4860]
+24,72:POKE c+25,14:GOTO 3260
3250 c=b-32:St=PEEK(c+26)+6+INT((FNP(c,28) [3244]
+511)/512)
3260 IF st<>st1 THEN STOP ELSE FOR i=0 TO  [4063]
7:POKE b+i,FNM(B$(r),1+i):NEXT i
3270 FOR i=8 TO 10:POKE b+i,FNM(B$(r),2+i) [3104]
:NEXT i:LOCATE 5,5
3280 PRINT"Insgesamt ";Nb;" Sektoren":LOCA [5130]
TE 5,7:PRINT"Laenge";Ds/1024;" kB"
3290 FOR i=11 TO 25:POKE b+i,PEEK(c+i):NEX [3921]
T i:POKE b+26,(st-6) MOD 256
3300 POKE b+27,(st-6)\256:POKE b+28,Ds-256 [2149]
*INT(Ds/256)
3310 POKE b+29,INT(Ds/256):POKE b+30,0:POK [1826]
E b+31,0
3320 FOR SB=5 TO 8:¦SECWRITE,1,TB,SB,&8800 [2540]
+512*(sb-5):NEXT SB
3330 FOR l=1 TO Nb:k=k+1:T=B1(l)\9-2*(Fi=0 [4238]
)-(Fi=3):S=B1(l) MOD 9:¦SETFORM,Fi
3340 ¦SECREAD,0,T,S,&9300:IF l<Nb THEN 337 [2638]
0
3350 FOR j=&9300 TO &94FF:IF PEEK(j)<>&1A  [1629]
THEN NEXT j
3360 IF j<&94FE THEN FOR j1=j+1 TO &94FF:P [3965]
OKE j1,0:NEXT j1
3370 rs=St+1:IF l<Nb THEN G=rs-6 ELSE G=&F [2247]
FF
3380 IF (k MOD 2)=0 THEN 3390 ELSE POKE M+ [2030]
H,G MOD 256
3385 M=M+1:POKE M+H,G\256:GOTO 3400        [1577]
3390 POKE M+H,PEEK(H+M)+16*(G MOD 16):M=M+ [2738]
1:POKE M+H,G\16:M=M+1
3400 TB=rs\9:SB=rs MOD 9:¦SETFORM,Fo:¦SECW [3676]
RITE,1,TB,SB,&9300
3410 NEXT l:¦SETFORM,Fo:¦SECWRITE,1,0,1,H  [2889]
3420 NEXT lf:RUN                           [443]
3430 '                                     [117]
3440 text$="  F A L S C H E S    F O R M A [3649]
 T  ":GOSUB 3480:STOP
3450 text$="  D A T E I    E X I S T I E R [3287]
 T  ":GOSUB 3480:GOSUB 7260:STOP
3460 text$="  D I R E C T O R Y      V O L [2720]
 L  ":GOSUB 3480:STOP
```

Listing "FILE TRANSFER"

```
3470 text$="  K E I N    P L A T Z    M E H [3627]
 R  ":GOSUB 3480:STOP
3480 LOCATE 20,16:PRINT e$;SPACE$(34);CHR$ [3602]
(7);:LOCATE 20,17
3490 PRINT text$;:LOCATE 20,18:PRINT;SPACE [4143]
$(34);e$;:LOCATE 1,15:RETURN
3500 ip=ipi:F=Fi:¦SETFORM,F:d=din:T=-2*(F= [3061]
0)-(F=3):sr=sri
3510 GOSUB 6700:GOSUB 7000:UM=178+9*(Fo=0) [4233]
+5*(Fo=3):l=0:sl=0:T=-2*(Fo=0)-(Fo=3)
3520 ¦SETFORM,Fo:FOR S=0 TO 3:¦SECREAD,do, [4169]
T,S,&8800+512*S:NEXT
3530 FOR b=0 TO 1+dr*9/2:blok(b)=1:NEXT:FO [4723]
R b=dr*9/2+2 TO 180:blok(b)=0:NEXT
3540 FOR b=&8800 TO &8FFF STEP 32:p1=PEEK( [4849]
b):IF p1>15 THEN 3570 ELSE f$=""
3550 x0=b:F=Fo:GOSUB 7300:l=l+1:B1$(l)=f$  [2381]
3560 FOR j=1 TO (PEEK(b+15)+7)\8:UM=UM-1:b [4026]
lok(PEEK(b+15+j))=1:NEXT j
3570 IF p1=&E5 THEN sl=sl+1:Q(sl)=b        [1551]
3580 NEXT b:l1=l:lg=0                      [1260]
3590 FOR lf=1 TO pf:GOSUB 5130:FOR j=1 TO  [3294]
l1:IF f1$=B1$(j) THEN 3450
3600 NEXT j:GOSUB 3780:IF Ds>UM THEN 3470  [3843]
ELSE UM=UM-INT(Ds+0.9):bw=&9300:c=0
3610 GOSUB 3640:IF Nb>0 THEN c=c+1:GOTO 36 [2818]
10
3620 NEXT:T=-2*(Fo=0)-(Fo=3):FOR S=0 TO 3: [4617]
¦SECWRITE,do,T,S,&8800+S*512:NEXT S
3630 RUN                                   [243]
3640 ln=INT((Ds1+127)/128):N=Nb:IF Nb>32 T [3956]
HEN N=32:ln=128
3650 lg=lg+1:ar=Q(lg):GOSUB 3800:POKE ar,c [4403]
:ar=ar+1:POKE ar,0:ar=ar+1
3660 POKE ar,0:ar=ar+1:POKE ar,ln:ar=ar+1: [4698]
Ds1=Ds1-&4000:Nb=Nb-32:z=32*c
3670 FOR l=1 TO N STEP 2:w=B1(l+z):T=w\9-( [3213]
Fi=3)-2*(Fi=0):s=w MOD 9
3680 ¦SETFORM,Fi:¦SECREAD,din,T,S,bw       [3532]
3690 FOR j=jp TO 180:IF blok(j)=0 THEN 370 [2861]
0 ELSE NEXT j:GOTO 3470
3700 jp=j:blok(j)=1:POKE ar,j:ar=ar+1:Ta=2 [2771]
*j\9-2*(Fo=0)-(Fo=3)
3710 sa=j*2 MOD 9:¦SETFORM,Fo:¦SECWRITE,do [2271]
,Ta,Sa,bw
3720 IF l+1>N THEN 3750 ELSE w=B1(l+1+z):s [3745]
=w MOD 9:T=w\9-(Fi=3)-2*(Fi=0)
3730 ¦SETFORM,Fi:¦SECREAD,din,T,S,bw:Sa=Sa [6779]
+1:IF Sa>8 THEN Sa=0:Ta=Ta+1
3740 ¦SETFORM,Fo:¦SECWRITE,do,Ta,Sa,bw     [2308]
3750 NEXT l:FOR ag=ar TO Q(lg)+31:POKE ag, [1639]
0:NEXT ag:RETURN
3760 r=F(lf):f1$=LEFT$(B$(r),12):LOCATE#1, [2017]
59,2:PRINT#1,SPACE$(16);
3770 LOCATE#1,59,2:PRINT#1,e$;r;f1$" ";e$: [1613]
RETURN
3780 CLS:F=Fi:GOSUB 7400:LOCATE 5,5:PRINT" [3275]
Insgesamt ";;Nb;" Sektoren":RETURN
3790 PRINT"Start von Sektor ";St,"Laenge"; [4041]
Ds;" kB":RETURN
3800 POKE ar,0:ar=ar+1:FOR j=1 TO 8:g=ASC( [3235]
MID$(f1$,j,1)):POKE ar,g
3810 ar=ar+1:NEXT j:FOR j=10 TO 12:g=ASC(M [2500]
ID$(f1$,j,1)):POKE ar,g
3820 ar=ar+1:NEXT j:RETURN                 [951]
3830 '                                     [117]
3840 ' *   *   *   *   *   *   *   *   *   [445]
 *   *   *   *   *   *   *
3850 '                                     [117]
4000 '      E D I T I E R E N              [603]
4010 '                                     [117]
4020 j=4:S$="P":GOSUB 5110:IF f1=0 THEN RU [4019]
N
4030 CLS:WINDOW#6,6,53,6,21:WINDOW#7,63,78 [3728]
,6,21:CLS#6:CLS#7
4040 FOR lf=1 TO pf:GOSUB 5130:GOSUB 7500: [2498]
CLS:ORIGIN 0,0
4050 xg=34:yg=60:xr=398:yr=264:GOSUB 6610: [4215]
xg=492:xr=140:GOSUB 6610
4060 FOR l=1 TO pl:x=1:y=1:ad0=&A300:GOSUB [4153]
 8420:R=B1(Q(l)):T=R\9-2*(F=0)-(F=3)
4070 S=R MOD 9:PRINT#2,,"Spur:";T,"Sektor: [6517]
";S:¦SECREAD,d,T,S,ad0:¦TEX,ad0
4080 e$=CHR$(24):LOCATE#c%,x,y:a$=COPYCHR$ [3601]
(#c%):PRINT#c%,e$;a$;e$;
4090 i$="":GOSUB 8240:IF c%=6 THEN i$=UPPE [2584]
R$(i$)
4100 i=ASC(i$):IF i>239 AND i<252 THEN i=i [2616]
-239:GOSUB 8100:GOTO 4080
```

Listing "FILE TRANSFER"

```
4110 IF i<i1 OR i>i2 THEN 4130 ELSE IF c%= [3860]
6 AND (i>57 AND i<65) THEN 4130
4120 a$=i$:GOSUB 8310:x=x-((x MOD v)=2):ON [3238]
 c%-5 GOTO 4140,4180
4130 IF i=13 THEN 4200 ELSE 4080          [1477]
4140 x6=3*((x-1)\3)+1:x7=x6\3+1:LOCATE#6,x [4863]
6,y:b1$=COPYCHR$(#6):LOCATE#6,x6+1,y
4150 b2$=COPYCHR$(#6):b=VAL("&"+b1$)*16+VA [3389]
L("&"+b2$):LOCATE#7,x7,y
4160 IF b<32 OR b>126 THEN PRINT#7,"." ELS [2896]
E PRINT#7,CHR$(b);
4170 POKE ad0+16*(y-1)+x7-1,b:GOSUB 8230:G [2794]
OTO 4080
4180 x7=x:x6=3*x7-2:LOCATE#6,x6,y:b=ASC(a$ [4244]
):PRINT#6,HEX$(b,2);
4190 POKE ad0+16*(y-1)+x7-1,b:GOSUB 8230:G [2794]
OTO 4080
4200 LOCATE#2,20,2:PRINT#2,"Sektor aufnehm [4166]
em      (J/N) ";:INPUT#2,s$
4210 IF UPPER$(s$)<>"J" THEN 4230          [1920]
4220 ¦SECWRITE,d,T,S,&A300                 [1816]
4230 NEXT l:NEXT lf                        [331]
4240 RUN                                   [243]
4250 '                                     [117]
4260 '   *   *   *   *   *   *   *   *   * [528]
   *   *   *   *   *   *   *
4270 '                                     [117]
5000 '     C L E A R I N G      P C        [1621]
5010 '                                     [117]
5020 CLS:j=5:GOSUB 6000:LOCATE 20,10:PRINT [2180]
 CHR$(7);
5030 INPUT "SIND   SIE   SICHER  (J/N) ";B [3020]
$:IF UPPER$(B$)<>"J" THEN RUN
5040 F=2:OPENOUT "B:Dumy":P=PEEK(&BE51):IF [2450]
 P>64 THEN 3440
5050 CLOSEOUT:POKE &A300,&FC:POKE &A301,&F [3864]
F:POKE &A302,&FF
5060 FOR x=&A303 TO &A4FF:POKE x,0:NEXT x: [5943]
¦SETFORM,F:¦SECWRITE,1,0,1,&A300
5070 FOR x=&A300 TO &A4FF:POKE x,&F6:NEXT  [2970]
x
5080 FOR x=&A300 TO &A4FF STEP 32:POKE x,0 [2425]
:NEXT
5090 FOR s=5 TO 8:¦SECWRITE,1,0,s,&A300:NE [2079]
XT s:RUN
5100 '                                     [117]
5110 GOSUB 6000:GOSUB 6100:y=2:GOSUB 6400  [2514]
5120 GOSUB 6700:GOSUB 7000:RETURN          [1453]
5130 r=F(lf):f1$=LEFT$(B$(r),12):LOCATE#1, [2017]
59,2:PRINT#1,SPACE$(16);
5140 LOCATE#1,59,2:PRINT#1,e$;r;f1$" ";e$: [2315]
GOSUB 7400:RETURN
5150 '                                     [117]
5160 '    *   *   *   *   *   *   *   *    [528]
*   *   *   *   *   *   *   *
5170 '                                     [117]
6000 '     S U B R O U T I N E S           [644]
6010 '                                     [117]
6020 '     Windows  definieren             [933]
6030 ¦DISPL:e$=CHR$(24):xg= 0:yg=18:xr=639 [4121]
:yr=330:GOSUB 6610
6040 WINDOW#0,2,79,5,23:CLS:ORIGIN 8,0     [2134]
6050 LOCATE#1,27,2:PRINT#1,text$(j);:RETUR [3390]
N
6060 '                                     [117]
6100 '     Diskettenlaufwerk  definieren   [1553]
6110 tx3$=tx0$+CHR$(242)+"  "+CHR$(243)+tx [5742]
1$:PRINT#2,tx3$:y=8:CLS:x=50:d$="B "
6120 d1$=" Discdrive ":p$=d1$+d$:GOSUB 621 [3793]
0:x=20:d$="A ":p$=d1$+d$
6130 GOSUB 6210:GOSUB 6140:IF d$="A " THEN [4553]
 d=0:GOTO 6300 ELSE d=1:GOTO 6300
6140 I$="":WHILE I$="":p$=d1$+d$:GOSUB 724 [3902]
0:I$=INKEY$:GOSUB 7250:WEND
6150 I=ASC(I$):IF i=242 THEN x=20:d$="A "  [2945]
ELSE IF i=243 THEN x=50:d$="B "
6160 IF I<>13 THEN 6140 ELSE p$=d1$+d$:GOS [2421]
UB 7240:RETURN
6170 '                                     [117]
6200 xg=8*x-12:yg=333-16*y:xr=111:yr=22:GO [2216]
SUB 6610:RETURN
6210 GOSUB 7250:GOSUB 6200:RETURN          [1401]

6220 '                                     [117]
6300 tx$(0)=" D I S K E T T E    I N    D [2406]
R I V E  "+d$+"  L E G E N ! "
6310 GOSUB 2360                            [819]
6320 form$(0)="CP/M":form$(1)="DATA":form$ [4190]
(2)=" IBM":form$(3)="JOYCE":RETURN
```

Listing "FILE TRANSFER"

```
6330 '                                     [117]
6400 '      Diskettenformat Bestimmen      [1598]
6410 '                                     [117]
6420 CLS#2:LOCATE#2,18,1:PRINT#2,"     D I [4232]
 S K E T T E N F O R M A T      ";
6430 sr=0:OPENOUT d$+":Dumy":IF PEEK(&BE51 [2773]
)>&B0 THEN F=1:dr=0:UM=178:GOTO 6450
6440 IF PEEK(&BE51)>&40 THEN F=0:dr=2:UM=1 [3495]
69:GOTO 6450
6442 F=3:dr=1:sr=0:ip=1:UM=173             [1465]
6445 IF d$="B " OR d$="B" THEN F=2:dr=0:sr [1678]
=5:ip=0:UM=175
6450 CLOSEOUT:LOCATE#1,3,y:PRINT#1," Forma [3091]
t in ";d$;": ";:T=dr
6460 PRINT#1,form$(F);:¦SETFORM,F:a=REMAIN [2865]
(1):CLS#2:RETURN
6470 I$="":WHILE I$="":l=5.5*(y-5)+(x+7)\7 [3409]
:p$=DEC$(Bl(l),"####")+" ":GOSUB 7240
6480 I$=INKEY$:GOSUB 7250:WEND:I=ASC(I$)   [1782]
6490 IF i=242 THEN x=x-7:IF x<0 THEN x=72: [2061]
IF y>5 THEN y=y-2
6500 IF i=243 THEN x=x+7:IF x>74 THEN x=2: [3884]
IF y<ymax THEN y=y+2
6510 IF i=240 THEN y=y+2*(y>5) ELSE IF i=2 [3050]
41 THEN y=y-2*(y<ymax)
6520 IF i<>224 THEN 6540 ELSE pl=pl+1:Q(pl [3352]
)=l:ORIGIN 8,0
6530 xg=56*(x\7)+6:yg=398-16*(y+4):xr=44:y [4796]
r=20:GOSUB 6610:GOTO 6470
6540 IF i<> 13 THEN 6470 ELSE RETURN       [1385]
6550 '                                     [117]
6600 xg=44+60*(x\10):yg=400-16*(y+8):xr=10 [2748]
0:yr=16
6610 MOVE xg,yg:DRAWR xr,0,1:DRAWR 0,yr:DR [4775]
AWR -xr,0,1:DRAWR 0,-yr:RETURN
6620 '                                     [117]
6700 '      Directory   laden              [1792]
6710 CLS#2:LOCATE#2,18,1:PRINT#2,"  D I R [3719]
E C T O R Y    L A D E N          ";
6720 l=0:lo=0:FOR S=sr TO sr+3:¦SECREAD,d, [3276]
T,S,&8000+512*(S-sr):NEXT
6730 CLS#2:LOCATE#2,18,1:PRINT#2,"  B L O [4430]
E C K E      B E S T I M M E N     ";
6740 FOR x=&8000 TO &87FF STEP 32:p1=PEEK( [5386]
x):x0=x:f$="":IF F=2 THEN 6800
6750 l=l+1:IF p1=32 THEN GOSUB 7300:B$(0)= [2393]
f$:GOTO 6830
6760 IF p1>15 THEN 6830 ELSE lo=lo+1:GOSUB [1941]
 7300
6770 f$=f$+FNS$(x,12):Nc=(PEEK(x+15)+3)\4  [2714]
6780 f$=f$+CHR$(Nc):IF S$="O" THEN B$(lo)= [3254]
f$:GOTO 6830 ELSE N=(Nc+1)\2
6790 FOR j=1 TO N:f$=f$+FNS$(x+15,j):NEXT  [2844]
j:B$(lo)=f$:GOTO 6830
6800 IF p1=0 THEN 6830 ELSE lo=lo+1:GOSUB  [1894]
7300:Nc=INT((FNP(x,28)+511)/512)
6805 Nc=Nc+128*PEEK(x+30):f$=f$+CHR$(0)+CH [1804]
R$(Nc)
6810 IF S$="O" THEN B$(lo)=f$:GOTO 6830    [1822]
6820 FOR j=24 TO 30:f$=f$+FNS$(x,j):NEXT j [3886]
:B$(lo)=f$:Ln=LEN(f$)
6830 NEXT x:fl=lo:FOR l=1 TO fl-1:f1$=LEFT [2517]
$(B$(l),12)
6840 FOR j=l+1 TO fl:IF f1$<>LEFT$(B$(j),1 [4585]
2) THEN 6880 ELSE N1=FNM(B$(l),14)
6850 N2=FNM(B$(j),14):f$=LEFT$(B$(l),13)+C [2660]
HR$(N1+N2)
6860 IF S$<>"O" THEN f$=f$+RIGHT$(B$(l),(N [2677]
1+1)\2)+RIGHT$(B$(j),(N2+1)\2)
6870 B$(l)=f$:FOR k=j+1 TO fl:B$(k-1)=B$(k [3775]
):Q(k-1)=Q(k):NEXT k:fl=fl-1:l=l-1
6880 NEXT j:NEXT l:RETURN                  [1734]
6890 '                                     [117]
6900 '          Bloecke sortieren          [1234]
6910 CLS#2:LOCATE#2,20,1:PRINT#2,"         [3071]
   S O R T I E R E N          ";
6920 FOR l=1 TO fl-1:f1$=LEFT$(B$(l),13):F [2070]
OR j=l+1 TO fl
6930 f2$=LEFT$(B$(j),13):IF f1$<f2$ THEN 6 [4054]
950 ELSE f0$=B$(j):B$(j)=B$(l)
6940 B$(l)=f0$:f1$=LEFT$(f0$,13):Q=Q(l):Q( [2227]
l)=Q(j):Q(j)=Q
6950 NEXT j:NEXT l:RETURN                  [1734]
6960 '                                     [117]
7000 CLS:LOCATE 20,1:PRINT"I N H A L T S V [4294]
 E R Z E I C H N I S":ZONE 20
7010 LOCATE 1,4:tx4$="     A U S W A H L   [2709]
M I T   CURSOR - TASTEN"
7015 tx4$=tx4$+" +  COPY"+tx1$             [1701]
```

Listing "FILE TRANSFER"


```
7020 TAG:ls=0:x=0:y=4:FOR l=1 TO fl:y=300- [3694]
20*((l-1)\4):x=160*((l-1) MOD 4)-3
7030 MOVE x,y:f$=LEFT$(B$(l),12):PRINT DEC [3937]
$(l,"###");" - ";f$;:NEXT l:TAGOFF
7040 ymax=300-20*((fl-1)\4):e$=CHR$(24):PR [5093]
INT#2,tx4$:p$=LEFT$(B$(l),12)
7050 x=45:y=300:GOSUB 7210:pf=0:GOSUB 7100 [4427]
:CLS#2:RETURN
7060 '                                     [117]
7100 I$="":WHILE I$="":l=4*((300-y)\20)+(x [4284]
\160)+1:p$=LEFT$(B$(l),12):z0=1:z1=0
7110 GOSUB 7200:I$=INKEY$:z0=0:z1=1:GOSUB  [3140]
7200:WEND:I=ASC(I$)
7120 IF i=242 THEN x=x-160:IF x<45 THEN x= [3073]
525:y=y-20*(y<300)
7130 IF i=243 THEN x=x+160:IF x>532 THEN x [3219]
=45:y=y+20*(y>ymax)
7140 IF i=240 THEN y=y-20*(y<300) ELSE IF  [3990]
i=241 THEN y=y+20*(y>ymax)
7150 IF i=224 THEN pf=pf+1:F(pf)=l:GOSUB 7 [3021]
170:GOTO 7100
7160 IF i=13 THEN RETURN ELSE 7100         [2009]
7170 z0=0:z1=1:GOSUB 7210:xg=x-2:yg=y-16   [1886]
7180 xr=100:yr=18:GOSUB 6610:RETURN        [2257]
7200 GOSUB 7210:GOSUB 7220:pause=60:GOTO 7 [2387]
260
7210 GRAPHICS PAPER z0:GRAPHICS PEN z1:RET [4336]
URN
7220 TAG:MOVE x,y:PRINT p$;:z0=0:z1=1:GOSU [4354]
B 7210:TAGOFF:RETURN
7230 '                                     [117]
7240 LOCATE x,y:PRINT e$;p$;e$;:pause=100: [1532]
GOTO 7260
7250 LOCATE x,y:PRINT p$;:pause=50         [1104]
7260 FOR q=1 TO pause:NEXT:RETURN          [2562]
7270 '                                     [117]
7300 FOR j=0 TO 7:g=PEEK(1+(F=2)+j+x0)     [2252]
7310 IF g>31 AND g<127 THEN f$=f$+CHR$(g)  [2366]
ELSE f$=f$+"?"
7320 NEXT j:IF p1<>32 THEN f$=f$+"."       [1258]
7330 FOR j=8 TO 10:g=PEEK(1+(F=2)+j+x0)    [1338]

7340 IF g>31 THEN f$=f$+CHR$(g) ELSE f$=f$ [1462]
+"?"
7350 NEXT j:RETURN                         [844]
7360 '                                     [117]
7370 '                                     [117]
7400 k=F(lf):x1=0:l1=PEEK(@B$(k)):Adr=PEEK [5157]
(@B$(k)+1)+256*PEEK(@B$(k)+2)-1
7410 Nb=PEEK(Adr+14):Ds1=512*Nb:Ds=Ds1/102 [3789]
4:FOR i=0 TO 128:B1(i)=0:NEXT
7420 IF F=2 THEN 7440 ELSE Nb1=(Nb+1)\2:FO [1831]
R k=1 TO Nb1:x1=x1+2:
7430 B1(x1-1)=2*PEEK(Adr+14+k):B1(x1)=B1(x [3350]
1-1)+1:NEXT k:RETURN
7440 Ds1=256*PEEK(Adr+20)+PEEK(Adr+19):Ds= [1869]
Ds1/1024+64*PEEK(Adr+21)
7450 St=256*PEEK(Adr+18)+PEEK(Adr+17):FOR  [4051]
x=1 TO Nb:B1(x)=St+6+x:NEXT x:RETURN
7460 '                                     [117]
7500 CLS:LOCATE 20,3:PRINT"DATEI IST IN FO [4819]
LGENDEN SEKTOREN GELADEN:  "
7510 CLS#2:PRINT#2,tx4$                    [1620]
7520 FOR x=1 TO Nb:LOCATE 7*((x-1) MOD 11) [3519]
+3,5+2*((x-1)\11):p$=DEC$(B1(x),"###")
7530 PRINT p$:NEXT x:x=2:y=5:ymax=5+2*((Nb [2230]
-1)\11):e$=CHR$(24):l=1
7540 p$=DEC$(B1(l),"####")+" ":GOSUB 7240: [3936]
p1=0:GOSUB 6470:CLS#2:RETURN
7550 '                                     [117]
7600 y=10:p2$="   DRUCKER   ":p1$="   MONI [3681]
TOR   ":p$=p2$:x=50:GOSUB 6210:x=20
7610 :p$=p1$:GOSUB 6210:GOSUB 7620:IF x=20 [2497]
 THEN 7700 ELSE 7660
7620 PRINT#2,tx3$:I$="":WHILE I$="":GOSUB  [4472]
7240:I$=INKEY$:GOSUB 7250:WEND
7630 I=ASC(I$):IF i=242 THEN x=20:p$=p1$ E [3790]
LSE IF i=243 THEN x=50:p$=p2$
7640 IF i=13 THEN GOSUB 7240:RETURN ELSE 7 [2370]
620
7650 xg=8*x-12:yg=333-16*y:xr=80:yr=22:GOS [3017]
UB 6610:RETURN
7660 tx$(0)="         DRUCKER EINSTELLEN U [3467]
ND PAPIER EINLEGEN !          "
7670 GOSUB 2360                            [819]
7680 G=32:q$=CHR$(14):WIDTH 126:PRINT#8,es [3554]
c$;CHR$(15);esc$;CHR$(67);CHR$(72);
7690 PRINT#8,esc$;CHR$(78);CHR$(6);esc$;CH [3713]
R$(108);CHR$(10):c%=8:c1%=8:RETURN
```

Listing "FILE TRANSFER"

```
7700 G=16:c%=0:c1%=2:q$="":RETURN          [1108]
7710 '                                     [117]
7800 FOR x=j3 TO j3+15:PRINT#c%,HEX$(PEEK( [3468]
a+x),2);f$;:NEXT x:RETURN
7810 FOR x=j3 TO j3+15:d%=PEEK(a+x):       [2156]
7820 IF d%<32 OR d%>127 THEN PRINT#c%,f$;" [4551]
.";f$; ELSE PRINT#c%,f$;CHR$(d%);f$;
7830 NEXT x:RETURN                         [1045]
7840 '                                     [117]
7900 CLS:LOCATE 10,10:PRINT e$;SPACE$(34); [4229]
CHR$(7);:LOCATE 10,11
7910 PRINT text$;:LOCATE 10,12:PRINT;SPACE [3735]
$(34);e$;:RETURN
7920 CLS:LOCATE 30,6:PRINT"Hexadezimal   c [3972]
od:":y=8:x=50:p$=h2$:GOSUB 7250
7930 GOSUB 7650:x=20:p$=h1$:GOSUB 7250:GOS [4461]
UB 7650:GOSUB 8000:IF x=50 THEN H1=1
7940 LOCATE 30,11:PRINT"     ASCII   cod:" [3371]
:y=13
7950 x=50:p$=h2$:GOSUB 7250:GOSUB 7650:x=2 [4391]
0:p$=h1$:GOSUB 7250:GOSUB 7650
7960 GOSUB 8000:IF x=50 THEN H2=1:RETURN E [3239]
LSE RETURN
7970 '                                     [117]
8000 PRINT#2,tx3$:I$="":WHILE I$="":GOSUB  [4472]
7240:I$=INKEY$:GOSUB 7250:WEND
8010 I=ASC(I$):IF i=242 THEN x=20:p$=h1$ E [3694]
LSE IF i=243 THEN x=50:p$=h2$
8020 IF i=13 THEN GOSUB 7240:RETURN ELSE 8 [2502]
000
8030 '                                     [117]
8100 GOSUB 8310:IF i>4 THEN 8200 ELSE y=y- [4831]
(i=1)*(y>1)+(i=2)*(y<16)
8110 x=x+v*(i=3)-v*(i=4):x=v*((x-1)\v)+1:I [6725]
F x<1 THEN x=xmax:y=y+(y>1):RETURN
8120 IF x<xmax+1 THEN RETURN ELSE x=1:y=y- [4553]
(y<16):RETURN
8130 '                                     [117]
8200 IF i=9 THEN 8400 ELSE IF i=10 THEN 84 [2751]
10 ELSE IF i<9 THEN 8220
8210 IF i=-11*(c%=7) THEN 8420 ELSE IF i=- [2448]
12*(c%=6) THEN 8430 ELSE RETURN
8220 IF i<7 THEN y=-(i=5)-16*(i=6)::RETURN [5363]
 ELSE x=-(i=7)-xmax*(i=8):RETURN
8230 IF x<xmax THEN x=x+1:RETURN ELSE x=1: [3862]
y=y-(y<16):RETURN
8240 WHILE i$="":i$=INKEY$:GOSUB 8300:GOSU [3963]
B 8310:WEND:RETURN
8250 '                                     [117]
8300 LOCATE#c%,x,y:PRINT#c%,e$;a$;e$;:RETU [3073]
RN
8310 LOCATE#c%,x,y:PRINT#c%,a$;:RETURN     [3382]
8320 '                                     [117]
8400 IF ad0=&A300 THEN RETURN ELSE ad0=&A3 [5348]
00:CLS#6:CLS#7:¦TEX,ad0:RETURN
8410 IF ad0=&A400 THEN RETURN ELSE ad0=&A4 [4698]
00:CLS#6:CLS#7:¦TEX,ad0:RETURN
8420 c%=6:v=3:xmax=47:i1=48:i2=70:x=3*x-2: [2540]
RETURN
8430 c%=7:v=1:xmax=16:i1=32:i2=126:x=x\3+1 [3390]
:RETURN
```

```
10 '   C P C T O O L - L O A D E R         [968]
20 ad=&A000:FOR i=0 TO 33:FOR j=1 TO 20:RE [3751]
AD c$:c=VAL("&"+c$)
30 POKE ad,c:ad=ad+1:sum=sum+c:NEXT j:READ [2588]
 c$:s=VAL("&"+c$)
40 IF s<>sum THEN 90 ELSE sum=0:NEXT i     [2475]
50 CLS:LOCATE 21,8:PRINT "D A T E N      S [3562]
 I N D    O. K.":LOCATE 19,12
60 PRINT "D I S K E T T E      E I N L E G [4979]
 E N":PRINT:PRINT:LOCATE 12,14
70 PRINT "U N D     E I N E     T A S T E [2705]
      D R U E C K E N"
```

Listing "FILE TRANSFER"

```
80 CALL &BB18:SAVE"CPCTOOL.BIN",b,&A000,66 [1966]
6:END
90 PRINT "F E L E R      I N      D A T A [3668]
     L I N I E:  ";100+10*I:STOP
100 DATA 01,09,A0,21,4E,A0,C3,D1,BC,20,A0, [2528]
C3,9C,A0,C3,A5,A0,C3,52,A0,A85
110 DATA C3,58,A0,C3,5E,A0,C3,39,A1,C3,9A, [3165]
A1,53,45,43,52,45,41,C4,53,9E1
120 DATA 45,43,57,52,49,54,C5,45,52,52,4F, [3763]
52,4F,46,C6,45,52,52,4F,52,702
130 DATA 4F,CE,53,45,54,46,4F,52,CD,54,45, [4145]
D8,44,49,53,50,CC,00,00,00,72A
140 DATA 00,00,3E,00,32,24,A1,C9,3E,01,32, [3134]
24,A1,C9,F5,3E,00,32,23,A1,626
150 DATA F1,FE,01,C2,04,A1,DD,7E,00,B7,28, [3176]
0B,FE,01,28,12,FE,02,18,19,806
160 DATA C3,04,A1,3E,0A,32,25,A1,3E,40,32, [3457]
26,A1,C9,3E,0A,32,25,A1,3E,666
170 DATA C0,32,26,A1,C9,3E,09,32,25,A1,3E, [2781]
00,32,26,A1,C9,CD,AC,A0,3E,818
180 DATA 84,CD,E6,A0,C9,CD,AC,A0,3E,85,18, [2861]
F5,F5,3E,00,32,23,A1,F1,FE,BA1
190 DATA 04,20,4C,DD,7E,06,FE,02,30,45,32, [2955]
1A,A1,DD,7E,04,FE,28,D2,03,78D
200 DATA A1,32,1B,A1,21,25,A1,DD,7E,02,BE, [3042]
30,2E,21,26,A1,8E,32,1C,A1,754
210 DATA DD,66,01,DD,6E,00,22,1D,A1,C9,21, [3197]
22,A1,77,CD,D4,BC,22,1F,A1,8D2
220 DATA 79,32,21,A1,21,1A,A1,5E,23,56,23, [3116]
4E,2A,1D,A1,DF,1F,A1,C9,E1,7C2
230 DATA 3E,FF,32,23,A1,3A,24,A1,B7,C8,21, [4223]
27,A1,7E,B7,C8,CD,5A,BB,23,99C
240 DATA 18,F7,00,00,00,00,00,00,00,00,00, [3340]
00,00,00,00,50,61,72,61,6D,300
250 DATA 65,74,61,72,20,65,72,72,6F,72,21, [2513]
07,00,FE,01,20,D1,DD,6E,00,759
```

Listing "FILE TRANSFER"

```
260 DATA DD,66,01,E5,3E,06,CD,B4,BB,CD,6C, [3872]
BB,E1,E5,06,00,C5,7E,23,E5,AB4
270 DATA F5,1F,1F,1F,1F,CD,8A,A1,F1,CD,8A, [2866]
A1,3E,20,CD,5A,BB,E1,C1,10,A44
280 DATA E7,3E,07,CD,B4,BB,CD,6C,BB,E1,06, [3168]
00,C5,7E,23,E5,FE,20,38,04,9E8
290 DATA FE,7F,38,02,3E,2E,CD,5A,BB,E1,C1, [3729]
10,EB,C9,E6,0F,FE,0A,30,04,99C
300 DATA C6,30,18,02,C6,37,CD,5A,BB,C9,3E, [3762]
02,CD,0E,BC,3E,01,CD,B4,BB,90A
310 DATA 21,00,00,11,02,4F,CD,66,BB,3E,01, [4371]
CD,96,BB,3E,00,CD,90,BB,CD,7F1
320 DATA 6C,BB,3E,1B,CD,6F,BB,3E,02,CD,72, [3779]
BB,21,44,A2,CD,39,A2,3E,3E,BDC
330 DATA CD,6F,BB,3E,01,CD,72,BB,21,5C,A2, [3060]
CD,39,A2,3E,3C,CD,6F,BB,3E,9A6
340 DATA 03,CD,72,BB,21,66,A2,CD,39,A2,06, [3668]
05,48,CD,38,BC,11,CA,00,21,7DE
350 DATA 68,01,CD,C0,BB,3E,00,CD,DE,BB,21, [4299]
76,A2,3E,FF,5E,BB,28,10,23,93F
360 DATA 56,23,7E,23,4E,23,E5,6F,61,CD,F9, [2778]
BB,E1,18,EA,3E,02,CD,B4,BB,A20
370 DATA 21,18,00,11,18,4F,CD,66,BB,CD,6C, [3349]
BB,3E,00,CD,B4,BB,21,04,00,732
380 DATA 11,16,4F,CD,66,BB,CD,6C,BB,7E,B7, [4081]
C8,23,E5,CD,5A,BB,E1,18,F5,B2D
390 DATA 20,44,20,49,20,53,20,43,20,4D,20, [3338]
4F,20,4E,20,49,20,54,20,4F,439
400 DATA 20,52,20,00,41,20,75,20,74,20,6F, [3027]
20,72,00,5A,61,68,61,72,20,533
410 DATA 53,76,65,74,69,73,6C,61,76,00,C0, [2715]
00,00,00,00,00,20,00,01,00,4A2
420 DATA E0,FF,00,00,20,00,3F,FF,00,00,00, [3490]
00,E0,FF,FF,FF,00,00,00,00,71A
430 DATA 20,00,FF,00,FF,00,00,00,00,00,00, [3607]
00,00,00,00,00,00,00,00,00,21E
```

Listing "FILE TRANSFER"

Komplex

Rechnen mit komplexen Zahlen

Das Programm KOMPLEX bietet die Möglichkeit (wie von FORTRAN bekannt), unter BASIC mit komplexen Zahlen zu rechnen.

Folgende Funktionen stehen zur Auswahl:
- **Summe**
- **Differenz**
- **Produkt**
- **Bruch**
- **Hochzahl**

als Funktionen von zwei komplexen Zahlen oder
- **Betrag**
- **Kehrwert**
- **Wurzel**
- **Quadrat**
- **Logarithmus**

als Funktionen von einer komplexen Zahl.

Beschreibung

Nachdem eine Funktion gewählt und die Zahlenwerte getrennt nach Realteil und Imaginärteil eingegeben wurden, wird die vom Benutzer gestellte Aufgabe noch einmal angezeigt. Die Aufgabenstellung kann jetzt auf ihre Richtigkeit hin überprüft werden, falsche Eingaben können korrigiert werden.

Sind alle Eingaben in Ordnung, werden im Ergebnismenü die gewählte Funktion, die Aufgabenstellung und das Ergebnis auf dem Bildschirm angezeigt. Danach kann der Benutzer wählen, ob die Daten auch noch auf dem Drucker ausgegeben werden sollen, oder ob eine neue Funktion gewählt werden soll. Wird die Druckerausgabe gewünscht, so stehen Fettdruck oder Entwurfsdruck zur Verfügung. Der Ausdruck kann natürlich wiederholt werden. Wählt der Benutzer die Eingabe einer neuen Funktion, so erscheint wieder das Eingabemenü.

Wählt man dort statt einer Funktion "I" für Informationen, so erhält man auf mehreren Bildschirmseiten zusätzliche Informationen zu den komplexen Zahlen und zum Aufbau der komplexen Zahlenebene.

Mit "E" läßt sich das Programm beenden.

Alle Eingaben sind mit einer Sicherheitsabfrage versehen und erklären sich von selbst. Eingabefehler werden von dem Programm erkannt, abgefangen und dem Benutzer gemeldet.

Das Programm KOMPLEX wurde auf einem CPC 464 (mit DDI-1 und Vortex Speichererweiterung SP 64) erstellt, ist aber auf allen CPC-Rechnern mit oder ohne Floppy und mit oder ohne Vortex-Erweiterung lauffähig.

Druckeranpassung

Die Steuerzeichen im Programm sind für den Drucker Brother M-1009 beziehungsweise NLQ 401 gedacht. Das Programm läßt sich aber sehr leicht an andere Drucker anpassen, weil es nur auf die Steuerzeichen für "Fettdruck", "engl. Zeichensatz", "Breitschrift" und "Hochindex" zugreift. Diese Steuerzeichen können in den folgenden Zeilen (1870 bis 1940) geändert werden, indem die entsprechenden CHR$-Werte aus dem Druckerhandbuch eingetragen werden.


```
1870 fde$=CHR$(27)+CHR$(69)          :REM
        Fettdruck ein
1880 fda$=CHR$(27)+CHR$(70)          :REM
        Fettdruck aus
1890 eze$=CHR$(27)+CHR$(55)          :REM
      engl. Zeichensatz ein
1900 eza$=CHR$(27)+CHR$(54)          :REM
      engl. Zeichensatz aus
1910 bse$=CHR$(14)                   :REM
      Breitschrift ein
1920 bsa$=CHR$(20)                   :REM
      Breitschrift aus
1930 hie$=CHR$(27)+CHR$(83)+CHR$(0):REM
      Hochindex ein
1940 hia$=CHR$(27)+CHR$(84)          :REM
      Hochindex aus
```

Hinweis

Das im Listing verwendete "Dach" (^), entspricht dem Hochpfeil auf der Pfundtaste.

Variablenliste von KOMPEX

a$	Nimmt die Eingabe nach Tastaturabfrage auf
a	Farbwert für Titelgrafik
aufg$	Aufgabenstellung
b	Farbwert für Titelgrafik
betr	Funktion "Betrag"
bez$	Name der entsprechenden Funktion
bsa$	"Breitschrift aus"
bse$	"Breitschrift ein"
diff	Funktion "Differenz"
ergeb$	Ergebnis der entsprechenden Funktion
eza$	"engl. Zeichensatz aus"
eze$	"engl. Zeichensatz ein"
fda$	"Fettdruck aus"
fde$	"Fettdruck ein"
h	Farbwert für Titelgrafik
hia$	"Hochindex aus"
hie$	"Hochindex ein"
hoch	Funktion "Hochzahl"
i$	Chr$(24), (invers)
i	Laufindex für Schleifen
idiff	Imaginärteil der Differenz
ihoch	Imaginärteil der Hochzahl
ikehr	Imaginärteil des Kehrwertes
iln	Imaginärteil des nat. Log.
im	Nimmt Imaginärteil der Eingabe auf (bei Fkt. von 2 Variablen)
ima	Nimmt Imaginärteil der Eingabe auf (bei Fkt. von 1 Variablen)
iprod	Imaginärteil des Produktes
iquad	Imaginärteil des Quadrates
iquot	Imaginärteil des Quotienten
isumm	Imaginärteil der Summe
iwurz	Imaginärteil der Wurzel
ix	Imaginärteil bei der Funktionsdefinition (1.Variable)
iy	Imaginärteil bei der Funktionsdefinition (2.Variable)
kehr	Funktion "Kehrwert"
ln	Funktion "ln"
prod	Funktion "Produkt"
quad	Funktion "Quadrat"
quot	Funktion "Quotient"
r	Farbwert für Titelgrafik, Radius für Infografik
rdiff	Realteil der Differenz
re	Nimmt Realteil der Eingabe auf (bei Fkt. von 2 Variablen)
rea	Nimmt Realteil der Eingabe auf (bei Fkt. von 1 Variablen)
rhoch	Realteil der Hochzahl
rkehr	Realteil des Kehrwertes
rln	Realteil des nat. Log.
rprod	Realteil des Produktes
rquad	Realteil des Quadrates
rquot	Realteil des Quotienten
rsumm	Realteil der Summe
rwurz	Realteil der Wurzel
rx	Realteil bei der Funktionsdefinition (1.Variable)
ry	Realteil bei der Funktionsdefinition (2. Variable)
summ	Funktion "Summe"
t$	Tabulator (3 Zeichen)
v%	Kennzahl der Funktion
wurz	Funktion "Wurzel"
zeichen$	Verknüpfungszeichen zwischen den einzelnen Zahlen

(Gert Ravensburg/cd)

```
SUMME

( 1.3 ,-4.1 ) + (-8.5 , 3 ) = (-7.2 ,-1.1 )

DIFFERENZ

( 2 , 7 ) - ( 5 , 2 ) = (-3 , 5 )

PRODUKT

( 2.42 ,-3.91 ) * ( 3.14 , 1.19 ) = ( 12.2517 ,-9.3976 )

BRUCH

( 11.4 ,-2.45 ) / ( 3 , 9.81 ) = ( 9.65971E-02 ,-1.13253912 )

POTENZWERT

( 3 , 0 ) ^ ( 2 , 0 ) = ( 9 , 0 )

ABSOLUTBETRAG

!( 3 , 4 )! =  5

KEHRWERT

1 / ( 4 , 2.3 ) = ( 0.187881635 ,-0.10803194 )

WURZEL

SQR ( 9 , 0 ) = ( 3 , 0 )

QUADRAT

( 2.7 , 5.1 )² = (-18.72 , 27.54 )

LOGARITHMUS

LN ( 1 , 0 ) = ( 0 , 0 )
```

Einige Beispiele von den Möglichkeiten die Komplex bietet.

für 464-664-6128

```
10 ' ***** KOMPLEX *****                                    [1222]
20 '                                                        [117]
30 ' ********************                                   [669]
40 ' *                  *                                   [714]
50 ' * Gert Ravensburg *                                    [1193]
60 ' *  Bad Wildungen  *                                    [266]
70 ' *    29.9.1986    *                                    [587]
80 ' ********************                                   [669]
90 '                                                        [117]
100 GOSUB 2510 :REM Titelbild                               [1705]
110 CLEAR                                                   [217]
120 GOSUB 1740 :REM Sonderzeichen-Definiti                  [4577]
on
130 GOSUB 1330 :REM Funktionen-Definition                   [3164]
140 GOSUB 1870 :REM Druckersteuerzeichen                    [3481]
150 ON ERROR GOTO 2840                                      [1507]
160 '------ Eingabe ---------------------                   [3160]
------------------------------------
170 MODE 1:BORDER 10:i$=CHR$(24):t$=CHR$(9                  [4382]
)+CHR$(9)+CHR$(9)
180 DRAWR 0,399,1:DRAWR 639,0:DRAWR 0,-399                  [2868]
:DRAWR -639,0:MOVER 0,96:DRAWR 639,0
190 LOCATE 13,2:PRINT i$;" K O M P L E X "                  [2266]
;i$
200 LOCATE 7,5:PRINT"Rechnen mit komplexen                  [3046]
 Zahlen":PRINT
210 PRINT t$+i$;"<S>";i$;"umme....                          [3504]
";i$;"<A>";i$;"bsolutbetrag":PRINT
220 PRINT t$+i$;"<D>";i$;"ifferenz                          [5436]
";i$;"<K>";i$;"ehrwert.....":PRINT
230 PRINT t$+i$;"<P>";i$;"rodukt..                          [3577]
";i$;"<W>";i$;"urzel.......":PRINT
240 PRINT t$+i$;"<B>";i$;"ruch....                          [4265]
";i$;"<Q>";i$;"uadrat......":PRINT
250 PRINT t$+i$;"<H>";i$;"ochzahl.                          [3674]
";i$;"<L>";i$;"ogarithmus..":PRINT
260 PRINT:PRINT t$+i$;"<E>";i$;"nde.....                    [3619]
      ";i$;"<I>";i$;"nformationen"
270 WINDOW 2,39,21,24                                       [967]
280 CLS:LOCATE 11,2:PRINT"Bitte w";CHR$(24                  [4710]
0);"hlen sie!"
290 GOSUB 1300                                              [950]
300 IF a$="S" THEN bez$="SUMME":zeichen$="                  [3590]
 + ":v%=1:GOTO 430
310 IF a$="D" THEN bez$="DIFFERENZ":zeiche                  [4223]
n$=" - ":v%=2:GOTO 430
320 IF a$="P" THEN bez$="PRODUKT":zeichen$                  [3577]
=" * ":v%=3:GOTO 430
330 IF a$="B" THEN bez$="BRUCH":zeichen$="                  [3573]
 / ":v%=4:GOTO 430
340 IF a$="H" THEN bez$="POTENZWERT":zeich                  [4440]
en$=" ^ ":v%=5:GOTO 430
350 IF a$="A" THEN bez$="ABSOLUTBETRAG":ze                  [4041]
ichen$="|":v%=6:GOTO 550
360 IF a$="K" THEN bez$="KEHRWERT":zeichen                  [3482]
$="1 / ":v%=7:GOTO 550
370 IF a$="W" THEN bez$="WURZEL":zeichen$=                  [4951]
CHR$(247)+" ":v%=8:GOTO 550
380 IF a$="Q" THEN bez$="QUADRAT":zeichen$                  [2374]
=CHR$(248):v%=9:GOTO 550
390 IF a$="L" THEN bez$="LOGARITHMUS":zeic                  [4711]
hen$="LN ":v%=10:GOTO 550
400 IF a$="I" THEN GOSUB 1970:GOTO 170                      [2417]
410 IF a$="E" THEN GOSUB 2780:CLEAR:END                     [2612]
420 GOTO 290                                                [411]
430 CLS                                                     [91]
440 PRINT bez$;" zweier komplexer Zahlen"                   [4329]
450 FOR i=1 TO 2                                            [436]
460 LOCATE 1,3:PRINT"Zahl";i;"    Realteil                  [4133]
";:INPUT re(i)
470 PRINT"Zahl";i;"Imagin";CHR$(240);"rtei                  [3589]
l";:INPUT im(i)
480 LOCATE 1,3:PRINT CHR$(20);                              [1218]
490 NEXT i                                                  [375]
500 CLS                                                     [91]
510 aufg$="("+STR$(re(1))+","+STR$(im(1))+                  [5316]
")"+zeichen$+"("+STR$(re(2))+","+STR$(im(2
))+")"
```

Listing "Komplex"

```
520 LOCATE 1,1:IF LEN(aufg$)>38 THEN PRINT                  [7185]
"  (";re(1);",";im(1);")":PRINT RIGHT$(zei
chen$,2);"(";re(2);",";im(2);")";:GOTO 640
530 PRINT aufg$;                                            [347]
540 GOTO 640                                                [373]
550 CLS                                                     [91]
560 PRINT bez$;" einer komplexen Zahl"                      [2775]
570 LOCATE 1,3:PRINT"Realteil    ";:INPUT                   [2211]
rea
580 PRINT"Imagin";CHR$(240);"rteil";:INPUT                  [3560]
 ima
590 LOCATE 1,3:PRINT CHR$(20);                              [1218]
600 CLS:LOCATE 1,1                                          [831]
610 IF LEFT$(bez$,1)="A" THEN PRINT zeiche                  [2672]
n$;"(";rea;",";ima;")";zeichen$:GOTO 640
620 IF LEFT$(bez$,1)="Q" THEN PRINT"(";rea                  [3492]
;",";ima;")";zeichen$:GOTO 640
630 PRINT zeichen$;"(";rea;",";ima;")"                      [2469]
640 LOCATE 1,4:PRINT"Aufgabenstellung korr                  [4420]
ekt?   (J/N)"
650 GOSUB 1300                                              [950]
660 IF a$<>"N" THEN 730                                     [903]
670 LOCATE 1,4:PRINT i$;"<Z>";i$;"ahlenwer                  [6978]
te oder ";i$;"<F>";i$;"unktion falsch ?";
680 GOSUB 1300                                              [950]
690 IF a$="Z" THEN ON v% GOTO 430,430,430,                  [2827]
430,430,550,550,550,550,550
700 IF a$="F" THEN 170                                      [1420]
710 GOTO 680                                                [460]
720 '------ Auswertung -------------------                  [3966]
------------------------------------
730 ON v% GOSUB 1190,1200,1210,1220,1230,1                  [3077]
240,1250,1260,1270,1280
740 '------ Ausgabe ----------------------                  [2782]
------------------------------------
750 MODE 1                                                  [506]
760 DRAWR 0,399,1:DRAWR 639,0:DRAWR 0,-399                  [2950]
:DRAWR -639,0:MOVER 0,46:DRAWR 639,0
770 LOCATE 13,2:PRINT i$;" K O M P L E X "                  [2266]
;i$
780 LOCATE 12,5:PRINT"Gew";CHR$(240);"hlte                  [2707]
 Funktion"
790 LOCATE 13,6:PRINT CHR$(150);STRING$(13                  [3805]
,CHR$(154));CHR$(156)
800 LOCATE 13,7:PRINT CHR$(149);SPC(13);CH                  [2955]
R$(149)
810 LOCATE 13,8:PRINT CHR$(147);STRING$(13                  [2307]
,CHR$(154));CHR$(153)
820 LOCATE 12,10:PRINT"Aufgabenstellung"                    [4033]
830 LOCATE 2,11:PRINT CHR$(150);STRING$(36                  [4411]
,CHR$(154));CHR$(156)
840 FOR i=1 TO 3:LOCATE 2,11+i:PRINT CHR$(                  [3483]
149);SPC(36);CHR$(149):NEXT i
850 LOCATE 2,15:PRINT CHR$(147);STRING$(36                  [4447]
,CHR$(154));CHR$(153)
860 LOCATE 16,17:PRINT"Ergebnis"                            [2206]
870 LOCATE 2,18:PRINT CHR$(150);STRING$(36                  [3882]
,CHR$(154));CHR$(156)
880 LOCATE 2,19:PRINT CHR$(149);SPC(36);CH                  [3196]
R$(149)
890 LOCATE 2,20:PRINT CHR$(147);STRING$(36                  [3792]
,CHR$(154));CHR$(153)
900 WINDOW#1,14,26,7,7:WINDOW#2,3,38,12,14                  [5138]
:WINDOW#3,3,38,19,19:WINDOW 2,39,24,24
910 FOR i=1 TO 3:PEN#i,0:PAPER#i,2:CLS#i:N                  [2200]
EXT i
920 PRINT#1,bez$;:PRINT#2                                   [1136]
930 IF v%<6 THEN IF LEN(aufg$)>36 THEN PRI                  [6780]
NT#2,"  (";re(1);",";im(1);")":PRINT#2,RIG
HT$(zeichen$,2);"(";re(2);",";im(2);")";:G
OTO 980
940 IF v%<6 THEN PRINT#2,aufg$;                             [1196]
950 IF v%=6 THEN PRINT#2,zeichen$;"(";rea;                  [1819]
",";ima;")";zeichen$
960 IF v%=9 THEN PRINT#2,"(";rea;",";ima;"                  [2415]
)";zeichen$
970 IF v%=7 OR v%=8 OR v%=10 THEN PRINT#2,                  [3675]
zeichen$;"(";rea;",";ima;")"
980 PRINT#3,ergeb$;                                         [805]
990 CLS:PRINT i$;"<N>";i$;"eue Eingabe ode                  [6079]
r ";i$;"<D>";i$;"ruckerausgabe ?";
1000 GOSUB 1300                                             [950]
1010 IF a$<>"D" THEN 170                                    [1743]
```

Listing "Komplex"


```
1020 CLS:PRINT i$;"<E>";i$;"ntwurfsdruck o                  [3716]
der ";i$;"<F>";i$;"ettdruck ?";
1030 GOSUB 1300                                             [950]
1040 IF a$="F" THEN CLS:PRINT"Druckerausga                  [6268]
be in Fettdruck":PRINT#8,fde$;:GOTO 1060
1050 CLS:PRINT"Druckerausgabe in Entwurfsd                  [3384]
ruck"
1060 PRINT#8,eze$;                                          [904]
1070 PRINT#8,bse$;bez$;bsa$:PRINT#8                         [2921]
1080 IF v%<6 THEN PRINT#8,"(";re(1);",";im                  [3792]
(1);")";zeichen$;"(";re(2);",";im(2);")";
1090 IF v%=6 THEN PRINT#8,"|(";rea;",";ima                  [2174]
;")|";
1100 IF v%=8 THEN PRINT#8,"SQR (";rea;",";                  [3231]
ima;")";
1110 IF v%=9 THEN PRINT#8,"(";rea;",";ima;                  [1790]
")";hie$;"2";hia$;
1120 IF v%=7 OR v%=10 THEN PRINT#8,zeichen                  [2279]
$;"(";rea;",";ima;")";
1130 PRINT#8," = ";ergeb$:PRINT#8,fda$;eza                  [2095]
$:PRINT#8
1140 CLS:PRINT"Noch ein Ausdruck?   (J/N)"                  [2075]
1150 GOSUB 1300                                             [950]
1160 IF a$<>"J" THEN 170 ELSE 1020                          [2101]
1170 END                                                    [110]
1180 '------ Ergebnisse -------------------                 [3954]
------------------------------------
1190 ergeb$=FN summ(re(1),im(1),re(2),im(2                  [1809]
)):RETURN
1200 ergeb$=FN diff(re(1),im(1),re(2),im(2                  [2684]
)):RETURN
1210 ergeb$=FN prod(re(1),im(1),re(2),im(2                  [2981]
)):RETURN
1220 ergeb$=FN quot(re(1),im(1),re(2),im(2                  [2072]
)):RETURN
1230 ergeb$=FN hoch(re(1),im(1),re(2),im(2                  [3165]
)):RETURN
1240 ergeb$=FN betr(rea,ima):RETURN                         [1588]
1250 ergeb$=FN kehr(rea,ima):RETURN                         [1543]
1260 ergeb$=FN wurz(rea,ima):RETURN                         [1320]
1270 ergeb$=FN quad(rea,ima):RETURN                         [2363]
1280 ergeb$=FN ln(rea,ima):RETURN                           [2435]
1290 '------ Tastaturabfrage -------------                 [3432]
------------------------------------
1300 a$=UPPER$(INKEY$):IF a$="" THEN 1300                   [1604]
1310 RETURN                                                 [555]
1320 '------ Funktionen -------------------                 [4703]
------------------------------------
1330 DEFSTR s,d,p,q,b,k,w,l,h                               [407]
1340 '------ Summe ------                                   [887]
1350 DEF FN rsumm(rx,ry)=rx+ry                              [1354]
1360 DEF FN isumm(ix,iy)=ix+iy                              [956]
1370 DEF FN summ(rx,ix,ry,iy)="("+STR$(FN                   [4660]
 rsumm(rx,ry))+" ,"+STR$(FN isumm(ix,iy))+"
 )"
1380 '------ Differenz ------                               [1196]
1390 DEF FN rdiff(rx,ry)=rx-ry                              [1162]
1400 DEF FN idiff(ix,iy)=ix-iy                              [1835]
1410 DEF FN diff(rx,ix,ry,iy)="("+STR$(FN                   [3797]
 rdiff(rx,ry))+" ,"+STR$(FN idiff(ix,iy))+"
 )"
1420 '------ Produkt ------                                 [1533]
1430 DEF FN rprod(rx,ix,ry,iy)=rx*ry-ix*iy                  [2099]
1440 DEF FN iprod(rx,ix,ry,iy)=rx*iy+ix*ry                  [1961]
1450 DEF FN prod(rx,ix,ry,iy)="("+STR$(FN                   [4697]
 rprod(rx,ix,ry,iy))+" ,"+STR$(FN iprod(rx,
 ix,ry,iy))+" )"
1460 '------ Quotient ------                                [1422]
1470 DEF FN rquot(rx,ix,ry,iy)=(rx*ry+ix*i                  [2682]
 y)/(ry^2+iy^2)
1480 DEF FN iquot(rx,ix,ry,iy)=(ix*ry-rx*i                  [2858]
 y)/(ry^2+iy^2)
1490 DEF FN quot(rx,ix,ry,iy)="("+STR$(FN                   [5089]
 rquot(rx,ix,ry,iy))+" ,"+STR$(FN iquot(rx,
 ix,ry,iy))+" )"
1500 '------ Kehrwert ------                                [1080]
1510 DEF FN rkehr(rx,ix)=rx/(rx^2+ix^2)                     [2871]
1520 DEF FN ikehr(rx,ix)=-ix/(rx^2+ix^2)                    [956]
1530 DEF FN kehr(rx,ix)="("+STR$(FN rkehr(                  [3817]
 rx,ix))+" ,"+STR$(FN ikehr(rx,ix))+" )"
1540 '------ Betrag ------                                  [677]
```

Listing "Komplex"

```
1550 DEF FN betr(rx,ix)=STR$(SQR(rx^2+ix^2                  [1927]
))
1560 '------ Wurzel ------                                  [1917]
1570 DEF FN rwurz(rx,ix)=SQR((SQR(rx^2+ix^                  [3413]
 2)+rx)/2)
1580 DEF FN iwurz(rx,ix)=SQR((SQR(rx^2+ix^                  [3119]
 2)-rx)/2)
1590 DEF FN wurz(rx,ix)="("+STR$(FN rwurz(                  [3403]
 rx,ix))+" ,"+STR$(FN iwurz(rx,ix))+" )"
1600 '------ Quadrat ------                                 [1754]
1610 DEF FN rquad(rx,ix)=rx^2-ix^2                          [1978]
1620 DEF FN iquad(rx,ix)=2*rx*ix                            [1649]
1630 DEF FN quad(rx,ix)="("+STR$(FN rquad(                  [5404]
 rx,ix))+" ,"+STR$(FN iquad(rx,ix))+" )"
1640 '------ nat. Logarithmus ------                        [1363]
1650 DEF FN rln(rx,ix)=LOG(SQR(rx^2+ix^2))                  [3482]
1660 DEF FN iln(rx,ix)=ATN(ix/rx)                           [1517]
1670 DEF FN ln(rx,ix)="("+STR$(FN rln(rx,i                  [3263]
 x))+" ,"+STR$(FN iln(rx,ix))+" )"
1680 '------ Hochzahl ------                                [2005]
1690 DEF FN rhoch(rx,ix,ry,iy)=(SQR(rx^2+i                  [5976]
 x^2))^ry*EXP(-(ATN(ix/rx)*iy))*COS(iy*LOG(
 SQR(rx^2+ix^2))+ATN(ix/rx)*ry)
1700 DEF FN ihoch(rx,ix,ry,iy)=(SQR(rx^2+i                  [6818]
 x^2))^ry*EXP(-(ATN(ix/rx)*iy))*SIN(iy*LOG(
 SQR(rx^2+ix^2))+ATN(ix/rx)*ry)
1710 DEF FN hoch(rx,ix,ry,iy)="("+STR$(FN                   [5316]
 rhoch(rx,ix,ry,iy))+" ,"+STR$(FN ihoch(rx,
 ix,ry,iy))+" )"
1720 RETURN                                                 [555]
1730 '------ Sonderzeichen ---------------                  [2790]
------------------------------------
1740 SYMBOL 240,108,0,120,12,124,204,118,0                  [2410]
      :REM ae
1750 SYMBOL 241,198,56,108,198,254,198,198                  [2620]
,0    :REM AE
1760 SYMBOL 242,102,0,60,102,102,102,60,0                   [2580]
      :REM oe
1770 SYMBOL 243,198,56,108,198,198,108,56,                  [3218]
0     :REM OE
1780 SYMBOL 244,102,0,102,102,102,102,62,0                  [3015]
      :REM ue
1790 SYMBOL 245,198,0,198,198,198,198,124,                  [2500]
0     :REM UE
1800 SYMBOL 246,124,198,198,220,198,198,22                  [2893]
0,192 :REM sz
1810 SYMBOL 247,7,5,4,228,52,28,12,4                        [2456]
      :REM Wurzel
1820 SYMBOL 248,12,18,4,8,30,0,0,0                          [2865]
      :REM Quadrat
1830 SYMBOL 249,110,219,219,126,24,24,24,4                  [3491]
8     :REM Phi
1840 SYMBOL 250,87,21,223,68,236,0,0,0                      [2960]
      :REM i*Phi
1850 RETURN                                                 [555]
1860 '------ Druckersteuerzeichen --------                  [3612]
------------------------------------
1870 fde$=CHR$(27)+CHR$(69)          :REM F                 [1943]
ettdruck ein
1880 fda$=CHR$(27)+CHR$(70)          :REM F                 [2273]
ettdruck aus
1890 eze$=CHR$(27)+CHR$(55)          :REM e                 [2569]
ngl. Zeichensatz ein
1900 eza$=CHR$(27)+CHR$(54)          :REM e                 [2740]
ngl. Zeichensatz aus
1910 bse$=CHR$(14)                   :REM B                 [2643]
reitschrift ein
1920 bsa$=CHR$(20)                   :REM B                 [2014]
reitschrift aus
1930 hie$=CHR$(27)+CHR$(83)+CHR$(0) :REM H                  [2830]
ochindex ein
1940 hia$=CHR$(27)+CHR$(84)          :REM H                 [2389]
ochindex aus
1950 RETURN                                                 [555]
1960 '------ Erklaerung der komplexen Zahl                  [3839]
en ------------------------------------
1970 MODE 1:BORDER 1                                        [1439]
1980 LOCATE 13,1:PRINT i$;" K O M P L E X                   [2593]
";i$
1990 LOCATE 1,4:PRINT"Zahlenpaare der Form                  [5400]
 z=(a,b) mit a,b ";CHR$(180);" R"
```

Listing "Komplex"

```
2000 PRINT"hei";CHR$(246);"en komplexe Zah [4614]
len.":PRINT
2010 PRINT"Man nennt Re(z) = a den Realtei [5875]
l von z,":PRINT
2020 PRINT"Im(z) = b den Imagin";CHR$(240) [3264]
;"rteil von z und":PRINT
2030 PRINT"schreibt die komplexe Zahl auch [5961]
 h";CHR$(240);"ufig":PRINT
2040 PRINT"in der Form z = a + b*i. Die ko [3973]
mplexen":PRINT
2050 PRINT"Zahlen der Form (a,0) identifiz [4758]
iert man":PRINT
2060 PRINT"mit den reellen Zahlen a, die d [5053]
er Form":PRINT
2070 PRINT"(0,b) werden dagegen als imagin [6168]
";CHR$(240);"re Zahl-"
2080 PRINT"en bezeichnet. Die komplexe Zah [4631]
l (0,1)":PRINT
2090 PRINT"hei";CHR$(246);"t imagin";CHR$( [6159]
240);"re Einheit i, mit i";CHR$(248);" = -
1."
2100 LOCATE 16,25:PRINT i$;" <TASTE> ";i$  [1890]
2110 CALL &BB18                            [389]
2120 CLS:LOCATE 13,1:PRINT i$;" K O M P L  [2516]
E X ";i$
2130 LOCATE 21,13:PRINT">"                 [934]
2140 TAG:MOVE 154,399:PRINT CHR$(160);:TAG [2847]
OFF
2150 MOVE 0,200:DRAW 320,200               [1410]
2160 MOVE 160,0:DRAW 160,390               [1835]
2170 LOCATE 9,4:PRINT"3i":LOCATE 9,7:PRINT [4786]
"2i":LOCATE 10,10:PRINT"i"
2180 LOCATE 9,16:PRINT"-i":LOCATE 8,19:PRI [5668]
NT"-2i":LOCATE 8,22:PRINT"-3i"
2190 LOCATE 1,14:PRINT"-3 -2 -1":LOCATE 13 [2967]
,14:PRINT"1  2  3"
2200 FOR i=1 TO 7 STEP 3:MOVE i*16+8,196:D [3576]
RAWR 0,8:NEXT i
2210 FOR i=12 TO 18 STEP 3:MOVE i*16+8,196 [2877]
:DRAWR 0,8:NEXT i
2220 FOR i=3 TO 21 STEP 3:MOVE 156,i*16+8: [2399]
DRAWR 8,0:NEXT i
2230 LOCATE 1,1:PRINT"imagin";CHR$(240);"r [3764]
e":LOCATE 5,2:PRINT"Achse"
2240 LOCATE 16,17:PRINT"reelle":LOCATE 17, [3866]
18:PRINT"Achse"
2250 LOCATE 19,7:PRINT CHR$(231)           [1460]
2260 INK 2,10:INK 3,16                     [382]
2270 MOVE 160,200:DRAW 296,296,1:DRAW 296, [2112]
200,2:DRAW 160,200,3
2280 LOCATE 14,9:PRINT"r":PEN 2:LOCATE 20, [5720]
10:PRINT"b":PEN 3:LOCATE 15,15:PRINT"a":PE
N 1
2290 LOCATE 20,6:PRINT"z"                  [1312]
2300 ORIGIN 160,200                        [776]
2310 DEG:r=75                              [530]
2320 FOR i=0 TO 35:PLOT r*COS(i),r*SIN(i), [3670]
1:NEXT i:RAD
2330 LOCATE 14,12:PRINT CHR$(249)          [2275]
2340 WINDOW 23,40,3,25:PEN 0:PAPER 7:CLS   [2050]
2350 PRINT" Darstellung einer";:PRINT      [2248]
2360 PRINT" komplexen Zahl z":PRINT        [2485]
2370 PRINT" als Punkt in der":PRINT        [2110]
2380 PRINT" Ebene mit den":PRINT           [2690]
2390 PRINT" Koordinaten (a,b)";:PRINT      [2137]
2400 PRINT" z=a+i*b":PRINT                 [1742]
2410 PRINT" a=Realteil z":PRINT            [2850]
2420 PRINT" b=Imagin";CHR$(240);"rteil z": [2987]
PRINT
2430 PRINT" r=Betrag von z":PRINT          [1243]
2440 PRINT" ";CHR$(249);"=Argument von z": [3072]
PRINT
2450 PRINT" z=r*(cos";CHR$(249);"+i*sin";C [4378]
HR$(249);")";:PRINT
2460 PRINT" z=r*e";CHR$(250)               [2479]
2470 PEN 1:PAPER 0:WINDOW 1,40,1,25:LOCATE [4701]
 13,25:PRINT i$;" <TASTE> ";i$
2480 CALL &BB18:INK 2,20:INK 3,6:BORDER 10 [2067]
2490 RETURN                                [555]
2500 '------ Titelbild ------------------- [3658]
-------------------------------------
```

Listing "Komplex"

```
2510 SYMBOL &F7,&3,&7,&1F,&3E,&38,&70,&F0, [2709]
&E0
2520 SYMBOL &F8,&FF,&FF,&FF,&0,&0,&0,&7,&F [1910]
2530 SYMBOL &F9,&C0,&E0,&F8,&7F,&1F,&0,&7, [2602]
&9F
2540 SYMBOL &FA,&E0,&E0,&E0,&E0,&E0,&E0,&E [2451]
0,&E0
2550 SYMBOL &FB,&1D,&18,&18,&18,&1F,&1F,&1 [2850]
F,&18
2560 SYMBOL &FC,&FC,&F0,&C0,&C0,&FF,&FF,&F [1847]
F,&7
2570 SYMBOL &FD,&E0,&F0,&70,&38,&3E,&1F,&7 [1904]
,&3
2580 SYMBOL &FE,&0,&0,&0,&0,&0,&FF,&FF,&FF [2753]
2590 SYMBOL &FF,&7,&7,&7,&1F,&7F,&F8,&E0,& [2247]
C0
2600 DEFINT a-z:MODE 1                     [1278]
2610 INK 0,0:INK 1,26:INK 2,13:INK 3,18:r= [2822]
0:h=1:b=2:a=3:PAPER a:PEN r:BORDER 6
2620 CLS:ORIGIN 16,173                     [718]
2630 PRINT CHR$(150);STRING$(38,CHR$(154)) [3723]
;CHR$(156);
2640 FOR i=1 TO 23:PRINT CHR$(149);SPC(38) [2258]
;CHR$(149);:NEXT i
2650 PRINT CHR$(147);STRING$(38,CHR$(154)) [3957]
;CHR$(153);
2660 MOVE 30,0:DRAW 0,0,h:DRAW 0,210,h:DRA [1640]
W 140,210,h:DRAW 140,75,h
2670 MOVE 30,0:DRAW 30,75,b:DRAW 45,75,b:D [9852]
RAW 45,55,b:DRAW 55,55,b:DRAW 55,75,b:DRAW
 70,75,b:DRAW 70,55,b:DRAW 80,55,b:DRAW 80
,75,b:DRAW 85,75,b:MOVE 125,55:DRAW 130,55
,b:DRAW 130,75,b:DRAW 140,75,b:DRAW 140,0,
b:DRAW 30,0,b
2680 MOVE 125,0:DRAW 125,160,r:DRAW 78,186 [4734]
,r:DRAW 10,127,r:DRAW 72,148,r:DRAW 57,128
,r:DRAW 125,0,r
2690 LOCATE 5,19:PRINT CHR$(247);CHR$(248) [9600]
;CHR$(249):LOCATE 5,20:PRINT CHR$(250);CHR
$(251);CHR$(252):LOCATE 5,21:PRINT CHR$(25
3);CHR$(254);CHR$(255)
2700 ORIGIN 0,0:WINDOW 13,37,4,22:PAPER b: [2628]
CLS
2710 LOCATE 7,3:PRINT"K O M P L E X"       [1736]
2720 LOCATE 7,6:PRINT"Version  2.00"       [1964]
2730 LOCATE 9,9:PRINT CHR$(164);" 1987 by" [1893]
2740 LOCATE 6,11:PRINT"Gert Ravensburg":LO [9288]
CATE 12,13:PRINT"aus":LOCATE 7,15:PRINT"Ba
d Wildungen":LOCATE 10,19:PRINT"<Taste>"
2750 CALL &BB18:DEFREAL a-z:INK 0,1:INK 1, [4748]
24:INK 2,20:INK 3,6:PAPER 0:PEN 1:BORDER 1
:MODE 1
2760 RETURN                                [555]
2770 '------ Ende ------------------------ [2892]
-------------------------------------
2780 MODE 1:BORDER 6,20                    [1538]
2790 LOCATE 5,12:PRINT"Programm wirklich b [5080]
eenden? (J/N)"
2800 GOSUB 1300                            [950]
2810 IF a$="N" THEN BORDER 1:GOTO 150      [2329]
2820 IF a$="J" THEN BORDER 1:MODE 1:RETURN [1642]
 ELSE 2800
2830 '------ Fehlerbehandlung ------------ [4120]
-------------------------------------
2840 LOCATE 1,3:PRINT CHR$(20);            [1218]
2850 IF ERR=2 THEN LOCATE 1,4:PRINT i$;" S [3527]
yntax Fehler! ";i$;" --- <TASTE>";:GOTO 29
00
2860 IF ERR=5 THEN LOCATE 1,4:PRINT i$;" i [7745]
st nicht definiert! ";i$;" --- <TASTE>";:G
OTO 2900
2870 IF ERR=6 THEN LOCATE 1,4:PRINT i$;" h [5118]
at zu gro";CHR$(246);"e Werte! ";i$;" ---
<TASTE>";:GOTO 2900
2880 IF ERR=11 THEN LOCATE 1,4:PRINT i$;"  [6031]
teilt durch die Null! ";i$;" --- <TASTE>";
:GOTO 2900
2890 LOCATE 1,4:PRINT i$;" FEHLER ";ERR;"  [4895]
in Zeile ";ERL;" ";i$;" <TASTE>";
2900 CALL &BB18                            [389]
2910 RESUME 670                            [645]
```

Listing "Komplex"

Haben Sie uneingeschränkten Zugriff auf jeden Sektor Ihrer Disketten? Und, wenn ja, gilt das auch für Ihre kopiergeschützten Disketten? Nein? Dann sind Sie hier genau richtig. Wenn Ihnen vielleicht auch etwas daran liegt, fremde Diskettenformate zu untersuchen, dann sind Sie hier sogar goldrichtig. Was wir Ihnen anzubieten haben, ist einmal ein sehr leistungsfähiger Diskettenmonitor, zum anderen eine RSX-Erweiterung zur bequemen FDC-Programmierung. Beide Programme bedienen sich des direkten Eingriffs in den Floppy-Disc-Controller. Sie können damit jeden Sektor lesen, auch Sektoren mit gelöschter DAM, Sektoren mit einfacher Dichte, jede Sektorgröße, Sektoren mit unnormaler Sektor-ID etc. Wahrscheinlich kommt einigen Lesern der eine oder andere Begriff unbekannt vor, deshalb möchten wir hier einige Erklärungen voranstellen.



Diskettenzauber

AMSMON PLUS und BASIC-765

Wie Sie vielleicht schon wissen, sind die Daten auf der Diskette sektorweise abgelegt. Dazu ist die Diskette in mehrere konzentrische Spuren aufgeteilt (unter AMSDOS werden 40 Spuren verwendet). Jede Spur ist in mehrere Sektoren unterteilt. Die Anzahl der Sektoren ist dabei abhängig von der Sektorgröße und dem Aufzeichungsmodus.

Der FDC 765 kann in zwei verschiedenen Modi arbeiten. Man nennt sie **MF (einfache Dichte)** und **MFM (doppelte Dichte)** . Der MFM-Modus bietet gegenüber dem MF-Modus die doppelte Spurkapazität bei ebenfalls doppelter Übertragungsgeschwindigkeit. Deshalb wird der MF-Modus heute praktisch nicht mehr angewandt.

Im MFM-Modus passen ungefähr 6500 Bytes auf eine Spur. Allerdings benötigt der FDC auch einige Bytes für die Verwaltung. Hierin sind z.B. die Sektor-IDs sowie der Sektorabstand enthalten (der Sektorabstand dient zum Ausgleich unterschiedlicher Umdrehungs- Geschwindigkeiten der Laufwerke). Nach Abzug dieses Overheads bleiben z.B. bei 512 Byte-Sektoren 5kB für Daten über, das entspricht 10 Sektoren. Bei anderen Sektorgrößen muß beachtet werden, daß der Verwaltungsaufwand natürlich mit der Anzahl der Sektoren steigt.

Damit nun der FDC einen Sektor findet, legt er für jeden Sektor eine Sektor-ID an. In dieser ID stehen die Tracknummer, die Diskettenseite, die Sektornummer sowie die Sektorgröße.

Die Tracknummer muß dabei nicht unbedingt mit dem physikalischen Track, also dem Track, auf dem der Schreib-/Lesekopf tatsächlich steht, übereinstimmen. Dies kann z.B. als Kopierschutz herangezogen werden. Mit den Routinen aus AMSDOS sind nämlich nur Sektoren lesbar, bei denen die Tracknummer in der ID mit der physikalischen übereinstimmt. Mit AMSMON PLUS oder BASIC-765 sind Sie von derartigen Beschränkungen nicht mehr abhängig.

Ebenfalls muß die angegebene Diskettenseite nicht mit der tatsächlichen übereinstimmen. Dies liegt einfach daran, daß der Inhalt der ID beim Formatieren von der CPU an den FDC geliefert wird. Bei der Übertragung sind natürlich beliebige Angaben möglich. Um den Sektor später zu bearbeiten, muß der Kopf über der gewünschten Spur positioniert werden. Anschließend müssen beim Lesen oder Schreiben die Werte aus der ID angegeben werden.

Das Ganze mag zwar auf den ersten Blick äußerst kompliziert aussehen, wenn Sie allerdings später mit BASIC-765 und/oder AMSMON PLUS die am Schluß folgenden Beispiele nach vollziehen, wird Ihnen vieles klarer erscheinen.

Zum Schluß dieser Einleitung noch eine Bemerkung zu den gelöschten Sektoren; dies sind Sektoren mit gelöschter 'DATA-ADDRESS-MARK'. Um es gleich vorwegzunehmen: gelöschte Sektoren sind keine "leeren" Sektoren oder vielleicht mit Nullen gefüllte Sektoren - sie enthalten genauso Daten wie "normale" Sektoren. Der Unterschied besteht im Prinzip nur aus einem "Flag", das den Zustand des Sektors beschreibt. Dieses Flag wird z.B. bei modernen Betriebssystemen dazu genutzt, um defekte Sektoren zu kennzeichnen. Einige Software-Firmen bauen mit Hilfe dieser Sektoren ihren Kopierschutz auf.

Die Programme

Kommen wir nun zu den beiden Programmen. Beide laufen auf allen Schneider CPCs unter AMSDOS. Für Besitzer von VDOS werde ich später für BASIC-765 eine Anpassung anführen.

BASIC-765 ist eine RSX-Erweiterung zur einfachen Programmierung des Floppy-Controllers FDC 765. Es sind alle wesentlichen FDC-Kommandos als RSX-Befehle ausführbar. Diese Erweiterung ist in erster Linie für Programmierer gedacht sowie für VDOS-Besitzer mit geringen Assembler-Kenntnissen, die die später beschriebene Anpassung vornehmen können.

Dagegen ist AMSMON PLUS für die Anwender die bessere Wahl. Dieser Diskettenmonitor bietet u.a. einen Full-Screen Editor und äußerst intelligente Disk-Routinen. Zum Beispiel braucht sich der Anwender nicht mehr um die Sektorgröße und die in der ID angegebenen Daten zu kümmern. Nach der Positionierung des Kopfes reicht allein die Angabe der Sektornummer, den Rest der Arbeit übernimmt der Diskettenmonitor. Desweiteren besitzt AMSMON PLUS einige weitere interessante Features. Der Monitor greift teilweise sehr stark in das Betriebssystem ein. Eine Anpassung an VDOS ist daher praktisch unmöglich. Nun gut, sehen wir uns erst einmal BASIC-765 an.

BASIC-765 – Floppy-Programmierung einfach gemacht.

BASIC-765 ist auf zwei verschiedenen Wegen in das BASIC integrierbar. Einmal kann es auf dem "normalen" Wege geschehen:

```
MEMORY &9DFF: LOAD "BASIC765: CALL &9E00
```

Anschließend stehen alle RSX-Befehle zur Verfügung. Eine Initialisierung des Betriebssystems wird dabei nicht vorgenommen. Die zweite Möglichkeit ist radikaler und einfacher. Sie starten die Datei mit

```
RUN "BASIC765
```

Anschließend wird eine komplette Initialisierung des Betriebssystems vorgenommen. Es erscheint wieder die Einschaltmeldung. Nach Initialisierung aller ROMs wird die RSX-Erweiterung eingebunden. Anschließend wird das BASIC initialisiert. BASIC-765 ist nun voll ins Betriebssystem integriert.
In beiden Fällen belegt die Erweiterung den Speicherbereich von &9E80 bis &A63F.
Kommen wir nun zu den neuen Befehlen, die die Erweiterung bereitstellt.

|ASCII

Dieser Befehl gibt den aktuellen Pufferinhalt in reiner ASCII- Darstellung aus. Die Länge des Puffers ist abhängig von der aktuellen Sektorlänge. Die Ausgabe kann mit SPACE angehalten und mit ESC abgebrochen werden.

|DREAD,sector
|DREAD,track,sector

Dieser Befehl dient zum Lesen eines gelöschten Sektors. Im Prinzip scheint er unnötig zu sein, da auch ein gelöschter Sektor mit dem READ- Befehl eingelesen werden kann. Der Befehl erwartet ein oder zwei Parameter. Falls nur ein Parameter angegeben wird, wird dieser als Sektornummer interpretiert. Als Tracknummer wird dann die Defaulttracknummer angenommen (die vom letzten SEEK oder HOME (=0) Befehl). Wenn zwei Parameter übergeben werden, gibt der erste den Track an, der zweite die Sektornummer. Sofern die Meldungen zugelassen sind (siehe MSGON, MSGOFF) wird anschließend die 'Result- Phase' des FDC hexadezimal ausgegeben. Der Sektor wird an die Anfangsadresse des aktuellen Puffers gelesen. Eine Kopfpositionierung wird nicht durchgeführt. Dies kann vorher mit dem SEEK-Befehl geschehen.

|DUMP

Der aktuelle Pufferinhalt wird als kombinierter Hex-/ASCII-Dump ausgegeben. Bezüglich der Pufferlänge und der Ausgabe gilt das Gleiche wie beim |ASCII-Befehl.

|DWRITE,sector
|DWRITE,track,sector

Dieser Befehl schreibt einen Sektor mit gelöschter DAM auf die Diskette. Als Datenadresse dient die Anfangsadresse des Puffers. Bezüglich der Parameterübergabe gilt das gleiche wie beim |DREAD-Befehl.

|ERROR,@integer%

Dieser Befehl holt den aktuellen Fehlerstatus. Dafür muß die Adresse einer Integervariablen übergeben werden. Der Fehlerstatus entspricht dem Inhalt des Status-Registers 0.

|FGAP,byte

Wie oben schon erwähnt, fügt der FDC zwischen den Sektoren Abstände ein, um ein Überschreiben von Daten zu vermeiden. Diese Abstände werden GAPs genannt. Es existieren auch GAPs zwischen der ID und den Daten, zu Statusinformationen usw. Einer dieser GAPs ist die GAP3. Dieser kann beim Formatieren und beim Sektorschreiben angegeben werden. Um ein Überschreiben zu vermeiden, sollte natürlich die GAP beim Formatieren größer sein als beim normalen Schreiben. Die Größe dieser GAP für das Formatieren kann mit diesem Befehl eingestellt werden. Als Default hat sie die Länge &52. Falls Sie nun einen Track mit 10 Sektoren mit je 512 Byte formatieren wollen, ist diese GAP zu groß. Die Kapazität eines Tracks reicht dann nicht mehr aus. In gewissen Grenzen kann nun diese GAP3 verringert werden. Bei einem Wert von &20 lassen sich i.A. diese 10 Sektoren problemlos formatieren. Man darf natürlich im Sinne der Datensicherheit diese GAP nicht unbegrenzt verkleinern. Der Wert von &20 erweist sich in der Praxis aber noch als recht datensicher.

|GAP3,byte

Dieser Befehl ist analog dem gerade beschriebenen FGAP-Befehl. Allerdings bezieht sich der hier angegebene Wert auf die Lese- und Schreiboperationen. Dieser Wert muß kleiner sein als der, der beim Formatieren der Diskette benutzt wurde! Bei dem oben beschriebenen Zehn-Sektor-Format ist z.B. ein Wert von &18 angebracht. Der Defaultwert ist &2A.

|GETDMA,@integer%

Mit diesem Befehl kann die aktuelle Adresse des Sektorpuffers in eine Integervariable geschrieben werden. Die Variable muß dabei mit dem Klammeraffen übergeben werden.

|HEAD,byte

Dieser Befehl ist in erster Linie nur für Besitzer von Doppelkopf-Laufwerken interessant. Durch den Parameter (nur 0 oder 1) kann der gewünschte Schreib-/Lesekopf ausgewählt werden. Die "normale" Seite (das ist die untere !) wird mit 0 angewählt, die andere mit 1.

|HOME

Der HOME-Befehl ist ein relativ wichtiger Befehl. Dieser Befehl ist der einzige, der den Kopf auf einen definierten Track positionieren kann. Dazu ein kurzer Einblick in den FDC. In dem FDC ist ein "Speicher" für die aktuelle Tracknummer. Jedesmal, wenn der FDC an das Laufwerk den Befehl 'Kopf einen Track weiter schieben' sendet, wird dieser Zähler erhöht. Wenn der Kopf in die andere Richtung bewegt wird, wird der Zähler erniedrigt. So weit, so gut - was ist allerdings, wenn nach dem Einschalten des Rechners der Kopf nicht auf Track 0 steht, oder falls gar durch Einschieben einer Diskette unglücklicherweise der Kopf ein Stück verschoben wird? Ganz einfach, dann wird die nächste Positionierung ein Ratespiel. Dieser Befehl ermöglicht es nun, den Kopf auf Track 0 zu positionieren. Wenn der Kopf nämlich Track 0 erreicht hat, dann wird dies durch eine Lichtschranke an den FDC übermittelt. Sofern dieser Befehl ausgeführt wurde, können Sie also ganz sicher sein, daß sich der Kopf auf Track 0 befindet. Hierzu noch eine Anmerkung für Besitzer von 80-Track-Laufwerken. Normalerweise bricht der FDC den HOME Befehl ab, wenn er nach 77 Step-Impulsen (d.h., jeweils eine Verschiebung um einen Track) kein Signal bekommt, daß der Track 0 erreicht ist. Der RSX-HOME berücksichtigt dies und führt eventuell zweimal dieses Kommando aus. So kann auch bei 80-Track- Laufwerken eine sichere Positionierung erreicht werden.

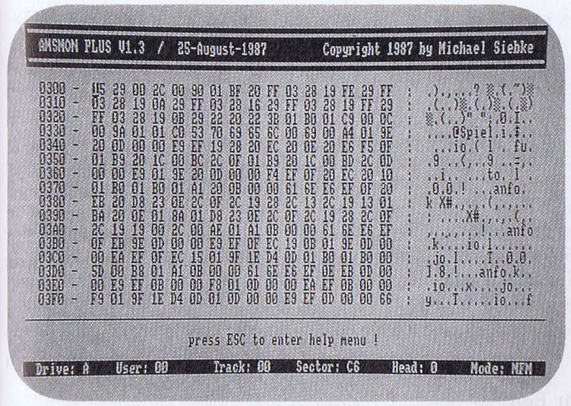


Abb. 1: Mit AMSMON PLUS wissen Sie genauestens über Ihre Disketten Bescheid. Der Inhalt wird in hexadezimaler Form ausgegeben.

|MF

Oben wurde schon erwähnt, daß der FDC mit zwei verschiedenen Aufzeichungsmodi arbeiten kann. Mit diesem Befehl kann nun die einfache Dichte (MF) ausgewählt werden.

|MFM

Dieser Befehl ist zu dem MF-RSX analog. Hier wird allerdings die doppelte Dichte ausgewählt. Dies ist die unter AMSDOS übliche Dichte.

|MSGOFF

Mit diesem Befehl werden alle Meldungen des BASIC-765 abgeschaltet. Dies kann zum Beispiel nützlich sein, wenn Sie auf der Basis dieser RSX-Erweiterung ein Disketten-Utility programmieren wollen. Sie müssen dann den Erfolg der Befehlsausführung aus der 'Result Phase' interpretieren. Bei AMSDOS liegt diese bei &BE4C. Die Länge der 'Result Phase' steht in &BE4B. AMSDOS Besitzer, die auch die AMSDOS-Fehlermeldungen unterdrücken wollen, können dies mit POKE &BE78,&FF erreichen. Mit POKE &BE78,&00 werden die AMSDOS- Meldungen wieder zugelassen.

|MSGON

Dieser Befehl läßt die BASIC-765 Meldungen, die mit MSGOFF abgeschaltet wurden, wieder zu.

|READ,sector
|READ,track,sector

Mit dem READ-Befehl wird der angegebene Sektor an die aktuelle Pufferadresse gelesen. Bezüglich der Parameterangabe gilt das gleiche wie beim |DREAD-Befehl.

|RESULT

Nach jeder Kommandoausführung sendet der FDC ein oder mehrere Bytes, die über den Erfolg der Ausführung Aufschluß geben. Die Information ist bitorientiert. So gibt z.B. nach den Sektor-R/W- Befehlen der erste Wert den Inhalt von Statusregister 0 des FDC an. Wenn dort das Bit 3 gesetzt ist, war das Laufwerk zum Zeit des Zugriffes nicht bereit; z.B. keine Diskette eingelegt.

|RSX765

Dieser Befehl dient zum Auflisten der kompletten Befehlsliste des BASIC-765. Bezüglich der Ausgabe Steuerung gilt das gleiche wie beim |ASCII-Befehl.

|SECTORS

Dies ist ein sehr hilfreicher Befehl. Es werden, sofern die Meldungen eingeschaltet sind, alle Sektornummern von dem Track ausgegeben. Gleichzeitig werden die Sektornummern an den Anfang des aktuellen Sektorpuffers geschrieben! In der ersten Speicherstelle wird die Anzahl der Sektoren auf dem Track vermerkt. Falls im Sektorpuffer noch Daten stehen, muß der Puffer vor der Ausführung des Befehles mit |SETDMA verlegt werden!

|SEEK,track

Dieser Befehl dient zum Positionieren des Kopfes über dem gewünschten Track. Die Trackangabe darf im Bereich 0...83 liegen, ansonsten wird eine Fehlermeldung ausgegeben.

|SETDMA,adresse

Mit diesem Befehl wird die Adresse des Sektorpuffers festgelegt. Nach der Initialisierung ist der Default &C000. Diese Adresse liegt im Bildschirmspeicher. Nach der Initialisierung ist daher mit dem MEMORY-Befehl die Speichergrenze tieferzusetzen und anschließend mit |SETDMA der Sektorpuffer anzugeben.

|SINFO,sector

Wenn Sie nähere Informationen über einen Sektor haben wollen, wird Ihnen dieser Befehl weiterhelfen. Es wird der Track, die Kopfnummer und die Sektorgröße aus der ID ausgegeben. Die Sektorgröße ist dabei verschlüsselt.

&00 = 0...255 Byte (i.A. 128)
&01 = 256 Byte
&02 = 512 Byte (AMSDOS !)
&03 = 1024Byte
&04= 2048 Byte
&05 = 4096 Byte

Als Defaultwert nach der Initialisierung ist die unter AMSDOS übliche Sektorgröße von 512 Byte eingestellt.

|SIZE,bytes

Mit diesem Befehl wird die Sektorgröße festgelegt. Die Sektorgröße wird in Bytes angegeben. Sie kann zwischen 0 und 255 liegen, oder 256, 512, 1024 2048 oder 4096 Bytes betragen. Die Sektorgröße 4096 Bytes ist allerdings im MF-Modus nicht benutzbar.

|STATUS

Mit |STATUS kann der Laufwerksstatus abgefragt werden. Es wird der Laufwerks-Fehlerstatus ausgegeben (von den meisten Laufwerken wird dies nicht unterstützt -> 0), der Zustand der Schreibschutz-Kontrolle, der Bereitschaftszustand, der Zustand der Track-Lichtschranke, das 'two side'-Signal sowie die aktuelle Kopf- und Laufwerksnummer. Das 'two side'-Signal gibt an, ob das Laufwerk ein Doppelkopflaufwerk ist oder nicht. Dies funktioniert allerdings nur, wenn keine schreibgeschützte Diskette eingelegt ist. Falls die Meldungen abgeschaltet sind, kann der Zustand mit dem |ERROR-Befehl abgefragt werden.

|STEP |STEP,ms

Es kann dem FDC mitgeteilt werden, wie schnell die Kopfpositionierung am Laufwerk durchgeführt werden kann. Bei den 3" Laufwerken von AMSTRAD ist ein Wert von 10 ms vorgegeben, das Schnellste (9 ms werden vom FDC auf 10 aufgerundet, da dieser die Steprate nur in 2 ms Schritten verändern kann). Die Steprate wird in Millisekunden angegeben und gilt bis zur nächsten Änderung für alle Laufwerke. Falls kein Parameter angegeben wird, wird die letzte Steprate neu gesetzt (dies kann notwendig sein, wenn andere Programme diese verändern).

|TFORMAT,spt
|TFORMAT,track,s1,s2,...

Mit diesem Befehl können Sie einen Track formatieren. Soweit der FDC mitspielt, gibt es für Sie keine Grenzen. Es gibt zwei Möglichkeiten der Ausführung. Wenn Sie nur einen Parameter angeben, wird dieser als SPT (Anzahl der Sektoren pro Track) angenommen. In diesem Fall müssen für jeden Sektor ab der aktuellen Sektorpufferadresse 4 Bytes stehen (Tracknr., Kopfnr., Sektornr., Sektorgröße). Wenn zwei oder mehr Parameter angegeben werden, wird der er-

ste als Tracknummer angenommen, die nachfolgenden geben die Sektornummern an. Als Kopfnummer und Sektorgröße werden die aktuellen Werte eingesetzt. Hier wird der Anfang des Sektorpuffers für die Formatiertabelle benutzt. Falls im Puffer wichtige Daten stehen, muß der Sektorpuffer vorher auf einen freien Bereich gesetzt werden.

|UNIT,drive
Wenn Sie auf ein anderes Laufwerk zugreifen wollen, müssen Sie diesen Befehl benutzen. Als Parameter wird das neue Bezugslaufwerk erwartet. Die Übergabe erfolgt als Zahlenwert zwischen 0 und 3 (Drive A = 0, Drive B = 1, ...). Die Laufwerke C und D können allerdings ohne Hardwareänderung nicht angesprochen werden. Wenn Drive C (=2) angewählt wird, fühlt sich wieder Laufwerk A zuständig. Dies liegt an der unvollständigen Laufwerksdekodierung im Floppy-Controller.

|VERIFY,sector
|VERIFY,track,sector
Mit dem |VERIFY-Befehl können Sie feststellen, ob der Inhalt des Sektorpuffers mit dem angegebenen Sektor übereinstimmt. Der Vergleich wird direkt vom FDC durchgeführt. Nach der Ausführung erfolgt bei eingeschalteten Meldungen das Vergleichsergebnis. Ansonsten muß anhand des Statusregister 2 (&BE4E), Bit 3 geprüft werden, ob Gleichheit vorliegt (Bit gesetzt). Bezüglich der Parameterübergabe gilt das beim |DREAD-Befehl Gesagte.

|WRITE,sector |WRITE,track,sector
Dieser Befehl schreibt einen Sektor ganz normal (im Gegensatz zu |DWRITE) auf Diskette zurück. Für die Parameterübergabe gilt das gleiche wie beim |DREAD-Befehl.

Beispiele zu Programmierung

Soweit zur Befehlserklärung. Nun zwei konkrete Beispiele. Als erstes wollen wir den Konfigurationsektor von Laufwerk A (natürlich von einer Backup !!!) lesen und ansehen. Ich möchte erstmal davon ausgehen, daß die Erweiterung frisch initialisiert wurde. Als erstes wird das Laufwerk initialisiert (nicht unbedingt notwendig, da AMSDOS diese Arbeit schon übernommen hat, aber doppelt schadet nicht):

```
10 |HOME
```

Mit dem |HOME-Befehl wurde der Kopf auf Track 0 positioniert. Dies können Sie einfach mit dem Befehl |STATUS abprüfen:

```
20 |STATUS
```

Bei der Angabe "Track 0" muß jetzt eine 1 stehen. Nun muß ein Datenpuffer reserviert werden. Dies kann z.B. so geschehen:

```
30 MEMORY&8FFF'ohneRSX!
40 |SETDMA,&9000 'mit RSX!
```

Nun liegt der Sektorpuffer an Adresse &9000. Mit dem |READ-Befehl wird nun der gewünschte Sektor eingelesen:

```
50 |READ,&42
```

Die Tracknummer braucht ja nicht angegeben zu werden; sie stimmt unter AMSDOS mit der physikalischen überein - die wurde aber gerade mit dem |HOME-Befehl gesetzt. Mit dem |DUMP-Befehl können Sie sich nun Ihren Sektor ansehen:

```
60 |DUMP
```

Falls Sie vor diesem Beispiel bereits mehrere Kommandos ausgeführt haben, sollten Sie vor dem |HOME- Befehl folgende Kommandos eingeben:

```
|UNIT,0'DriveA anwählen
|HEAD,0 'untererKopf |SIZE,512'512Byte
```

Die Sektoren

Nun zum zweiten Beispiel. Sie wollen eine komplette Diskette mit 10 Sektoren pro Track formatieren. Ich möchte jetzt nicht mehr jeden Befehl erläutern. Sie können sich den Ablauf anhand der Befehlsbeschreibung sicher selbst klarmachen. (Befehle des BASIC-765 müssen mit vorangestelltem RSX-Zeichen eingegeben werden!):

```
100 |UNIT,0:|HEAD,0:|HOME:|MFM:PRINT"
    Diskette in Laufwerk A einlegen!"
110 |GAP,&20:|GAP3,&18:MEMORY &
    7FFF:|SETDMA,&8000:|MSGOFF
120 IF INKEY$<>"" THEN GOTO 120
130 IF INKEY$="" THEN GOTO 130
140 FOR track=0 to 42
150 |SEEK,track
160 |TFORMAT,track,&41,&42,&43,&
    44,&45,&46,&47,&48,&49,&4A
170 NEXT track 180 PRINT "Formatierung
    fertig!"
190 |HOME
```

Das Ganze ist natürlich noch äußerst unkomfortabel, aber Sie können sich ja nun Ihr persönliches Format-Programm zusammenstricken. Allerdings werden die zusätzlichen Sektoren nicht ohne weiteres von AMSDOS unterstützt. Dazu ist eine Änderung im DPB notwendig, auf die ich hier allerdings nicht näher eingehen möchte.

Für Freunde des VDOS...

Nun noch die versprochene Anpassungsmöglichkeit für andere DOS-Versionen (Hallo VDOS-Freak...). Ich muß nun geringe Assembler-Kenntnisse voraussetzen – diese sind für die Anpassung sowieso notwendig.
Wer sich die BASIC-Erweiterung mit einem Disassembler anschaut, wird feststellen, daß sie ziemlich seltsam anfängt – im Prinzip scheint dort Müll zu stehen. Was Sie dort sehen, ist die Patch-Area zur Anpassung an andere DOS-Versionen. Da BASIC-765 fast die komplette FDC-Programmierung selbst übernimmt, werden nur fünf Routinen aus dem DOS benutzt. Dies sind Routinen zur Synchronisation mit den interruptgesteuerten Motorroutinen, sowie zum Senden eines Bytes an den FDC und zum Auslesen der 'Result Phase'. Alle diese Routinen werden über den Restart &18 aufgerufen. Dieser erwartet eine Adresse auf einen drei Byte langen Parameterblock, in dem die Adresse der Routine und die ROM-Nummer steht. Genau diese drei Bytes finden Sie am Anfang der Erweiterung:

&9E80: Adr.Motoreinschalten
&9E82: ROM Nummer
&9E83: Adr. Motor aus (DE=delay)
&9E85: ROM-Nummer
&9E86: Adr.Motor Event aus
&9E88: ROM-Nummer
&9E89: Byte an FDC senden
&9E8B: ROM-Nummer
&9E8C: Result Phase nach &BE4C
&9E8E: ROM-Nummer

Sofern Sie die entsprechenden Adressen im VDOS-ROM kennen, brauchen Sie diese nur in den entsprechenden Speicherstellen zu vermerken. Ansonsten müssen Sie eigene Routinen im RAM aufrufen (ROM Nummer = &FF!). Dazu möchte ich nun eine Primitiv-Lösung vorschlagen, die Sie natürlich entsprechend erweitern können.

```
1. Motor einschalten
    PUSH    BC
    LD      BC ,&FA7F
    OUT     (C),C
    POP     BC
    PUSH    AF
    LD      A ,&FF
    LD(&BE5F),A ;Motorflag
    POP     AF
    RET

2. Motor ausschalten
    PUSH    BC
    LD      BC,&FA7E
    OUT     (C),C
W1: CALL    &BD19
    DJNZ    W1
    POP     BC
    PUSH    AF
    XOR     A
    LD (&BE5F)  ,A
    POP     AF
    RET

3. Motor Event ausschalten
    RET

4. Byte an FDC senden
    PUSH    AF
    PUSH    AF
L1: IN      A,(C)
    ADD     A,A
    JR      NC,L1
    ADD     A,A
    JR      NC,L2
    POP     AF
    POP     AF
    RET

L2: POP     AF
    INC     C
    OUT     (C),A
    DEC     C
    LD      A,&05
L3: DEC     A
    NOP
    JR      NZ,L3
    POP     AF
    RET

5. Result Phase auslesen
    PUSH    HL
    PUSH    DE
    LD      D,&00
    LD      HL,&BE4C
    PUSH    HL
R1: IN      A,(C)
    CP      &C0
    JR      C,R1
    INC     C
    IN      A,(C)
    DEC     C
    LD      (HL),A
    INC     HL
    INC     D
    LD      A,&05
R2: DEC     A
    JR      NZ,R2
    IN      A,(C)
    AND     &10
    JR      NZ,R1
    POP     HL
    LD      A,(HL)
    AND     &C0
    DEC     HL
    LD      (HL),D
    POP     DE
    POP     HL
    RET
    NZ
    SCF
    RET
```

Diese Routinen müssen Sie im RAM unterbringen und die entsprechenden Adressen in der Patch-Area eintragen. In dieser Primitiv-Lösung steckt allerdings ein Nachteil. Die Motorroutinen sind einmal nicht mehr interruptgesteuert (häufiges Motor Ein-/Ausschalten mit Wartezeit) und zum anderen nicht mit den VDOS Routinen synchronisiert. Sie müssen jeweils warten, bis VDOS den Motor abgeschaltet hat, damit er nicht während der Ausführungsphase über den VDOS Interrupt abgeschaltet wird.

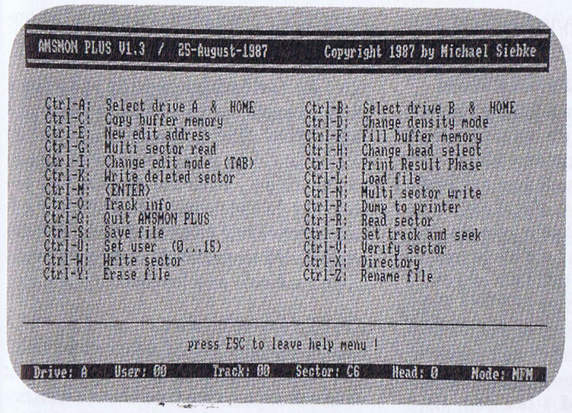


Abb. 2: Auf der HELP-Seite von AMSMON (mit ESC anwählbar) haben Sie einen Überblick über alle Befehle.

Systemadressen

Zu guter Letzt noch die wenigen Systemadressen, die BASIC-765 aus dem DOS-RAM-Bereich benutzt:
&BE46: Motor aus Verzögerung
&BE4B: Anzahl Bytes aus ResultPh.
&BE4C: Result Phase
&BE5F: Motor Flag
Vielleicht gelingt einem VDOS-Freak ja eine optimale Anpassung...
So, und nun kommen wir zu AMSMON PLUS.

AMSMON PLUS

– Mikroskop für Disketten

Dieser Diskettenmonitor benutzt wie BASIC-765 eigene FDC-Routinen. Im Vergleich zu der RSX-Erweiterung sind die Routinen allerdings weitaus intelligenter. Sie nehmen Ihnen praktisch die meiste Arbeit ab. Von nun ab kann es Ihnen (fast) egal sein, wie groß ein Sektor ist. Nach Angabe der Sektornummer sucht der Monitor aus dem aktuellen Track nach dem Sektor und stellt sich automatisch auf die Sektorgröße und andere Parameter aus der ID ein! Allerdings kennt AMSMON PLUS keine Formatierroutinen.
AMSMON PLUS wird in den Rechner "gebootet", d.h., es findet eine komplette Initialisierung des Betriebssystems statt. Dies ist wegen den zahlreichen Eingriffen in das Betriebssystem notwendig, damit der Monitor eine definierte Umgebung vorfindet. Eine Anpassung an die verschiedenen CPC-Versionen erfolgt automatisch.
Das Programm belegt den Speicherplatz ab &9000. Um in den Monitor zu gelangen, brauchen Sie nur *RUN "AMSMON* einzugeben. Anschließend meldet sich der Diskettenmonitor - Sie befinden sich im Editor. Vielleicht ist Ihnen beim Bildschirmaufbau schon etwas aufgefallen, sofern Sie den Monitor schon lauffähig zur Verfügung haben. Die Bildschirmausgabe im Monitor erfolgt nämlich mit einer eigenen FASTPRINT-Routine, die mehr als doppelt so schnell wie die Original-Routine ist. Doch nun zum Aufbau des Bildschirmes.

Der Disk-Monitor und seine Möglichkeiten

Ganz unten befindet sich die Statuszeile. In dieser werden die aktuellen Daten für Drive, User, Track, Sektor, Kopf und Aufzeichnungsmodus angegeben. Direkt über dieser Zeile befindet sich die Kommandozeile. In dieser werden weitere Eingaben erwartet, sowie wichtige Meldungen ausgegeben. Der größte Teil des Bildschirmes ist dem Editor gewidmet. Dieser stellt ein 16 * 16 Byte großes Hex- und ASCII Feld zum Editieren zur Verfügung. Nach der Initialisierung steht der Cursor links oben im Hexfeld. Mit der TAB-Taste können Sie zwischen dem Hex- und dem ASCII-Feld wechseln. Den Cursor selbst bewegen Sie mit den Cursortasten. Wenn diese zusammen mit Shift betätigt werden, wird der Cursor an den jeweiligen Rand des Feldes gesetzt. Die Kombination der Ctrl-Taste mit Cursor-links oder Cursor-rechts ermöglicht es Ihnen, an den Anfang (links) oder an das Ende (rechts) des Pufferbereiches zu wandern, den Ihnen AMSMON PLUS zur Verfügung stellt. Mit der Kombination Ctrl-Taste und Cursor-auf oder Cursor-ab blättern Sie jeweils eine Seite im Speicher zurück oder hoch.
Die am Rand angegebenen Adressen sind die absoluten Adressen im Speicher. Der Pufferbereich reicht von &0200 bis &87FF. Dahinter liegt ein 2048 Byte großer Bereich, den der Monitor für Dateioperationen benötigt. Der Pufferbereich von AMSDOS liegt nun im Bereich von &8000 bis &8FFF.
Aufgrund der Tatsache, daß Sie den Monitor verlassen und später mit dem RSX-Befehl AMSMON wieder aufrufen können, müssen Sie Folgendes beachten: Nach Aufruf des Monitors ist der Pufferbereich nicht mit dem MEMORY-Befehl reserviert. Falls Sie also zeitweise im BASIC arbeiten, müssen Sie den benutzten Pufferbereich mit dem MEMORY-Befehl schützen. Solange Sie sich allerdings im Monitor befinden, übernimmt AMSMON PLUS die Speicherverwaltung.
Noch eine Bemerkung zum Editor. Teilweise ist es störend, daß der Editor immer aktiv ist. Dies ist z.B. der Fall, wenn gerade wichtige Daten im Puffer sind und man nicht durch einen versehentli-

chen Tastendruck diese ungewollt verändern möchte. Deshalb läßt sich der Editor mit CAPS LOCK sperren. Als Kennzeichen dieser Sperre ist die Randfarbe nun dunkel. Mit einem erneuten Drücken der CAPS LOCK- Taste läßt sich die Sperre wieder aufheben.

Die Befehle

Kommen wir nun zu den Befehlen, die Ihnen AMSMON PLUS bietet. Eine Befehlsübersicht erhalten Sie durch Drücken der ESC-Taste. Alle Befehle werden durch eine Ctrl-Funktion ausgeführt, d.h., Sie müssen die entsprechende Taste zusammen mit der Ctrl-Taste drücken. Durch erneutes Drücken der ESC-Taste verlassen Sie die Hilfsseite. Bei allen Befehlen erfolgt eine Überprüfung der Eingaben. Die Befehle können nur auf den Pufferbereich angewendet werden!

Ctrl-A: Laufwerk A anwählen
Mit diesem Kommando wird Laufwerk A zum aktiven Laufwerk. Gleichzeitig wird ein HOME ausgeführt, um den Kopf in eine definierte Position zu bringen (siehe auch Beschreibung des HOME Befehls bei BASIC-765). Im übrigen wird auch AMSDOS über das neue Laufwerk informiert.

Ctrl-B: Laufwerk B anwählen
Hier wird die beim Ctrl-A Kommando beschriebene Funktion für Laufwerk B ausgeführt.

Ctrl-C: Pufferbereich kopieren
Diese Routine dient zum Kopieren eines Pufferbereiches an eine andere Stelle im Puffer. Es wird die erste und die letzte Adresse des Quellblockes abgefragt, anschließend die Zieladresse. Nach der Ausführung wird das Editierfeld neu aufgebaut. Die Routine arbeitet intelligent, d.h., auch sich überlappende Bereiche werden richtig kopiert.

Ctrl-D: Aufzeichnungsdichte ändern
Wie in der Einleitung erwähnt, kann der FDC in zwei verschiedenen Aufzeichnungsmodi arbeiten. Mit diesem Befehl wird zwischen den beiden Modi umgeschaltet. Der aktuelle Modus wird in der Statuszeile angezeigt.

Ctrl-E: Neue Editadresse
Wenn Sie zu einer bestimmten Stelle im Puffer wollen, können Sie diese Adresse hier angeben. Das Editierfeld wird ab dieser Adresse neu aufgebaut. Die Adresse muß im Bereich &0200... &8700 liegen.

Ctrl-F: Speicherbereich füllen
Wie beim Ctrl-C-Kommando wird zuerst die Anfangs- und Endadresse des Blockes abgefragt. Anschließend muß das Füllbyte angegeben werden.

Ctrl-G: Folge von Sektoren lesen
Dieses Kommando (und Ctrl-N) benutzt ausnahmsweise noch die DOS-Routinen für den Sektor Transfer. Mit beiden Kommandos kann eine Folge von Sektoren bearbeitet werden. Mit Ctrl-G werden Sektoren in den Puffer gelesen. Beide Befehle können nur auf die Standardformate System,Data und IBM angewendet werden! Es wird zuerst der Track abgefragt, auf dem der erste Sektor liegt. Danach wird nach dem ersten logischen Sektor gefragt. Dieser logische Sektor liegt zwischen 1 und 8 bei dem IBM-Format, bei den anderen zwischen 1 und 9. Der Sektoroffset wird vom Monitor verarbeitet. Zum Schluß wird nach der Anzahl der Sektoren gefragt, die gelesen (bei Ctrl-N geschrieben) werden sollen. Wenn der letzte Sektor auf dem Track verarbeitet ist, wird mit dem ersten Sektor auf dem nächsten Track fortgefahren. Dies geschieht solange, bis die gewünschte Anzahl von Sektoren gelesen (geschrieben) wurde, ein Fehler auftrat oder das Ende des Puffers erreicht wurde. In der Statuszeile wird nach Ausführung des Kommandos der Sektor angegeben, der dem zuletzt Gelesenen folgt. Sie können sich mit diesem Befehl sehr einfach die Systemspuren einlesen, bearbeiten und zurückschreiben (Track 0, log. Sektor 1, 18 Sektoren).

Ctrl-H: Kopfnummer wählen
Besitzer von Zweikopflaufwerken können mit diesem Befehl auf den zweiten Kopf umschalten. Eine Anzeige des aktuellen Kopfes erfolgt in der Statuszeile.

Ctrl-I: Wechsel Hex-/ASCII-Modus
Dieser Tastencode entspricht TAB. Es kann im Editor zwischen dem Hex- und dem ASCII-Feld umgeschaltet werden.

Ctrl-J: Result Phase ausgeben
Dieser Befehl ist dann sehr nützlich, wenn Sie z.B. nähere Informationen über den zuletzt gelesenen Sektor haben wollen. Nach dem Lesen rufen Sie diesen Befehl auf. Es werden Ihnen sieben Bytes hexadezimal angezeigt. Die ersten drei geben den Inhalt der FDC Statusregister an und sind für Sie im allgemeinen uninteressant, was danach kommt ist viel wichtiger. Das vierte Byte gibt die Tracknummer an, die in der Sektor-ID des gerade gelesenen Sektors steht (AMSMON PLUS verarbeitet auch Sektoren, bei denen diese Tracknummer von der physikalischen verschieden ist, s.o.), dahinter folgen die Kopfnummer und die Sektornummer. Das letzte Byte gibt die Sektorgröße an. Diese Angabe ist verschlüsselt. Die Umrechnungstabelle ist beim SINFO-Befehl des BASIC-765 angegeben.

Ctrl-K: Gelöschten Sektor schreiben
Nach Angabe der Sektornummer sucht AMSMON PLUS einen existierenden Sektor auf dem aktuellen Track. Wenn er ihn findet, wird der Sektor mit gelöschter DAM (s.o.) auf Diskette geschrieben. Die Daten werden ab der aktuellen Editieradresse (Adresse links oben im Feld) geschrieben.

Ctrl-L: Datei laden
Sie können mit dem Monitor auch Dateien verarbeiten. Diese Routine ermöglicht es Ihnen sogar, aus einer Datei einen beliebigen Ausschnitt zu laden! Nach Eingabe des Dateinamens versucht AMSMON PLUS die Datei zu öffnen. Wenn dies glückt (es werden nur Binärdateien angenommen), fragt der Monitor nach einem Offset. Hier können Sie eine Anzahl von Bytes angeben, die beim Laden übergangen werden sollen. Der Offset muß natürlich kleiner als die Dateilänge sein. Anschließend fragt der Monitor die Anzahl der Bytes ab, die in den Puffer geladen werden sollen. Zum Schluß können Sie noch die Ladeadresse wählen. Wenn Sie keinen Offset (also 0) angeben und die komplette Datei laden, verwendet AMSMON PLUS die schnelle DOS Routine, ansonsten wird die Datei byteweise geladen, was natürlich bedeutend länger dauert.

Ctrl-N: Folge v. Sektoren schreiben
siehe Ctrl-G Kommando

Ctrl-O: Track Info
Diese Routine gibt alle Sektornummern aus, die auf dem aktuellen Track vorhanden sind.

Ctrl-P: Druckerausgabe
Mit diesem Kommando können Sie einen Pufferbereich auf dem Drucker ausgeben. Es werden die Bereichsgrenzen abgefragt. Die Druckroutine ist jederzeit, auch bei nicht angeschlossenem Drucker, unterbrechbar.

Ctrl-Q: AMSMON PLUS verlassen
Mit diesem Befehl gelangen Sie in das BASIC zurück. Mit dem RSX Kommando AMSMON können Sie später wieder in den Monitor zurückkehren. Bitte denken Sie daran, evtl. Daten im Puffer mit dem MEMORY-Befehl zu sichern (s.o.).

Ctrl-R: Sektor lesen
Der Ablauf ist der gleiche wie beim Ctrl-K Kommando. Hier wird der gewünschte Sektor gelesen. Es können auch Sektoren mit gelöschter DAM (s.o.) eingelesen werden, in diesem Fall erfolgt in der Statuszeile eine entsprechende Meldung.

Ctrl-S: Datei speichern
Hiermit können Sie einen Pufferbereich auf Diskette sichern. Es wird der Bereich und der Dateiname abgefragt.

Ctrl-T: Track anfahren
Mit diesem Kommando können Sie den Kopf über einem beliebigen Track positionieren (0...83).

Ctrl-U: User auswählen
Die Funktion dürfte klar sein: es kann ein neuer User angegeben werden. Alle folgenden Dateioperationen beziehen sich dann auf den neuen User. Anmerkung: Da AMSMON PLUS die AMSDOS Systemadressen benutzt, bleibt der User-Bereich auch nach Verlassen des Monitors aktiv.

Ctrl-V: Sektor vergleichen
Der Funktionsablauf ist wie beim Ctrl-R-Kommando. Nur wird hier der Sektor nicht gelesen, sondern der FDC vergleicht die Daten im Puffer mit dem angegebenen Sektor. Wie bei allen sektororientierten Befehlen wird die aktuelle Editieradresse als erste Datenadresse benutzt.

Ctrl-W: Sektor schreiben
Wie Ctrl-K, allerdings ganz normal, ohne gelöschte DAM.

Ctrl-X: Directory anzeigen
Es wird eine DIR-Maske abgefragt. Wenn keine Maske angegeben wird, erfolgt die Ausgabe sortiert mit Größenangabe, ansonsten werden nur die Dateien ausgegeben, die der Maske entsprechen.

Ctrl-Y: Datei löschen
Nach Angabe des Dateinamens (auch Wildcards zulässig) wird die entsprechende Datei gelöscht.

Ctrl-Z: Datei umbenennen
Auch dieser Befehl dürfte klar sein, er dient dazu, einer Datei einen neuen Namen zu geben.

WICHTIG:
Eine allgemeine Anmerkung!

Durch den großen Pufferbereich ist es möglich, mehrere Sektoren und/oder Dateien im Puffer zu halten. Die sektororientierten Befehle arbeiten nun immer bezüglich der aktuellen Editieradresse. Wenn Sie also einen bearbeiteten Sektor auf Diskette zurückschreiben wollen, müssen Sie, sofern Sie nicht schon dort sind, erst wieder auf die ursprüngliche Ladeadresse zurückkehren! Es empfiehlt sich deshalb, einen Sektor entweder auf den Pufferanfang &0200 zu laden oder auf eine gerade Adresse wie &1000, &2000,... So kann man sich die Adresse problemlos merken.

Zu den Programmen

So, nun sind Sie an der Reihe. Wenn Sie nun mit BASIC-765 oder/und AMSMON PLUS arbeiten wollen, müssen Sie nur noch den jeweiligen DATA-Lader eintippen. Beide Lader arbeiten mit einer aufwendigen Prüfsumme, die sich auch von vertauschten oder vergessenen Zeilen nicht täuschen läßt. Auch Vertauschungen innerhalb einer Zeile werden erkannt. Sie können so sicher sein, daß bei fehlerfreiem Durchlauf des BASIC-Laders das jeweilige Programm auch einwandfrei läuft. Die Lader speichern das Programm anschließend auf Diskette ab. Zum Schluß noch eine Bitte. Wenn Sie die Programme ausprobieren - tun Sie dies bitte mit einer Backup-Diskette und nicht mit Ihrer wichtigsten Daten-Diskette, Sie wissen schon...

(Michael Siebke/jb)

für 464-664-6128

```
1000 '================================    [2011]
1010 '**                              **    [104]
1020 '**  AMSMON PLUS  Version 1.3    **    [1124]
1030 '**  ------------------------    **    [819]
1040 '**                              **    [104]
1050 '**       August 1987 M.Siebke   **    [1153]
1060 '**                              **    [104]
1070 '================================    [2011]
1080 :                                      [174]
1090 MODE 2: MEMORY &8FFF: zz=0             [2222]
1100 PRINT "Bitte warten...": PRINT         [3056]
1110 :                                      [174]
1120 FOR adr=&9000 TO &A58F STEP 8          [1910]
1130   :                                    [174]
1140   PRINT "*";: s=0                      [709]
1150   FOR i=0 TO 7                         [486]
1160     READ b$: b=VAL("&"+b$)             [1052]
1170     s=s+b XOR i: POKE adr+i,b          [1100]
1180   NEXT                                 [350]
1190   READ c$: c=VAL("&"+c$)               [1443]
1200   s=s+16*((zz XOR &55) MOD 256)        [1376]
1210   s=s+    (zz XOR &AA) MOD 256         [1985]
1220   zz=zz+10: IF c=s GOTO 1260           [1889]
1230   PRINT " DATA Fehler in Zeile";       [1766]
1240   PRINT zz+1290: STOP                  [625]
1250   :                                    [174]
1260 NEXT                                   [350]
1270 :                                      [174]
1280 SAVE "AMSMON",b,&9000,&1590,&9000      [1365]
1290 :                                      [174]
```

Listing. "AMSMON PLUS"

```
1300 DATA 21,06,91,0E,FF,3A,02,A7,0896    [1688]
1310 DATA 32,E5,91,C3,16,BD,13,90,0A7B    [1964]
1320 DATA C3,44,91,41,4D,53,4D,4F,07E7    [1577]
1330 DATA CE,00,00,00,00,00,1C,00,064E    [1818]
1340 DATA 14,14,1C,01,00,00,1D,14,08C4    [2128]
1350 DATA 14,04,02,18,92,FF,4E,9A,09B7    [2292]
1360 DATA 98,20,20,41,4D,53,4D,4F,0981    [1095]
1370 DATA 4E,20,50,4C,55,53,20,56,0440    [2424]
1380 DATA 31,2E,33,20,20,2F,20,20,0291    [1056]
1390 DATA 32,35,2D,41,75,67,75,73,047D    [1697]
1400 DATA 74,2D,31,39,38,37,FF,09,0664    [2107]
1410 DATA 20,43,6F,70,79,72,69,67,076F    [1595]
1420 DATA 68,74,20,31,39,38,37,20,059F    [1265]
1430 DATA 62,79,20,4D,69,63,68,61,1069    [1521]
1440 DATA 65,6C,20,53,69,65,62,6B,1091    [1079]
1450 DATA 65,20,20,92,FF,4E,9A,98,1026    [796]
1460 DATA 18,1F,01,15,FF,50,5F,F8,124B    [1710]
1470 DATA 1F,01,19,18,20,20,44,72,1135    [1551]
1480 DATA 69,76,65,3A,20,F9,20,20,10FF    [954]
1490 DATA 20,20,55,73,65,72,3A,20,10F9    [1217]
1500 DATA FC,20,20,20,20,20,20,20,0C0E    [942]
1510 DATA 54,72,61,63,6B,3A,20,FA,0C31    [1569]
1520 DATA 20,20,20,20,53,65,63,74,0B0F    [1098]
1530 DATA 6F,72,3A,20,FB,20,20,20,0E12    [2337]
1540 DATA 20,20,48,65,61,64,3A,20,0CB4    [1120]
1550 DATA FD,20,20,20,20,20,4D,6F,0D99    [1015]
1560 DATA 64,65,3A,20,FE,12,18,F8,0901    [553]
1570 DATA 1F,01,18,FF,50,5F,1F,01,085C    [1172]
1580 DATA 17,12,1F,1A,17,70,72,65,0732    [1602]
1590 DATA 73,73,20,45,53,43,20,74,0A63    [1023]
1600 DATA 6F,20,65,6E,74,65,72,20,0AE5    [2142]
1610 DATA 68,65,6C,70,20,6D,65,6E,09E3    [1531]
1620 DATA 75,20,21,F8,00,00,3E,C9,04EB    [1790]
1630 DATA 32,5A,BB,11,40,00,21,FF,058C    [1981]
1640 DATA AB,CD,CB,BC,3E,CF,32,5A,05B4    [2698]
```

Listing. "AMSMON PLUS"

```
1650 DATA BB,11,FF,8F,3E,02,CD,A0,05B1    [1185]
1660 DATA 97,CD,E6,91,CD,E6,91,11,09C0    [1578]
1670 DATA FF,7F,3A,39,00,FE,39,21,0693    [2019]
1680 DATA 92,B0,28,04,13,21,76,B0,062E    [1397]
1690 DATA CD,E6,91,21,1A,90,01,0E,1078    [1104]
1700 DATA 90,CD,D1,BC,3E,02,CD,0E,107B    [1333]
1710 DATA BC,CD,02,BC,01,1A,1A,3E,0FDE    [955]
1720 DATA 01,CD,32,BC,3E,FF,32,78,12C1    [830]
1730 DATA BE,3E,03,32,66,BE,3E,04,1251    [1639]
1740 DATA CD,A0,97,22,04,91,21,88,123E    [1808]
1750 DATA 94,22,D4,BD,21,1E,90,CD,0DBF    [916]
1760 DATA C7,93,0E,07,CD,0F,B9,21,0D19    [1642]
1770 DATA 00,00,11,18,4F,CD,66,BB,0B16    [1482]
1780 DATA AF,32,BD,92,32,BE,92,CD,1019    [984]
1790 DATA C1,94,3A,E5,91,FE,FF,28,1156    [2066]
1800 DATA 03,CD,DA,97,3E,FF,32,E5,0F0B    [1722]
1810 DATA 91,21,D8,90,CD,C7,93,31,0F76    [1686]
1820 DATA 00,C0,CD,7D,BC,CD,92,BC,0B43    [1825]
1830 DATA CD,8D,96,21,88,90,CD,C7,09EB    [2028]
1840 DATA 93,CD,DC,92,CD,8A,BB,CD,0AF3    [1171]
1850 DATA 7F,97,4F,CD,8D,BB,79,FE,0CAB    [2709]
1860 DATA F0,30,05,FE,20,D2,5E,92,0AF9    [1380]
1870 DATA DD,21,EB,91,DD,7E,00,B9,0C0C    [1291]
1880 DATA 28,0C,A7,CA,A9,92,DD,23,05E0    [1353]
1890 DATA DD,23,DD,23,18,EE,DD,6E,06DF    [749]
1900 DATA 01,DD,66,02,E9,00,73,23,0483    [1472]
1910 DATA 72,23,C9,01,D5,97,02,D8,07E9    [1219]
1920 DATA 97,03,9F,98,04,F5,98,05,07C5    [1063]
1930 DATA 3D,99,06,FD,98,07,02,9B,0631    [2011]
1940 DATA 08,33,99,09,3E,93,0A,2D,0F59    [1707]
1950 DATA A3,0B,48,A3,0C,71,99,0E,10CD    [1204]
1960 DATA 09,9B,0F,CE,9C,10,73,9D,0F95    [995]
1970 DATA 11,02,9E,12,26,A3,13,61,0EE0    [1029]
1980 DATA 9E,14,D3,9E,15,FD,9F,16,13B4    [1681]
1990 DATA 4F,A3,17,41,A3,18,18,9F,1146    [1125]
2000 DATA 19,5C,9F,1A,B0,9F,F0,55,1260    [1176]
2010 DATA 93,F1,6C,93,F2,48,93,F3,0EE9    [1540]
2020 DATA 61,93,F4,72,93,F5,75,93,0DBC    [1484]
2030 DATA F6,79,93,F7,7F,93,F8,83,0EE0    [1579]
2040 DATA 93,F9,90,93,FA,96,93,FB,1135    [872]
2050 DATA 9B,93,FC,1F,A0,00,3E,03,0F2A    [1260]
2060 DATA CD,A0,97,7E,CB,7F,C2,A9,1061    [2314]
2070 DATA 92,3A,BF,92,A7,79,20,41,09BA    [1586]
2080 DATA CD,25,93,30,34,CD,7F,94,0A0B    [1486]
2090 DATA 5F,CD,C0,92,E6,0F,B3,77,0995    [2254]
2100 DATA CD,02,93,CD,D6,92,24,CD,0C5C    [1145]
2110 DATA 75,BB,CD,8A,BB,CD,7F,97,0D9B    [2173]
2120 DATA FE,0D,28,05,CD,25,93,30,099B    [1535]
2130 DATA F4,F5,CD,8D,BB,F1,30,1A,0C7D    [1962]
2140 DATA 5F,CD,C0,92,E6,F0,B3,18,07D7    [2550]
2150 DATA 0E,3E,07,CD,5A,BB,C3,9F,0507    [2034]
2160 DATA 91,CB,BF,FE,20,38,F2,CD,06C2    [2146]
2170 DATA 02,93,C3,61,93,00,00,00,0644    [1099]
2180 DATA D5,3A,BE,92,2A,17,95,16,067F    [1949]
2190 DATA 00,CD,7F,94,5F,3A,BD,92,0780    [1705]
2200 DATA 83,5F,19,D1,7E,C9,F5,3A,117A    [1202]
2210 DATA BD,92,18,0A,F5,3A,BF,92,11C7    [1378]
2220 DATA A7,3A,BD,92,20,18,4F,CB,1082    [1937]
2230 DATA 27,81,C6,0B,67,3A,BE,92,12E8    [1766]
2240 DATA C6,05,6F,E5,CD,75,BB,E1,149B    [1167]
2250 DATA F1,C9,F5,3A,BD,92,C6,3F,1391    [1369]
2260 DATA 18,EA,CD,D6,92,4F,CD,69,0E7C    [1073]
2270 DATA 94,CD,FA,92,CD,C0,92,79,0FDF    [1635]
2280 DATA 77,C3,19,95,CD,7F,97,FE,0D5D    [1902]
2290 DATA FC,CA,94,91,FE,0D,C8,CD,0EB3    [1241]
2300 DATA 25,93,30,F0,C9,FE,30,38,1021    [2235]
2310 DATA 13,FE,3A,38,0B,CB,AF,FE,0EB8    [2044]
2320 DATA 41,38,09,FE,47,D0,D6,07,0E4E    [1779]
2330 DATA D6,30,37,C9,A7,C9,21,BF,0A36    [2240]
2340 DATA 92,3E,01,AE,77,C3,9F,91,08ED    [1493]
2350 DATA 3A,BD,92,3D,E6,0F,32,BD,094E    [1835]
2360 DATA 92,FE,0F,20,09,3A,BE,92,0AF6    [1410]
2370 DATA 3D,E6,0F,32,BE,92,C3,9F,0C50    [1962]
2380 DATA 91,3A,BD,92,3C,E6,0F,32,0ADF    [1226]
2390 DATA BD,92,20,F2,3A,BE,92,3C,0673    [1479]
2400 DATA 18,E7,AF,18,E4,3E,0F,18,058D    [1265]
2410 DATA E0,AF,32,BD,92,18,DF,3E,0565    [2597]
2420 DATA 0F,18,F7,2A,17,95,25,CD,0702    [1629]
2430 DATA A0,93,22,17,95,C3,77,91,0886    [2270]
```

Listing. "AMSMON PLUS"

```
2440 DATA 2A,17,95,24,18,F1,21,00,050E    [1741]
2450 DATA 02,18,EF,21,00,87,18,EA,0643    [846]
2460 DATA C5,01,00,87,D5,E5,A7,11,11A7    [1534]
2470 DATA 00,02,ED,52,38,0D,D1,60,0F67    [1341]
2480 DATA 69,ED,52,30,02,50,59,EB,1030    [880]
2490 DATA D1,C1,C9,E1,18,F9,C5,01,145B    [1428]
2500 DATA FF,87,18,E0,00,00,40,7E,119A    [581]
2510 DATA 23,FE,F8,C8,DC,5A,BB,38,1408    [1533]
2520 DATA F6,E5,D5,D6,F8,21,E5,93,0F87    [1824]
2530 DATA 3D,28,04,23,23,18,F9,5E,0C36    [2296]
2540 DATA 23,56,EB,D1,E9,F3,93,FE,0EDA    [1004]
2550 DATA 93,10,94,18,94,1D,94,26,0E76    [994]
2560 DATA 94,40,94,3A,00,A7,C6,41,0F28    [746]
2570 DATA CD,5A,BB,E1,18,C9,3A,00,0E66    [1793]
2580 DATA A7,5F,3E,16,CD,63,CA,6E,09CC    [1498]
2590 DATA 26,00,CD,4D,94,E1,18,B7,0A16    [1736]
2600 DATA 3A,C4,93,CD,69,94,18,0B,083E    [1697]
2610 DATA 21,01,A7,18,EA,3A,C5,93,08CB    [2314]
2620 DATA CD,72,94,E1,18,A1,3A,C6,0CC1    [2190]
2630 DATA 93,A7,21,38,94,28,03,21,0979    [1239]
2640 DATA 3C,94,CD,C7,93,E1,18,8F,0BAB    [2075]
2650 DATA 20,4D,46,F8,4D,46,4D,F8,05A9    [1227]
2660 DATA E1,46,23,7E,23,CD,5A,BB,050B    [2004]
2670 DATA 10,FB,C3,C7,93,11,0A,00,0523    [1057]
2680 DATA CD,56,94,11,01,00,AF,ED,0745    [1804]
2690 DATA 52,38,03,3C,18,F9,19,C6,072D    [1997]
2700 DATA 30,C3,5A,BB,7C,CD,69,94,07F0    [1698]
2710 DATA 7D,F5,0F,0F,0F,0F,CD,72,108F    [1691]
2720 DATA 94,F1,E6,0F,C6,30,FE,3A,1264    [1399]
2730 DATA 38,02,C6,07,C3,5A,BB,CB,1014    [1316]
2740 DATA 27,CB,27,CB,27,CB,27,C9,1314    [1880]
2750 DATA F5,4C,06,00,60,29,29,29,120A    [1657]
2760 DATA 29,54,5D,29,29,19,09,7C,1002    [1545]
2770 DATA E6,07,67,3E,C0,84,67,F1,12E4    [1692]
2780 DATA EB,26,00,6F,29,29,29,01,0C38    [1020]
2790 DATA 00,38,09,01,00,08,EB,3E,0A51    [1583]
2800 DATA 08,F5,1A,E5,2A,04,91,AE,0C6B    [1838]
2810 DATA E1,77,13,09,F1,3D,20,F1,0F25    [977]
2820 DATA C9,21,05,01,CD,75,BB,ED,0E74    [1903]
2830 DATA 5B,17,95,06,10,C5,3E,20,0D86    [1108]
2840 DATA CD,5A,BB,CD,5A,BB,EB,CD,0B2E    [1344]
2850 DATA 64,94,EB,21,0A,95,CD,C7,0A83    [2326]
2860 DATA 93,D5,06,10,1A,CD,69,94,08E2    [1584]
2870 DATA 3E,20,CD,5A,BB,13,10,F4,0B55    [1228]
2880 DATA 21,0F,95,CD,C7,93,D1,06,0BD7    [1934]
2890 DATA 10,1A,CD,19,95,13,10,F9,0993    [1772]
2900 DATA 21,14,95,CD,C7,93,C1,10,0606    [2032]
2910 DATA C4,C9,20,2D,20,20,F8,20,0600    [1856]
2920 DATA 3A,20,20,F8,0D,0A,F8,00,0387    [1348]
2930 DATA 02,CB,BF,FE,20,30,02,3E,04AC    [907]
2940 DATA 2E,C3,5A,BB,E5,6F,7D,CD,0938    [1954]
2950 DATA 69,94,3E,08,CD,5A,BB,CD,0732    [1661]
2960 DATA 5A,BB,CD,14,93,30,0D,CD,06FB    [1359]
2970 DATA 7F,94,06,04,CB,27,CB,15,1057    [2065]
2980 DATA 10,FA,18,E2,7D,E1,C9,CD,1186    [1663]
2990 DATA 64,94,06,04,3E,08,CD,5A,0F8D    [1626]
3000 DATA BB,10,FB,CD,14,93,D0,CD,13F7    [2399]
3010 DATA 7F,94,06,04,CB,27,CB,15,12AF    [2270]
3020 DATA CB,14,10,F8,18,E1,CD,C7,1342    [1564]
3030 DATA 93,21,91,95,CD,C7,93,2A,0E01    [1632]
3040 DATA 17,95,CD,AC,95,22,A6,95,0E15    [1702]
3050 DATA 21,9D,95,CD,C7,93,2A,A6,0CF2    [1671]
3060 DATA 95,CD,AC,95,22,A8,95,ED,107B    [1485]
3070 DATA 5B,A6,95,A7,ED,52,38,EE,10DE    [1104]
3080 DATA C9,20,20,20,20,20,66,69,0CA4    [1064]
3090 DATA 72,73,74,2D,F8,FF,09,09,0E85    [1304]
3100 DATA 6C,61,73,74,2D,F8,00,00,0949    [1628]
3110 DATA 00,00,00,00,E5,CD,47,95,07B6    [1737]
3120 DATA E5,CD,BE,93,D1,A7,ED,52,0B0C    [1375]
3130 DATA EB,D1,C8,EB,18,EE,2A,A8,0D0D    [2262]
3140 DATA 95,ED,5B,A6,95,A7,ED,52,0BF0    [1294]
3150 DATA 44,4D,C9,CD,7F,97,CB,AF,0C3B    [1703]
3160 DATA FE,59,28,04,FE,4E,20,F3,05E2    [1613]
3170 DATA CD,5A,BB,FE,59,C9,CD,D4,084B    [1422]
3180 DATA BC,22,ED,95,79,32,EF,95,0655    [988]
3190 DATA C9,DF,ED,95,C9,00,00,00,082B    [1487]
3200 DATA 21,04,01,11,13,4F,CD,66,0628    [963]
3210 DATA BB,C3,6C,BB,00,96,03,96,06D4    [1806]
3220 DATA 00,06,96,00,15,96,00,00,0EC9    [1166]
```

Listing. "AMSMON PLUS"


```
3230 DATA 00,00,00,00,00,00,00,00,0E10    [917]
3240 DATA 00,00,00,00,00,00,00,00,0C4E    [1010]
3250 DATA 00,00,00,00,00,00,00,00,0CE4    [1184]
3260 DATA 00,00,00,00,CD,8A,BB,CD,12B5    [2174]
3270 DATA 7F,97,F5,CD,8D,BB,F1,FE,1495    [1649]
3280 DATA FC,C0,C3,94,91,A7,F5,06,13F0    [1622]
3290 DATA 00,21,06,96,28,03,21,15,0AB6    [1155]
3300 DATA 96,C5,E5,CD,24,96,E1,C1,0E23    [1335]
3310 DATA FE,0D,28,19,FE,7F,28,29,0C7E    [1390]
3320 DATA 30,EF,FE,20,38,EB,4F,3E,0F49    [1531]
3330 DATA 0F,B8,28,16,79,77,CD,5A,0F06    [1450]
3340 DATA BB,23,04,18,DC,F1,21,00,0E08    [2040]
3350 DATA 96,28,03,21,03,96,78,77,087C    [1044]
3360 DATA A7,C9,3E,07,CD,5A,BB,18,09E9    [2026]
3370 DATA C8,78,A7,28,F5,EB,21,89,098B    [1446]
3380 DATA 96,CD,C7,93,EB,2B,05,18,0BC8    [1344]
3390 DATA B8,08,20,08,F8,CD,DF,C9,0CC9    [2181]
3400 DATA 3A,5F,BE,A7,C8,ED,5B,46,0AF6    [1520]
3410 DATA BE,C3,CD,C9,CD,86,C9,01,0C84    [1695]
3420 DATA 7E,FB,C9,21,00,00,2B,7C,05C6    [1398]
3430 DATA B5,20,FB,3E,08,CD,5C,C9,0570    [1536]
3440 DATA CD,1C,C9,3A,4C,BE,CB,5F,05AC    [2013]
3450 DATA 20,1C,C9,CD,9C,96,3E,04,074A    [1859]
3460 DATA CD,5C,C9,3A,00,A7,CD,5C,072E    [1266]
3470 DATA C9,CD,1C,C9,3A,4C,BE,CB,084A    [944]
3480 DATA 6F,28,03,87,87,C9,21,E5,10B9    [1671]
3490 DATA 96,CD,C7,93,CD,8D,96,CD,134C    [1746]
3500 DATA 7F,97,C3,94,91,20,2D,20,1071    [1498]
3510 DATA 44,72,69,76,65,20,F9,20,12B3    [1568]
3520 DATA 6E,6F,74,20,72,65,61,64,12AD    [1462]
3530 DATA 79,20,21,07,12,F8,21,03,1043    [609]
3540 DATA 97,18,D6,20,2D,20,44,72,0C5E    [2421]
3550 DATA 69,76,65,20,F9,20,69,73,0DAF    [1388]
3560 DATA 20,77,72,69,74,65,20,70,0B5F    [2308]
3570 DATA 72,6F,74,65,63,74,65,64,0C8A    [949]
3580 DATA 20,21,07,12,F8,06,00,21,0D7B    [1100]
3590 DATA 7D,97,CD,24,96,FE,0D,28,0E8E    [2255]
3600 DATA 32,FE,7F,28,1E,FE,30,38,0E43    [1914]
3610 DATA F1,FE,3A,30,ED,4F,3E,02,09AD    [2352]
3620 DATA B8,28,09,79,77,23,04,CD,07E5    [1085]
3630 DATA 5A,BB,18,DE,3E,07,CD,5A,0919    [1387]
3640 DATA BB,18,D7,78,A7,28,F5,EB,0C75    [2190]
3650 DATA 21,89,96,CD,C7,93,EB,2B,0CB7    [1191]
3660 DATA 05,18,C7,78,A7,CA,94,91,0B50    [1342]
3670 DATA 3A,7D,97,D6,30,05,C8,06,057D    [1329]
3680 DATA 09,4F,81,10,FD,4F,3A,7E,0571    [1577]
3690 DATA 97,D6,30,81,C9,00,00,CD,04D4    [1197]
3700 DATA 09,BB,F5,C5,D5,E5,01,14,0871    [1146]
3710 DATA 14,3E,03,CD,A0,97,7E,CB,084C    [1723]
3720 DATA 7F,28,03,01,00,00,CD,38,04A0    [1388]
3730 DATA BC,E1,D1,C1,F1,30,E0,C9,097F    [1245]
3740 DATA FD,E5,FD,21,C5,97,F5,3A,137B    [1376]
3750 DATA 39,00,FE,39,28,04,FD,21,0F58    [1298]
3760 DATA CD,97,F1,3D,28,06,FD,23,10A2    [1137]
3770 DATA FD,23,18,F7,FD,6E,00,FD,13DB    [1818]
3780 DATA 66,01,FD,E1,C9,64,C0,7B,1315    [2066]
3790 DATA AE,E8,B4,90,B2,58,C0,5E,1406    [1114]
3800 DATA AE,32,B6,30,B7,AF,18,02,0CC0    [1576]
3810 DATA 3E,01,21,94,91,22,61,9A,0CB6    [2430]
3820 DATA CB,FF,32,02,A7,CB,BF,F5,0E6A    [1857]
3830 DATA 4F,3A,00,A7,ED,47,79,32,0ED7    [1687]
3840 DATA 00,A7,21,26,98,CD,C7,93,0F8D    [1516]
3850 DATA 3E,C9,32,5A,BB,F1,FD,21,0EE9    [2325]
3860 DATA 00,A7,CD,DF,CD,3E,CF,32,0A5F    [2391]
3870 DATA 5A,BB,3A,02,A7,CB,7F,21,09EB    [1894]
3880 DATA 3C,98,28,41,ED,57,32,00,078F    [1607]
3890 DATA A7,32,02,A7,CD,C7,93,CD,09CC    [2083]
3900 DATA 7F,97,2A,61,9A,E9,1F,02,0B9B    [1369]
3910 DATA 17,12,20,20,20,73,65,6C,08E7    [1757]
3920 DATA 65,63,74,20,64,72,69,76,0A3D    [960]
3930 DATA 65,20,F9,F8,20,2D,20,65,0560    [1069]
3940 DATA 72,72,6F,72,20,64,75,72,0482    [1878]
3950 DATA 69,6E,67,20,6C,6F,67,20,04A4    [880]
3960 DATA 69,6E,20,21,F8,21,87,98,072A    [784]
3970 DATA CD,C7,93,CD,9C,96,CD,70,09E1    [1882]
3980 DATA 98,3A,00,A7,5F,3E,16,CD,0693    [1292]
3990 DATA 63,CA,36,00,2A,61,9A,E9,1117    [1314]
4000 DATA 3E,07,CD,5C,C9,3A,00,A7,10BE    [1414]
4010 DATA CD,5C,C9,CD,A3,96,CB,67,119E    [1421]
```

Listing. "AMSMON PLUS"

```
4020 DATA 20,EE,CB,6F,28,EA,C9,20,139B    [2513]
4030 DATA 20,2D,3E,20,20,72,65,63,1205    [1572]
4040 DATA 61,6C,69,62,72,61,74,65,117A    [1124]
4050 DATA 20,2E,2E,2E,20,20,F8,21,10C9    [1061]
4060 DATA E1,98,CD,66,95,21,EE,98,0F24    [1636]
4070 DATA CD,C7,93,2A,A6,95,CD,AC,0DF5    [1875]
4080 DATA 95,E5,CD,BE,95,D1,19,01,0D9D    [1319]
4090 DATA 00,88,A7,ED,42,30,EC,D5,0FD1    [1376]
4100 DATA CD,BE,95,03,EB,D1,E5,A7,1015    [2477]
4110 DATA ED,52,E1,38,04,ED,B0,18,0F57    [1225]
4120 DATA 0D,ED,4A,E5,EB,ED,4A,EB,0B0C    [1188]
4130 DATA E1,1B,2B,2B,ED,B8,C3,77,0A89    [1274]
4140 DATA 91,1F,01,17,12,1F,05,17,0693    [1443]
4150 DATA 63,6F,70,79,3A,F8,1F,2D,0B33    [1459]
4160 DATA 17,74,6F,2D,F8,21,C6,93,0BC5    [1095]
4170 DATA 3E,40,C3,38,99,21,1D,99,09C3    [942]
4180 DATA CD,66,95,21,2A,99,CD,C7,0662    [853]
4190 DATA 93,AF,CD,24,95,CD,BE,95,07B8    [1091]
4200 DATA 62,6B,13,77,78,B1,28,02,03BC    [1482]
4210 DATA ED,B0,C3,77,91,1F,01,17,054B    [1338]
4220 DATA 12,1F,05,17,66,69,6C,6C,0696    [1776]
4230 DATA 3A,F8,1F,2D,17,77,69,74,062B    [1722]
4240 DATA 68,2D,F8,21,C5,93,3E,01,06AF    [1413]
4250 DATA AE,77,C3,A9,92,21,52,99,117D    [1278]
4260 DATA CD,C7,93,2A,17,95,CD,47,1099    [2057]
4270 DATA 95,CD,A0,93,22,17,95,C3,1144    [1222]
4280 DATA 77,91,1F,01,17,12,1F,05,107F    [1507]
4290 DATA 17,65,6E,74,65,72,20,6E,1283    [1854]
4300 DATA 65,77,20,65,64,69,74,20,11B2    [1525]
4310 DATA 61,64,72,65,73,73,3A,20,0CAA    [1630]
4320 DATA F8,21,63,9A,CD,C7,93,21,0E4E    [584]
4330 DATA 94,91,22,61,9A,AF,CD,35,0CA9    [1075]
4340 DATA 96,CA,22,98,C5,CD,BB,96,109D    [385]
4350 DATA C1,11,00,88,21,06,96,3E,0E79    [932]
4360 DATA C9,32,5A,BB,CD,77,BC,F5,0F7B    [1879]
4370 DATA 3E,CF,32,5A,BB,F1,21,ED,0F57    [1706]
4380 DATA 9A,D2,1C,98,FE,02,21,73,0A1A    [1358]
4390 DATA 9A,C2,1C,98,78,B1,21,8A,0916    [1179]
4400 DATA 9A,CA,1C,98,ED,43,A6,95,09B7    [2045]
4410 DATA 21,9B,9A,CD,C7,93,21,77,0BCF    [1857]
4420 DATA 91,22,61,9A,C5,21,00,00,0978    [1686]
4430 DATA CD,47,95,C1,22,A8,95,A7,0BF4    [2036]
4440 DATA ED,42,30,F0,21,A7,9A,CD,068A    [1170]
4450 DATA C7,93,2A,A8,95,7C,B5,ED,077F    [1488]
4460 DATA 47,2A,A6,95,ED,5B,A8,95,05EF    [1843]
4470 DATA A7,ED,52,E5,CD,47,95,D1,0987    [1551]
4480 DATA 22,AA,95,2B,A7,ED,52,30,07F4    [1712]
4490 DATA E8,23,ED,57,B4,B5,ED,47,07EC    [1129]
4500 DATA 21,B2,9A,CD,C7,93,2A,17,1147    [1158]
4510 DATA 95,CD,AC,95,22,A6,95,ED,12FF    [1037]
4520 DATA 5B,AA,95,19,11,01,88,A7,0F48    [1220]
4530 DATA ED,52,30,EA,21,17,0F,CD,1051    [1685]
4540 DATA 75,BB,2A,A6,95,ED,57,A7,144A    [1225]
4550 DATA 28,2F,ED,4B,A8,95,78,B1,127B    [2079]
4560 DATA 28,06,CD,48,9A,0B,18,F6,119A    [984]
4570 DATA ED,4B,AA,95,78,B1,28,1E,0D74    [1435]
4580 DATA CD,48,9A,77,23,0B,18,F4,0C24    [1647]
4590 DATA E5,C5,CD,80,BC,C1,E1,D8,0F91    [1145]
4600 DATA FE,1A,C8,21,BC,9A,C3,1C,0F98    [1307]
4610 DATA 98,CD,83,BC,30,F5,C3,22,10B2    [1787]
4620 DATA 98,94,91,1F,05,17,12,6C,0D8C    [903]
4630 DATA 6F,61,64,20,66,69,6C,65,0902    [1375]
4640 DATA 3A,20,F8,20,2D,20,69,6E,08BE    [1866]
4650 DATA 76,61,6C,69,64,20,66,69,07E7    [874]
4660 DATA 6C,65,20,74,79,70,20,21,0A67    [1344]
4670 DATA 07,F8,20,2D,20,65,6D,70,0B22    [2391]
4680 DATA 74,79,20,66,69,6C,65,20,0989    [1187]
4690 DATA 21,07,F8,1F,22,17,6F,66,0999    [1312]
4700 DATA 66,73,65,74,3A,20,F8,FF,06B5    [2623]
4710 DATA 07,09,62,79,74,65,73,3A,03CD    [1594]
4720 DATA 20,F8,FF,07,09,6C,6F,61,04F1    [1679]
4730 DATA 64,3A,20,F8,1F,10,17,12,0604    [1623]
4740 DATA 20,2D,20,73,6F,6D,65,74,05C3    [2248]
4750 DATA 68,69,6E,67,20,67,6F,65,06C7    [2060]
4760 DATA 73,20,77,72,6F,6E,67,20,1016    [1234]
4770 DATA 2D,20,63,6F,6D,6D,61,6E,1096    [1170]
4780 DATA 64,20,61,62,6F,72,74,65,1007    [841]
4790 DATA 64,20,21,07,F8,20,2D,20,1187    [1204]
4800 DATA 66,69,6C,65,20,6E,6F,74,129F    [1126]
```

Listing. "AMSMON PLUS"

```
4810 DATA 20,66,6F,75,6E,64,20,21,10CF    [2008]
4820 DATA 07,F8,3E,84,21,35,9C,18,0C8D    [1590]
4830 DATA 11,21,17,02,CD,75,BB,CD,0D5F    [1640]
4840 DATA BB,96,DA,FE,96,3E,85,21,0D29    [2231]
4850 DATA 4E,9C,32,C9,9C,22,A8,95,0D08    [1961]
4860 DATA 21,2C,9B,22,61,9A,3A,00,0E4B    [1434]
4870 DATA A7,C3,E0,97,2A,A8,95,CD,0FE3    [1929]
4880 DATA C7,93,FD,21,CA,9C,21,15,0F0A    [2799]
4890 DATA 9C,22,61,9A,21,68,9C,CD,0993    [1556]
4900 DATA C7,93,AF,CD,25,97,FD,77,0A14    [1518]
4910 DATA 00,CD,21,9C,30,EE,21,73,08EE    [1366]
4920 DATA 9C,CD,C7,93,3A,00,A7,5F,0BA1    [1149]
4930 DATA AF,CD,63,CA,7E,CB,2F,CB,0D2A    [1002]
4940 DATA 2F,FD,77,01,C6,30,CD,5A,0B31    [1542]
4950 DATA BB,21,88,9C,CD,C7,93,CD,0754    [1252]
4960 DATA 25,97,FD,77,02,A7,28,F1,0662    [2047]
4970 DATA 5F,FD,7E,01,BB,38,EA,21,0501    [1987]
4980 DATA 90,9C,CD,C7,93,AF,CD,25,0902    [1830]
4990 DATA 97,FD,77,03,2A,17,95,22,07AC    [1396]
5000 DATA AA,95,3A,00,A7,5F,3E,0F,05C4    [711]
5010 DATA CD,63,CA,FD,7E,02,3D,86,07C6    [1311]
5020 DATA FD,77,02,21,17,17,CD,75,10E3    [1274]
5030 DATA BB,21,C9,9C,CD,DE,95,ED,1218    [1485]
5040 DATA 5B,AA,95,21,00,86,A7,ED,10A3    [1274]
5050 DATA 52,21,D1,A4,DA,1C,98,3A,12EE    [1591]
5060 DATA 00,A7,5F,FD,56,00,FD,4E,1214    [1948]
5070 DATA 02,2A,AA,95,CD,E9,95,21,12D1    [1996]
5080 DATA 9B,9C,D2,1C,98,2A,AA,95,0DAA    [931]
5090 DATA 24,24,22,AA,95,FD,34,02,0CE0    [1533]
5100 DATA 79,E6,0F,FD,BE,01,20,21,0CAF    [883]
5110 DATA 3E,F0,FD,A6,02,3C,FD,77,1049    [2211]
5120 DATA 02,FD,34,00,FD,7E,00,CD,0F4F    [1105]
5130 DATA 21,9C,21,AF,9C,D2,1C,98,0E3B    [1458]
5140 DATA 3E,F0,FD,A6,02,3C,FD,77,0A8B    [2323]
5150 DATA 02,FD,35,03,C2,AF,9B,FD,0AC8    [1485]
5160 DATA 7E,02,32,C4,93,21,77,91,0800    [1505]
5170 DATA 22,61,9A,3A,00,A7,C3,E0,090B    [1799]
5180 DATA 97,FE,54,D0,3A,00,A7,3C,0C2E    [1073]
5190 DATA 5F,3A,79,BE,A3,37,C0,FD,0B77    [1155]
5200 DATA 7E,00,FE,2A,C9,1F,02,17,09D3    [1136]
5210 DATA 12,6D,75,6C,74,69,20,73,04E4    [1938]
5220 DATA 65,63,74,6F,72,20,72,65,0466    [1550]
5230 DATA 61,64,20,2D,20,F8,1F,02,0437    [1884]
5240 DATA 17,12,6D,75,6C,74,69,20,065C    [1531]
5250 DATA 73,65,63,74,6F,72,20,77,079F    [1146]
5260 DATA 72,69,74,65,20,2D,20,F8,06C1    [1458]
5270 DATA 1F,17,17,74,72,61,63,6B,0FF8    [1717]
5280 DATA 3A,20,F8,1F,28,17,6C,6F,1045    [1246]
5290 DATA 67,2E,20,73,65,63,74,6F,0F3B    [1765]
5300 DATA 72,20,28,31,2E,2E,2E,F8,11C5    [2044]
5310 DATA 1F,3A,17,29,3A,20,12,F8,11E9    [1475]
5320 DATA 1F,46,17,63,6F,75,6E,74,10D3    [1721]
5330 DATA 3A,20,F8,20,65,78,65,63,11E3    [1713]
5340 DATA 75,74,69,6F,6E,20,65,72,0D52    [1121]
5350 DATA 72,6F,72,20,21,12,F8,20,0BA6    [1019]
5360 DATA 69,6E,76,61,6C,69,64,20,0C0B    [1261]
5370 DATA 74,72,61,63,6B,20,72,65,0E92    [1640]
5380 DATA 61,63,68,65,64,20,21,12,0CF2    [1839]
5390 DATA F8,00,00,00,00,00,21,F0,0D4B    [865]
5400 DATA 9C,CD,C7,93,21,94,91,22,09DF    [1393]
5410 DATA 61,9A,CD,02,9D,21,00,88,095E    [1201]
5420 DATA 43,7E,CD,69,94,3E,20,CD,0932    [1585]
5430 DATA 5A,BB,23,10,F4,C3,1F,98,0BB2    [1772]
5440 DATA 1F,05,17,12,74,72,61,63,0A13    [908]
5450 DATA 6B,20,69,6E,66,6F,3A,20,0957    [1765]
5460 DATA 20,F8,CD,48,9D,3A,4D,BE,0651    [1785]
5470 DATA CB,47,21,2F,9D,C2,1C,98,0651    [1959]
5480 DATA 3A,51,BE,FD,21,47,9D,FD,055C    [1436]
5490 DATA 77,00,DD,21,00,88,1E,00,03BB    [1140]
5500 DATA CD,48,9D,DD,77,00,DD,23,08A8    [325]
5510 DATA 1C,FD,BE,00,20,F2,C9,20,0710    [1863]
5520 DATA 2D,20,6E,6F,20,73,65,63,05ED    [1543]
5530 DATA 74,6F,72,20,49,44,20,66,0FE0    [1633]
5540 DATA 6F,75,6E,64,20,21,F8,00,0F79    [1324]
5550 DATA D5,CD,9C,96,3A,C6,93,F6,127F    [1502]
5560 DATA 0A,CD,5C,C9,3A,00,A7,5F,125C    [2338]
5570 DATA 3A,C5,93,CB,27,CB,27,B3,13D5    [2053]
5580 DATA CD,5C,C9,CD,1C,C9,D1,3A,138B    [2070]
5590 DATA 4C,BE,CB,5F,3A,51,BE,C8,0E19    [1254]
5600 DATA C3,D6,96,21,D7,9D,CD,66,0EE9    [1930]
```

Listing. "AMSMON PLUS"

```
5610 DATA 95,2A,5B,BB,22,AA,95,ED,0CCB    [1953]
5620 DATA 5B,A6,95,2A,A8,95,A7,ED,1039    [1948]
5630 DATA 52,DA,94,91,21,EC,9D,CD,110E    [1405]
5640 DATA C7,93,EB,CD,64,94,EB,3E,0F8F    [1253]
5650 DATA C3,21,BF,9D,CD,B8,9D,06,0F6E    [1580]
5660 DATA 01,CD,CD,94,3E,CF,2A,AA,0A90    [2262]
5670 DATA 95,CD,B8,9D,CD,09,BB,30,099C    [1247]
5680 DATA D2,FE,FC,20,CE,C3,94,91,0AEE    [1731]
5690 DATA 32,5A,BB,22,5B,BB,C9,CD,0BE3    [1216]
5700 DATA 2B,BD,D8,CD,09,BB,30,F7,0B5A    [1665]
5710 DATA FE,FC,20,F3,3E,CF,2A,AA,0C6C    [1665]
5720 DATA 95,CD,B8,9D,CA,94,91,1F,06B9    [584]
5730 DATA 05,17,12,64,75,6D,70,20,0498    [1879]
5740 DATA 74,6F,20,70,72,69,6E,74,04EA    [1815]
5750 DATA 65,72,3A,F8,1F,36,17,3C,06EB    [1708]
5760 DATA 45,53,43,3E,20,74,6F,20,068E    [1411]
5770 DATA 61,62,6F,72,74,20,2D,20,0595    [382]
5780 DATA 26,F8,21,32,9E,CD,C7,93,11AA    [1906]
5790 DATA CD,7F,97,FE,FC,CA,94,91,13E8    [829]
5800 DATA FE,0D,20,F4,3E,01,CD,0E,0F8D    [1317]
5810 DATA BC,CD,02,BC,CD,51,BB,AF,11B3    [1264]
5820 DATA 32,78,BE,3E,0A,32,66,BE,12E0    [1620]
5830 DATA 3E,01,CD,A0,97,0E,00,C3,119C    [1091]
5840 DATA 1B,00,1F,01,17,12,1F,16,0F41    [2296]
5850 DATA 17,70,72,65,73,73,20,3C,0C3A    [1696]
5860 DATA 45,4E,54,45,52,3E,20,74,0B18    [1349]
5870 DATA 6F,20,65,78,69,74,20,2D,0BF2    [2158]
5880 DATA 20,3C,45,53,43,3E,20,74,0D6B    [1704]
5890 DATA 6F,20,61,62,6F,72,74,07,0EA6    [1714]
5900 DATA F8,21,B5,9E,CD,66,95,21,0F75    [1397]
5910 DATA 94,91,22,61,9A,21,C5,9E,09DE    [905]
5920 DATA CD,C7,93,AF,CD,35,96,CA,0B6E    [2147]
5930 DATA 94,91,C5,21,17,10,CD,75,085C    [849]
5940 DATA BB,CD,BB,96,DA,FE,96,21,0D4E    [1082]
5950 DATA 06,96,C1,11,00,88,3E,C9,0B6B    [713]
5960 DATA 32,5A,BB,CD,8C,BC,F5,3E,0B4B    [2851]
5970 DATA CF,32,5A,BB,F1,D2,53,9A,0BFE    [1574]
5980 DATA CD,BE,95,23,EB,3E,02,CD,06E3    [1627]
5990 DATA 98,BC,30,03,CD,8F,BC,D2,05E7    [1636]
6000 DATA 53,9A,C3,94,91,1F,05,17,0498    [1060]
6010 DATA 12,73,61,76,65,20,66,69,06AC    [2035]
6020 DATA 6C,65,3A,20,F8,1F,32,17,05B7    [840]
6030 DATA 66,69,6C,65,6E,61,6D,65,0703    [1379]
6040 DATA 3A,20,F8,21,04,9F,CD,C7,10EA    [2035]
6050 DATA 93,CD,25,97,FE,54,30,F3,1263    [1304]
6060 DATA F5,CD,9C,96,3E,0F,CD,5C,116E    [1370]
6070 DATA C9,3A,00,A7,CD,5C,C9,5F,1375    [1478]
6080 DATA 3E,16,CD,63,CA,F1,77,CD,1419    [2144]
6090 DATA 5C,C9,CD,A3,96,CB,6F,28,12E3    [2020]
6100 DATA F9,C3,94,91,1F,05,17,12,0CF0    [1856]
6110 DATA 73,65,65,6B,20,74,6F,20,0D11    [1773]
6120 DATA 74,72,61,63,6B,3A,20,F8,0BF9    [1965]
6130 DATA 21,47,9F,CD,C7,93,AF,CD,0DCE    [1855]
6140 DATA 35,96,F5,CD,BB,96,CD,F0,11A3    [1717]
6150 DATA 95,F1,20,08,11,00,88,CD,0DD2    [996]
6160 DATA 9B,BC,18,0D,3E,01,FD,21,0DB9    [850]
6170 DATA 00,A7,DD,21,FC,95,CD,2A,0A0B    [1158]
6180 DATA C0,CD,7F,97,C3,77,91,1F,0997    [1593]
6190 DATA 05,17,12,65,6E,74,65,72,07DC    [1666]
6200 DATA 20,64,69,72,20,6D,61,73,0A50    [1036]
6210 DATA 6B,3A,20,F8,21,96,9F,CD,0C0C    [1620]
6220 DATA C7,93,AF,CD,35,96,28,2B,0B60    [1957]
6230 DATA 21,A7,9F,3E,01,F5,E5,CD,06A7    [1898]
6240 DATA BB,96,DA,FE,96,21,AA,9F,07A5    [2332]
6250 DATA CD,C7,93,E1,CD,DE,95,F1,0759    [1692]
6260 DATA F5,DD,21,FC,95,CD,E9,95,09E7    [1015]
6270 DATA 47,F1,C6,A7,B8,CA,93,9F,0A03    [1458]
6280 DATA CD,7F,97,C3,94,91,1F,05,06E5    [1361]
6290 DATA 17,12,65,72,61,73,65,20,05E9    [1476]
6300 DATA 66,69,6C,65,3A,20,F8,45,112D    [1646]
6310 DATA 52,C1,1F,01,17,12,0B,F8,0F05    [1237]
6320 DATA 21,D6,9F,CD,C7,93,AF,CD,11EF    [2306]
6330 DATA 35,96,28,0B,21,E8,9F,CD,129D    [1683]
6340 DATA C7,93,3E,01,CD,35,96,28,11C1    [1753]
6350 DATA CA,3E,02,21,17,26,CD,75,11A8    [1267]
6360 DATA BB,21,FA,9F,18,97,1F,05,0CC0    [2142]
6370 DATA 17,12,72,65,6E,61,6D,65,0CB7    [1399]
6380 DATA 20,66,69,6C,65,3A,20,F8,0C5E    [1448]
6390 DATA 1F,26,17,6E,65,77,20,66,0DE2    [1514]
6400 DATA 69,6C,65,6E,61,6D,65,3A,0EEB    [1213]
```

Listing. "AMSMON PLUS"


```
6410 DATA 20,F8,52,45,CE,21,10,A0,0DE0    [1476]
6420 DATA CD,C7,93,CD,25,97,FE,10,0AB8    [1749]
6430 DATA 30,F3,32,01,A7,C3,94,91,0A79    [1950]
6440 DATA 1F,05,17,12,6E,65,77,20,0695    [593]
6450 DATA 75,73,65,72,3A,20,F8,21,0892    [1661]
6460 DATA 2F,A0,CD,C7,93,CD,7F,97,0D23    [2259]
6470 DATA FE,FC,20,F9,C3,77,91,1F,0BFB    [1550]
6480 DATA 03,05,12,1F,03,06,12,20,079C    [2104]
6490 DATA 43,74,72,6C,2D,41,3A,20,0479    [1893]
6500 DATA 20,53,65,6C,65,63,74,20,03E0    [1704]
6510 DATA 64,72,69,76,65,20,41,20,0483    [1092]
6520 DATA 20,26,20,20,48,4F,4D,45,0591    [1708]
6530 DATA 1F,2C,06,43,74,72,6C,2D,0697    [1337]
6540 DATA 42,3A,20,20,53,65,6C,65,05ED    [1778]
6550 DATA 63,74,20,64,72,69,76,65,10AF    [1780]
6560 DATA 20,42,20,20,26,20,20,48,0F06    [1183]
6570 DATA 4F,4D,45,1F,03,07,12,20,0DA4    [1709]
6580 DATA 43,74,72,6C,2D,43,3A,20,11C1    [1745]
6590 DATA 20,43,6F,70,79,20,62,75,12A6    [1670]
6600 DATA 66,66,65,72,20,6D,65,6D,112E    [1384]
6610 DATA 6F,72,79,1F,2C,07,43,74,1131    [2037]
6620 DATA 72,6C,2D,44,3A,20,20,43,0C3A    [1203]
6630 DATA 68,61,6E,67,65,20,64,65,0BD4    [1212]
6640 DATA 6E,73,69,74,79,20,6D,6F,0C37    [1364]
6650 DATA 64,65,1F,03,08,12,20,43,0CE6    [1170]
6660 DATA 74,72,6C,2D,45,3A,20,20,0CEE    [1343]
6670 DATA 4E,65,77,20,65,64,69,74,0E2C    [1397]
6680 DATA 20,61,64,64,72,65,73,73,08C0    [1623]
6690 DATA 1F,2C,08,43,74,72,6C,2D,0869    [1598]
6700 DATA 46,3A,20,20,46,69,6C,6C,07D1    [2338]
6710 DATA 20,62,75,66,66,65,72,20,0AC0    [1099]
6720 DATA 6D,65,6D,6F,72,79,1F,03,0ADD    [1906]
6730 DATA 09,12,20,43,74,72,6C,2D,08C9    [1625]
6740 DATA 47,3A,20,20,4D,75,6C,74,048D    [2079]
6750 DATA 69,20,73,65,63,74,6F,72,05EB    [1404]
6760 DATA 20,72,65,61,64,1F,2C,09,0314    [1586]
6770 DATA 43,74,72,6C,2D,48,3A,20,0408    [1714]
6780 DATA 20,43,68,61,6E,67,65,20,071C    [2287]
6790 DATA 68,65,61,64,20,73,65,6C,0632    [1748]
6800 DATA 65,63,74,1F,03,0A,12,20,0502    [981]
6810 DATA 43,74,72,6C,2D,49,3A,20,0FC1    [1792]
6820 DATA 20,43,68,61,6E,67,65,20,0F14    [2130]
6830 DATA 65,64,69,74,20,6D,6F,64,1012    [1727]
6840 DATA 65,20,20,28,54,41,42,29,10E9    [1190]
6850 DATA 1F,2C,0A,43,74,72,6C,2D,11CB    [1833]
6860 DATA 4A,3A,20,20,50,72,69,6E,1145    [1962]
6870 DATA 74,20,52,65,73,75,6C,74,0CD9    [1522]
6880 DATA 20,50,68,61,73,65,1F,03,0C2F    [981]
6890 DATA 0B,12,20,43,74,72,6C,2D,0AAB    [1199]
6900 DATA 4B,3A,20,20,57,72,69,74,0E0F    [1810]
6910 DATA 65,20,64,65,6C,65,74,65,0F34    [1140]
6920 DATA 64,20,73,65,63,74,6F,72,0D84    [1590]
6930 DATA 1F,2C,0B,43,74,72,6C,2D,0D22    [1720]
6940 DATA 4C,3A,20,20,4C,6F,61,64,08BE    [1422]
6950 DATA 20,66,69,6C,65,1F,03,0C,071E    [1548]
6960 DATA 12,20,43,74,72,6C,2D,4D,0785    [1701]
6970 DATA 3A,20,20,3C,45,4E,54,45,0994    [1382]
6980 DATA 52,3E,1F,2C,0C,43,74,72,0910    [1482]
6990 DATA 6C,2D,4E,3A,20,20,4D,75,09A3    [2479]
7000 DATA 6C,74,69,20,73,65,63,74,0506    [1610]
7010 DATA 6F,72,20,77,72,69,74,65,05C8    [1873]
7020 DATA 1F,03,0D,12,20,43,74,72,0346    [1990]
7030 DATA 6C,2D,4F,3A,20,20,54,72,0666    [1822]
7040 DATA 61,63,6B,20,69,6E,66,6F,0755    [1083]
7050 DATA 1F,2C,0D,43,74,72,6C,2D,0514    [1444]
7060 DATA 50,3A,20,20,44,75,6D,70,0FD0    [2072]
7070 DATA 20,74,6F,20,70,72,69,6E,10E0    [1478]
7080 DATA 74,65,72,1F,03,0E,12,20,0DF5    [1926]
7090 DATA 43,74,72,6C,2D,51,3A,20,0F51    [2427]
7100 DATA 20,51,75,69,74,20,41,4D,123B    [1459]
7110 DATA 53,4D,4F,4E,20,50,4C,55,10D2    [1098]
7120 DATA 53,1F,2C,0E,43,74,72,6C,10F3    [1552]
7130 DATA 2D,52,3A,20,20,52,65,61,0BA9    [1570]
7140 DATA 64,20,73,65,63,74,6F,72,0BE0    [1456]
7150 DATA 1F,03,0F,12,20,43,74,72,0AFE    [1896]
7160 DATA 6C,2D,53,3A,20,20,53,61,0D72    [2217]
7170 DATA 76,65,20,66,69,6C,65,1F,0EA6    [2097]
7180 DATA 2C,0F,43,74,72,6C,2D,54,0D73    [1806]
7190 DATA 3A,20,20,53,65,74,20,74,0854    [1326]
7200 DATA 72,61,63,6B,20,61,6E,64,0932    [1800]
```

Listing. "AMSMON PLUS"

```
7210 DATA 20,73,65,65,6B,1F,03,10,06D8    [1531]
7220 DATA 12,20,43,74,72,6C,2D,55,0A21    [1703]
7230 DATA 3A,20,20,53,65,74,20,75,0AAB    [1148]
7240 DATA 73,65,72,20,20,28,30,2E,08C2    [1497]
7250 DATA 2E,2E,31,35,29,1F,2C,10,088A    [2312]
7260 DATA 43,74,72,6C,2D,56,3A,20,0524    [1710]
7270 DATA 20,56,65,72,69,66,79,20,041D    [1192]
7280 DATA 73,65,63,74,6F,72,1F,03,0436    [847]
7290 DATA 11,12,20,43,74,72,6C,2D,0601    [1069]
7300 DATA 57,3A,20,20,57,72,69,74,059B    [1848]
7310 DATA 65,20,73,65,63,74,6F,72,06D7    [1316]
7320 DATA 1F,2C,11,43,74,72,6C,2D,0F6A    [1676]
7330 DATA 58,3A,20,20,44,69,72,65,1030    [1470]
7340 DATA 63,74,6F,72,79,1F,03,12,0F67    [1235]
7350 DATA 12,20,43,74,72,6C,2D,59,11BB    [1637]
7360 DATA 3A,20,20,45,72,61,73,65,11EE    [1406]
7370 DATA 20,66,69,6C,65,1F,2C,12,1067    [1280]
7380 DATA 43,74,72,6C,2D,5A,3A,20,0C38    [1940]
7390 DATA 20,52,65,6E,61,6D,65,20,0CE4    [2063]
7400 DATA 66,69,6C,65,1F,03,13,12,0A69    [1483]
7410 DATA 1F,03,14,12,1F,27,17,6C,0A3B    [1413]
7420 DATA 65,61,76,65,07,F8,3E,06,0F00    [2025]
7430 DATA 21,84,A4,18,27,21,17,05,0C81    [1398]
7440 DATA CD,75,BB,3E,12,CD,5A,BB,0F1D    [1812]
7450 DATA CD,69,A4,CD,7F,97,C3,94,0ADC    [2293]
7460 DATA 91,3E,05,21,96,A4,18,0C,0759    [1251]
7470 DATA 3E,09,21,A9,A4,18,05,3E,07AE    [1829]
7480 DATA 11,21,BD,A4,32,8C,A5,CD,0B4F    [1213]
7490 DATA C7,93,21,94,91,22,61,9A,0BF9    [1687]
7500 DATA ED,5B,17,95,21,00,80,A7,0AA0    [974]
7510 DATA ED,52,21,D1,A4,DA,1C,98,06B7    [1174]
7520 DATA 3A,C4,93,CD,24,95,32,C4,0693    [1748]
7530 DATA 93,CD,02,9D,43,21,00,88,0415    [1357]
7540 DATA 3A,C4,93,BE,28,09,23,10,06CD    [1693]
7550 DATA FA,21,F3,A4,C3,1C,98,DD,09B2    [2049]
7560 DATA 21,C4,93,CD,48,9D,DD,BE,07BD    [1691]
7570 DATA 00,20,F8,3A,8C,A5,5F,E6,075A    [1956]
7580 DATA 03,FE,01,F5,CD,BB,96,30,123B    [2530]
7590 DATA 05,F1,F5,CA,FE,96,CD,9C,126E    [843]
7600 DATA 96,3A,8C,A5,5F,3A,C6,93,10B1    [1051]
7610 DATA B3,CD,5C,C9,3A,00,A7,5F,1327    [1652]
7620 DATA 3A,C5,93,CB,27,CB,27,B3,128B    [2292]
7630 DATA CD,5C,C9,21,4F,BE,7E,23,12CF    [1069]
7640 DATA CD,5C,C9,7E,23,CD,5C,C9,0DFB    [1386]
7650 DATA 7E,23,CD,5C,C9,7E,CD,5C,0E5A    [1375]
7660 DATA C9,2B,7E,CD,5C,C9,3E,11,0D01    [2104]
7670 DATA CD,59,C9,3E,FF,F3,CD,5C,1114    [1408]
7680 DATA C9,F1,21,FF,A3,E5,2A,17,1087    [1208]
7690 DATA 95,CA,F5,C6,C3,E5,C6,CD,10E1    [1770]
7700 DATA 1C,C9,CD,8D,96,11,4C,BE,09E2    [1835]
7710 DATA 1A,CB,5F,C2,D6,96,13,1A,0A37    [2720]
7720 DATA CB,6F,21,0B,A5,C2,77,91,089B    [883]
7730 DATA 13,1A,CB,77,C4,51,A5,3A,08CD    [2118]
7740 DATA 8C,A5,FE,11,21,26,A5,C2,0C48    [1976]
7750 DATA 35,A4,21,1B,A5,1A,CB,5F,0A12    [975]
7760 DATA 28,03,21,27,A5,E5,21,4C,098E    [1903]
7770 DATA BE,11,E5,96,CB,5E,20,0B,05AE    [1551]
7780 DATA 23,11,57,A4,CB,6E,20,03,03D9    [1464]
7790 DATA E3,EB,AF,EB,F5,CD,C7,93,0864    [1990]
7800 DATA F1,C4,7F,97,C3,77,91,20,0892    [1876]
7810 DATA 2D,20,64,61,74,61,20,65,06D4    [1621]
7820 DATA 72,72,6F,72,20,21,07,12,05CD    [1158]
7830 DATA F8,21,3C,A5,CD,C7,93,21,11D8    [1314]
7840 DATA 4B,BE,46,78,A7,C8,23,7E,1193    [2410]
7850 DATA CD,69,94,3E,20,CD,5A,BB,1064    [1891]
7860 DATA 23,10,F4,C9,1F,02,17,12,118C    [1263]
7870 DATA 72,65,61,64,20,73,65,63,12F5    [1665]
7880 DATA 74,6F,72,3A,20,F8,1F,02,1104    [1929]
7890 DATA 17,12,77,72,69,74,65,20,1128    [1101]
7900 DATA 73,65,63,74,6F,72,3A,20,0D1E    [1348]
7910 DATA F8,1F,02,17,12,77,72,69,0B74    [1450]
7920 DATA 74,65,20,64,65,6C,65,74,0C0F    [1857]
7930 DATA 65,64,3A,20,F8,1F,02,17,0DD1    [1844]
7940 DATA 12,76,65,72,69,66,79,20,0D71    [1623]
7950 DATA 73,65,63,74,6F,72,3A,20,0E2C    [1100]
7960 DATA F8,20,69,6E,76,61,6C,69,096D    [979]
7970 DATA 64,20,44,4D,41,20,61,64,088F    [1345]
7980 DATA 72,65,73,73,20,28,74,6F,086C    [1184]
7990 DATA 6F,20,68,69,67,68,29,20,0A6C    [1941]
8000 DATA 21,12,F8,09,09,20,2D,20,09BC    [1759]
```

Listing. "AMSMON PLUS"

```
8010 DATA 73,65,63,74,6F,72,20,6E,09F0    [1449]
8020 DATA 6F,74,20,66,6F,75,6E,64,0545    [2015]
8030 DATA 20,21,F8,20,2D,20,64,61,0539    [1341]
8040 DATA 74,61,20,65,72,72,6F,72,0431    [1650]
8050 DATA 20,21,F8,20,2D,20,66,61,040F    [1344]
8060 DATA 69,6C,65,64,20,21,F8,20,0787    [1148]
8070 DATA 2D,20,73,65,63,74,6F,72,0627    [1683]
8080 DATA 20,76,65,72,69,66,69,65,0666    [1623]
8090 DATA 64,20,21,F8,09,09,20,20,0F4D    [2040]
8100 DATA 20,20,72,65,73,75,6C,74,0F77    [1472]
8110 DATA 20,70,68,61,73,65,3A,20,0FB9    [1425]
8120 DATA F8,CD,78,BB,E5,21,5F,A5,1422    [2452]
8130 DATA CD,C7,93,E1,C3,75,BB,1F,14D2    [1589]
8140 DATA 02,19,18,12,1F,18,19,2A,0F9B    [1743]
8150 DATA 2A,2A,20,20,64,65,6C,65,0C14    [1529]
8160 DATA 74,65,64,20,64,61,74,61,0CEB    [1491]
8170 DATA 20,61,64,72,65,73,73,20,0B70    [1785]
8180 DATA 6D,61,72,6B,20,20,2A,2A,0DD7    [1469]
8190 DATA 2A,18,07,F8,00,00,00,00,0D73    [1939]
```

Listing. "AMSMON PLUS"

```
1000 '================================    [2011]
1010 '**                              **    [104]
1020 '**  BASIC-765     Version 1.3   **    [1813]
1030 '**  ------------------------    **    [819]
1040 '**                              **    [104]
1050 '**       August 1987 M.Siebke   **    [1153]
1060 '**                              **    [104]
1070 '================================    [2011]
1080 :                                     [174]
1090 MODE 2: MEMORY &9DFF: zz=0            [2070]
1100 PRINT "Bitte warten...": PRINT        [3056]
1110 :                                     [174]
1120 FOR adr=&9E00 TO &A62F STEP 8         [1125]
1130   :                                   [174]
1140   PRINT "*";: s=0                     [709]
1150   FOR i=0 TO 7                        [486]
1160     READ b$: b=VAL("&"+b$)            [1052]
1170     s=s+b XOR i: POKE adr+i,b         [1100]
1180   NEXT                                [350]
1190   READ c$: c=VAL("&"+c$)              [1443]
1200   s=s+16*((zz XOR &55) MOD 256)       [1376]
1210   s=s+    (zz XOR &AA) MOD 256        [1985]
1220   zz=zz+10: IF c=s GOTO 1260          [1889]
1230   PRINT " DATA Fehler in Zeile";      [1766]
1240   PRINT zz+1290: STOP                 [625]
1250   :                                   [174]
1260 NEXT                                  [350]
1270 :                                     [174]
1280 SAVE "BASIC765",b,&9E00,&82F,&9E09    [1685]
1290 :                                     [174]
1300 DATA 21,E5,9E,01,8F,9E,C3,D1,0A66    [1773]
1310 DATA BC,21,11,9E,0E,FC,C3,16,09FF    [1553]
1320 DATA BD,3E,01,CD,0E,BC,3A,3E,07E1    [1516]
1330 DATA F6,21,5C,06,FE,B0,28,0A,08B5    [1521]
1340 DATA 21,57,06,FE,2A,28,03,21,0A40    [1291]
1350 DATA 77,06,CD,4E,9E,11,40,00,0991    [1890]
1360 DATA 21,FF,AB,CD,CB,BC,21,4F,0BBD    [2032]
1370 DATA 9E,CD,BC,A5,CD,00,9E,31,0680    [870]
1380 DATA 00,C0,CD,00,B9,21,7F,9E,04C2    [2154]
1390 DATA 11,40,00,C3,0C,C0,E9,20,04CD    [1557]
1400 DATA 42,41,53,49,43,2D,37,36,05D8    [1634]
1410 DATA 35,20,56,31,2E,33,20,2D,0602    [1098]
1420 DATA 20,A4,20,31,39,38,37,20,0587    [1570]
1430 DATA 4D,2E,53,69,65,62,6B,65,1062    [700]
1440 DATA 0A,0D,0A,00,00,00,00,00,0DD7    [902]
1450 DATA 00,00,00,00,00,00,00,00,0C6C    [1000]
1460 DATA 86,C9,07,CD,C9,07,DF,C9,13FD    [1235]
1470 DATA 07,5C,C9,07,1C,C9,07,E9,12F0    [1051]
1480 DATA 9E,C3,73,9F,C3,04,A0,C3,12CD    [1415]
1490 DATA 0A,A0,C3,0F,A0,C3,14,A0,1253    [1853]
1500 DATA C3,C3,A0,C3,2D,A1,C3,63,0F19    [1688]
1510 DATA A1,C3,BE,A1,C3,00,A2,C3,0DCB    [1722]
```

Listing. "BASIC 765"

```
1520 DATA DE,A2,C3,29,A3,C3,BA,A3,0E27    [1895]
1530 DATA C3,10,A4,C3,14,A4,C3,19,0F4C    [1355]
1540 DATA A4,C3,AB,A4,C3,C0,A4,C3,1042    [985]
1550 DATA EC,A4,C3,FE,A4,C3,10,A5,10B1    [1587]
1560 DATA C3,39,A5,C3,90,A3,C3,96,0AAC    [1256]
1570 DATA A3,C3,62,A5,C3,66,A5,C3,0B54    [857]
1580 DATA 6B,A5,C3,84,A5,00,00,00,0880    [1819]
1590 DATA 00,52,45,41,C4,57,52,49,0A7C    [999]
1600 DATA 54,C5,44,52,45,41,C4,44,0B4F    [1867]
1610 DATA 57,52,49,54,C5,53,49,4E,09BD    [1435]
1620 DATA 46,CF,53,45,43,54,4F,52,0521    [1317]
1630 DATA D3,56,45,52,49,46,D9,48,0640    [949]
1640 DATA 4F,4D,C5,53,45,45,CB,53,045A    [1988]
1650 DATA 54,41,54,55,D3,53,54,45,048D    [2319]
1660 DATA D0,54,46,4F,52,4D,41,D4,07F3    [1816]
1670 DATA 53,49,5A,C5,4D,46,CD,4D,06B6    [1344]
1680 DATA C6,44,55,4D,D0,41,53,43,06B1    [1314]
1690 DATA 49,C9,53,45,54,44,4D,C1,10A4    [1205]
1700 DATA 47,45,54,44,4D,C1,45,52,0F61    [1008]
1710 DATA 52,4F,D2,48,45,41,C4,55,1070    [1403]
1720 DATA 4E,49,D4,46,47,41,D0,47,126C    [1964]
1730 DATA 41,50,B3,4D,53,47,4F,CE,1302    [1169]
1740 DATA 4D,53,47,4F,46,C6,52,45,11CD    [1475]
1750 DATA 53,55,4C,D4,52,53,58,37,0CDE    [1552]
1760 DATA 36,B5,00,F5,3E,06,32,23,0C5F    [1574]
1770 DATA A6,F1,3D,28,04,3D,C2,AE,0C53    [584]
1780 DATA A5,CD,FE,A5,3A,23,A6,5F,1011    [554]
1790 DATA E6,03,FE,01,F5,3A,17,A6,100C    [788]
1800 DATA B3,CD,FA,A5,CD,E7,A5,3A,1018    [1533]
1810 DATA 18,A6,CD,FA,A5,3A,15,A6,0F2F    [1343]
1820 DATA CD,FA,A5,DD,7E,00,CD,FA,0BF6    [2423]
1830 DATA A5,3A,19,A6,CD,FA,A5,DD,0A0D    [1431]
1840 DATA 7E,00,CD,FA,A5,3A,1C,A6,0926    [1652]
1850 DATA CD,FA,A5,3A,1A,A6,2A,1E,0B68    [2007]
1860 DATA A6,CD,FA,A5,F1,F3,28,19,0C25    [1098]
1870 DATA 18,06,0C,ED,78,0D,77,23,09A8    [837]
1880 DATA ED,78,F2,D0,9F,E6,20,20,06EA    [1473]
1890 DATA F1,18,0F,0C,7E,ED,79,0D,05B3    [1407]
1900 DATA 23,ED,78,F2,E1,9F,E6,20,06C6    [1493]
1910 DATA 20,F1,CD,F6,A5,CD,05,A6,0921    [1889]
1920 DATA 3A,23,A6,FE,11,CA,33,A1,07FA    [1772]
1930 DATA CD,6B,A5,3A,4C,BE,CB,5F,075D    [1410]
1940 DATA C8,C3,88,A1,F5,3E,05,C3,122F    [2037]
1950 DATA 76,9F,F5,3E,0C,18,F8,F5,1261    [1082]
1960 DATA 3E,09,18,F3,3D,C2,AE,A5,0FF2    [1405]
1970 DATA CD,0D,A1,3A,4D,BE,CB,47,10C8    [1955]
1980 DATA 21,68,A0,C2,B1,A5,3A,51,13A8    [1167]
1990 DATA BE,FD,21,20,A6,FD,77,00,12A2    [1231]
2000 DATA CD,0D,A1,DD,BE,00,28,0E,11E6    [1882]
2010 DATA FD,BE,00,20,F3,CD,05,A6,0DCC    [1437]
2020 DATA 21,7D,A0,C3,B1,A5,3A,22,0C73    [1721]
2030 DATA A6,A7,C8,11,4F,BE,06,04,0C93    [1223]
2040 DATA 21,90,A0,CD,BC,A5,1A,13,0F0A    [2602]
2050 DATA CD,7D,A4,3E,20,CD,5A,BB,1020    [1449]
2060 DATA 10,F1,CD,B1,A5,C3,05,A6,0FAA    [1834]
2070 DATA 6E,6F,20,73,65,63,74,6F,092F    [1147]
2080 DATA 72,20,49,44,20,66,6F,75,08CD    [1750]
2090 DATA 6E,64,20,21,00,73,65,63,073C    [2268]
2100 DATA 74,6F,72,20,6E,6F,74,20,0AC0    [1566]
2110 DATA 66,6F,75,6E,64,20,21,00,0AD1    [1311]
2120 DATA 54,72,61,63,6B,20,20,3A,092D    [1438]
2130 DATA 20,26,00,0D,0A,48,65,61,08A9    [982]
2140 DATA 64,20,20,20,3A,20,26,00,03F6    [1686]
2150 DATA 0D,0A,53,65,63,74,6F,72,03E9    [987]
2160 DATA 20,3A,20,26,00,0D,0A,53,027E    [1345]
2170 DATA 69,7A,65,20,20,20,3A,20,05FC    [1590]
2180 DATA 26,00,00,CD,0D,A1,DD,2A,05CA    [1185]
2190 DATA 1E,A6,DD,36,00,00,3A,4D,0612    [1835]
2200 DATA BE,CB,47,21,68,A0,C2,B1,11A4    [1003]
2210 DATA A5,3A,51,BE,FD,21,20,A6,1194    [1546]
2220 DATA FD,77,00,1E,00,CD,0D,A1,1007    [2254]
2230 DATA DD,77,01,DD,23,1C,FD,BE,139E    [1849]
2240 DATA 00,20,F2,2A,1E,A6,73,3A,124D    [1441]
2250 DATA 22,A6,A7,C8,43,23,7E,CD,122C    [1500]
2260 DATA 7D,A4,3E,20,CD,5A,BB,23,0D3A    [1407]
2270 DATA 10,F4,C3,05,A6,D5,CD,FE,0F6C    [1756]
2280 DATA A5,3A,17,A6,F6,0A,CD,FA,0CF1    [1482]
2290 DATA A5,CD,E7,A5,CD,F6,A5,D1,0F57    [922]
2300 DATA 3A,4C,BE,CB,5F,3A,51,BE,0FD1    [1724]
2310 DATA C8,F1,C3,88,A1,F5,3E,11,0FAD    [1382]
```

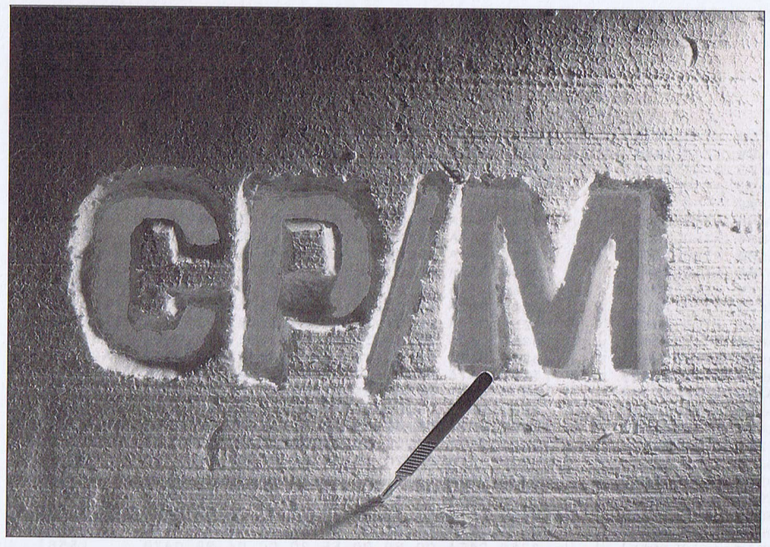
Listing. "BASIC 765"


```
2320 DATA C3,76,9F,CB,5F,C2,88,A1,0FD9    [1579]
2330 DATA 3A,4E,BE,21,48,A1,CB,5F,0958    [1974]
2340 DATA 20,03,21,5A,A1,C3,B1,A5,085C    [1942]
2350 DATA 73,65,63,74,6F,72,20,76,08CC    [1278]
2360 DATA 65,72,69,66,69,65,64,20,0A98    [1947]
2370 DATA 21,00,66,61,69,6C,65,64,0AA6    [1829]
2380 DATA 20,21,00,CD,FE,A5,CD,70,0B54    [1733]
2390 DATA A1,AF,32,18,A6,C3,05,A6,05F8    [1187]
2400 DATA 3E,07,CD,FA,A5,CD,E7,A5,078A    [2581]
2410 DATA CD,A3,A1,CB,5F,20,09,CB,054F    [2044]
2420 DATA 67,20,ED,CB,6F,28,F1,C9,08AA    [1520]
2430 DATA CD,05,A6,21,91,A1,C3,B1,08F1    [1235]
2440 DATA A5,64,72,69,76,65,20,6E,0645    [2003]
2450 DATA 6F,74,20,72,65,61,64,79,0696    [1786]
2460 DATA 20,21,00,21,00,00,2B,7C,0EF3    [2145]
2470 DATA B5,20,FB,3E,08,CD,FA,A5,1122    [1372]
2480 DATA CD,F6,A5,3A,4C,BE,CB,7F,11C2    [2321]
2490 DATA C8,CB,77,20,EE,C9,3D,C2,1414    [2335]
2500 DATA AE,A5,21,F0,A1,DD,7E,00,12C8    [597]
2510 DATA FE,54,D2,B1,A5,32,18,A6,1374    [1211]
2520 DATA F5,CD,FE,A5,3E,0F,CD,FA,0EFB    [2266]
2530 DATA A5,CD,E7,A5,F1,CD,FA,A5,1065    [1559]
2540 DATA 21,05,A6,E5,CD,A3,A1,CB,0DCD    [1211]
2550 DATA 5F,20,9D,CB,6F,28,F5,C9,0FF4    [1789]
2560 DATA 69,6E,76,61,6C,69,64,20,0EDB    [1181]
2570 DATA 74,72,61,63,6B,20,21,00,0CE4    [1580]
2580 DATA CD,FE,A5,3E,04,CD,FA,A5,0B10    [2139]
2590 DATA CD,E7,A5,CD,F6,A5,CD,05,0C1F    [630]
2600 DATA A6,3A,22,A6,A7,C8,21,35,083B    [1196]
2610 DATA A2,06,08,3A,4C,BE,4F,CD,0880    [1393]
2620 DATA BC,A5,CB,21,CD,2C,A2,10,0C3E    [1954]
2630 DATA F6,C3,BC,A5,3E,31,38,02,0AC5    [2375]
2640 DATA 3E,30,C3,5A,BB,46,61,75,0A76    [2417]
2650 DATA 6C,74,20,20,20,20,20,20,03BC    [1149]
2660 DATA 20,20,20,20,20,3A,20,00,0244    [1524]
2670 DATA 0A,0D,57,72,69,74,65,20,042A    [1504]
2680 DATA 70,72,6F,74,65,63,74,65,074C    [1840]
2690 DATA 64,20,3A,20,00,0A,0D,52,05B1    [1187]
2700 DATA 65,61,64,79,20,20,20,20,05C3    [1232]
2710 DATA 20,20,20,20,20,20,20,3A,0EBE    [1433]
2720 DATA 20,00,0A,0D,54,72,61,63,0F7D    [1832]
2730 DATA 6B,20,30,20,20,20,20,20,0DC7    [1895]
2740 DATA 20,20,20,20,3A,20,00,0A,103A    [1533]
2750 DATA 0D,54,77,6F,20,73,69,64,1289    [2123]
2760 DATA 65,20,20,20,20,20,20,20,0F73    [910]
2770 DATA 20,3A,20,00,0A,0D,48,65,1000    [1711]
2780 DATA 61,64,20,61,64,72,65,73,0D30    [1415]
2790 DATA 73,20,20,20,20,20,3A,20,0A55    [961]
2800 DATA 00,0A,0D,55,6E,69,74,20,0ADD    [636]
2810 DATA 73,65,6C,65,63,74,20,31,0E59    [1327]
2820 DATA 20,20,20,3A,20,00,0A,0D,0B85    [1056]
2830 DATA 55,6E,69,74,20,73,65,6C,0E30    [1073]
2840 DATA 65,63,74,20,30,20,20,20,07AA    [1801]
2850 DATA 3A,20,00,0A,0D,00,FE,02,07BD    [1106]
2860 DATA D2,AE,A5,A7,3A,1B,A6,28,0975    [1519]
2870 DATA 13,21,15,A3,DD,7E,00,FE,0B35    [1442]
2880 DATA 21,D2,B1,A5,FE,02,DA,B1,0CEC    [1482]
2890 DATA A5,32,1B,A6,3D,07,07,07,08AC    [2178]
2900 DATA 2F,F5,01,7E,FB,3E,03,CD,05D4    [2214]
2910 DATA FA,A5,F1,F6,03,CD,FA,A5,08C5    [1247]
2920 DATA 3E,03,C3,FA,A5,69,6E,76,04FC    [1384]
2930 DATA 61,6C,69,64,20,73,74,65,049E    [1141]
2940 DATA 70,20,72,61,74,65,20,21,0703    [1667]
2950 DATA 00,FE,01,DA,AE,A5,28,58,06F2    [1558]
2960 DATA 3D,47,DD,23,DD,23,10,FA,06F4    [1527]
2970 DATA 32,21,A6,2A,1E,A6,3A,21,0F98    [1027]
2980 DATA A6,47,3A,18,A6,77,23,3A,0F51    [1396]
2990 DATA 15,A6,77,23,DD,7E,FE,77,1147    [1894]
3000 DATA 23,3A,19,A6,77,23,DD,2B,11D2    [1581]
3010 DATA DD,2B,10,E6,CD,FE,A5,3A,146A    [1471]
3020 DATA 17,A6,F6,0D,CD,FA,A5,CD,13E5    [1873]
3030 DATA E7,A5,3A,19,A6,CD,FA,A5,0ECD    [2415]
3040 DATA 3A,21,A6,CD,FA,A5,3A,1D,0DA6    [2277]
3050 DATA A6,CD,FA,A5,3E,E5,CD,FA,0E98    [925]
3060 DATA A5,F3,2A,1E,A6,C3,E1,9F,1067    [2115]
3070 DATA DD,7E,00,32,21,A6,18,CC,0F6C    [1803]
3080 DATA FD,21,1D,A6,18,04,FD,21,0D83    [1328]
3090 DATA 1C,A6,3D,C2,AE,A5,21,AC,0EDD    [1402]
3100 DATA A3,DD,7E,00,A7,CA,B1,A5,0B2B    [1782]
3110 DATA FD,77,00,C9,69,6E,76,61,091F    [675]
```

Listing. "BASIC 765"

```
3120 DATA 6C,69,64,20,47,41,50,20,0797    [1134]
3130 DATA 21,00,3D,C2,AE,A5,DD,B6,0BCE    [936]
3140 DATA 01,28,1E,DD,7E,00,A7,21,0952    [1578]
3150 DATA FA,A3,C2,B1,A5,DD,4E,01,0C5D    [1569]
3160 DATA 06,05,11,F0,A3,1A,13,B9,049D    [2012]
3170 DATA 1A,28,06,13,10,F7,C3,B1,056C    [1999]
3180 DATA A5,32,19,A6,A7,3E,FF,28,055C    [1750]
3190 DATA 03,DD,7E,00,32,1A,A6,C9,075F    [1737]
3200 DATA 01,01,02,02,04,03,08,04,0473    [1309]
3210 DATA 10,05,69,6E,76,61,6C,69,05A4    [2048]
3220 DATA 64,20,73,65,63,74,6F,72,1090    [1512]
3230 DATA 20,73,69,7A,65,20,21,00,1024    [1490]
3240 DATA 3E,40,18,01,AF,32,17,A6,0E77    [1475]
3250 DATA C9,CD,93,A4,CD,43,A4,C5,1232    [1869]
3260 DATA E5,06,10,7E,CD,7D,A4,3E,137F    [1921]
3270 DATA 20,CD,5A,BB,23,10,F4,CD,127E    [1821]
3280 DATA 4B,A4,E1,C1,CD,56,A4,E6,13E4    [2407]
3290 DATA 0F,20,F9,CD,65,A4,78,B1,0DB5    [1603]
3300 DATA 20,DA,C9,7A,CD,7D,A4,7B,0D88    [2267]
3310 DATA CD,7D,A4,3E,20,CD,5A,BB,0D8C    [1573]
3320 DATA CD,5A,BB,C3,5A,BB,7E,FE,1090    [1729]
3330 DATA 20,30,02,3E,2E,CD,5A,BB,0EA2    [1222]
3340 DATA 23,13,0B,79,C9,3E,0D,CD,0DC1    [1990]
3350 DATA 5A,BB,3E,0A,CD,5A,BB,CD,0A10    [1092]
3360 DATA 09,BB,D0,FE,20,CC,06,BB,0A7B    [556]
3370 DATA FE,FC,C0,E1,C9,F5,0F,0F,0A69    [1229]
3380 DATA 0F,0F,CD,86,A4,F1,E6,0F,0BE5    [1801]
3390 DATA C6,30,FE,3A,38,02,C6,07,0BB3    [1316]
3400 DATA C3,5A,BB,3A,19,A6,A7,20,0A36    [2566]
3410 DATA 02,3E,01,47,21,80,00,29,0890    [1754]
3420 DATA 10,FD,44,4D,2A,1E,A6,11,054B    [870]
3430 DATA 00,00,C9,CD,93,A4,CD,43,0555    [1623]
3440 DATA A4,CD,56,A4,E6,3F,20,F9,063D    [1230]
3450 DATA CD,65,A4,78,B1,20,EF,C9,08CF    [1163]
3460 DATA 3D,C2,AE,A5,DD,5E,00,DD,0790    [1999]
3470 DATA 56,01,7A,A7,21,D7,A4,CA,079C    [2553]
3480 DATA B1,A5,ED,53,1E,A6,C9,69,11D6    [1561]
3490 DATA 6E,76,61,6C,69,64,20,44,10B8    [2440]
3500 DATA 4D,41,20,61,64,72,65,73,0FB5    [1494]
3510 DATA 73,20,21,00,3D,C2,AE,A5,126A    [1368]
3520 DATA DD,6E,00,DD,66,01,ED,5B,1373    [1542]
3530 DATA 1E,A6,73,23,72,C9,3D,C2,11DE    [2060]
3540 DATA AE,A5,DD,6E,00,DD,66,01,0D9C    [1195]
3550 DATA 3A,4C,BE,77,23,36,00,C9,0D35    [2293]
3560 DATA 3D,C2,AE,A5,21,23,A5,DD,0CB4    [1861]
3570 DATA 7E,00,FE,02,D2,B1,A5,32,0D0A    [1606]
3580 DATA 15,A6,C9,69,6E,76,61,6C,0FA2    [1847]
3590 DATA 69,64,20,68,65,61,64,20,0D67    [1975]
3600 DATA 73,65,6C,65,63,74,20,21,0DB3    [918]
3610 DATA 00,3D,C2,AE,A5,21,4C,A5,0942    [1021]
3620 DATA DD,7E,00,FE,04,D2,B1,A5,098F    [1430]
3630 DATA 32,16,A6,C9,69,6E,76,61,0901    [1423]
3640 DATA 6C,69,64,20,75,6E,69,74,0AB1    [824]
3650 DATA 20,73,65,6C,65,63,74,20,0AEA    [1502]
3660 DATA 21,00,3E,01,18,01,AF,32,08B4    [1020]
3670 DATA 22,A6,C9,21,4B,BE,7E,A7,0632    [1444]
3680 DATA C8,47,3A,22,A6,A7,C8,23,0617    [1368]
3690 DATA 7E,CD,7D,A4,3E,20,CD,5A,0519    [982]
3700 DATA BB,10,F4,C9,3A,22,A6,A7,084B    [1591]
3710 DATA C8,21,E9,9E,3E,7C,CD,5A,0909    [1807]
3720 DATA BB,7E,E6,7F,CD,5A,BB,CB,083B    [1109]
3730 DATA 7E,23,28,F5,3E,0A,CD,5A,06B5    [2057]
3740 DATA BB,3E,0D,CD,5A,BB,7E,A7,120D    [1455]
3750 DATA C8,CD,6F,A4,18,DE,21,C5,1122    [1743]
3760 DATA A5,3A,22,A6,A7,C8,CD,BC,1173    [2403]
3770 DATA A5,21,E3,A5,7E,23,A7,C8,1398    [1181]
3780 DATA CD,5A,BB,18,F7,69,6E,76,1292    [1746]
3790 DATA 61,6C,69,64,20,6E,75,6D,1208    [842]
3800 DATA 62,65,72,20,6F,66,20,70,0C44    [1611]
3810 DATA 61,72,61,6D,65,74,65,72,0D67    [1438]
3820 DATA 20,21,00,07,0A,0D,00,D5,0A76    [1259]
3830 DATA 3A,16,A6,5F,3A,15,A6,87,0E95    [1930]
3840 DATA 87,B3,D1,C3,FA,A5,DF,8C,11A0    [1326]
3850 DATA 9E,C9,DF,89,9E,C9,DF,80,1027    [1934]
3860 DATA 9E,01,7E,FB,C9,DF,86,9E,0AE4    [1203]
3870 DATA 3A,5F,BE,A7,C8,ED,5B,46,0AD8    [1522]
3880 DATA BE,DF,83,9E,C9,00,00,40,088F    [1714]
3890 DATA 00,02,00,0A,2A,52,00,C0,06B0    [1359]
3900 DATA 00,00,01,00,00,00,00,00,0855    [1263]
3910 DATA 00,00,00,00,00,00,00,00,0708    [1031]
```

Listing. "BASIC 765"



Man liest es immer wieder: Die Hochsprachen sind im Kommen. Seit C in der Landschaft der Programmiersprachen Wurzeln gefaßt hat, ist die Assemblerprogrammierung immer mehr ins Abseits geraten. Aber solange es Probleme gibt, die sich besser auf der Maschinenebene lösen lassen (man nennt das Maschinen- oder Systemprogrammierung), solange wird es auch die 'echten' Programmierer geben...

Control Program for Microcomputers – das unbekannte Wesen

CP/M – Eine Einführung für Aufsteiger

Zwar werden Sie durch diesen Artikel nicht zum hochbezahlten Spitzenprogrammierer, aber es kann nie verkehrt sein, wenn man die Abläufe im Inneren seines Rechners besser kennt als z.B. die C-Freaks.

Die Voraussetzungen...

Wie Sie sicherlich verstehen werden, kann man durch einen mehrseitigen Artikel nicht die Programmierung unter einem Betriebssystem wie CP/M erlernen. So ist es vielmehr als Anregung für diejenigen gedacht, die bisher schon in Assembler programmiert haben und nun die Stärken von CP/M nutzen möchten. Auf Sekundärliteratur sollten Sie bei tiefergehenden Arbeiten nicht verzichten.

...und das Werkzeug

Spätestens jetzt sollten Sie sich vergewissern, daß Sie einen Texteditor, einen Z80-Assembler sowie einen Linker in Ihrer Programmsammlung vorfinden und diese auch bedienen können. Sie gehören nämlich zur Mindestausstattung für die Programmierung unter CP/M. Dabei sieht der Ablauf einer Programmentwicklung in etwa so aus:

1. Erstellen des Sourceprogrammes mit dem Texteditor
2. Assemblieren des Sourceprogrammes
3. Linken des Zwischencodes zum .COM-File
4. Ausprobieren und die Schritte 1.-4. wiederholen...

Natürlich sollte man auch mal zu einem Ende kommen, und das ist spätestens dann erreicht, wenn a) einem nach einer durchgearbeiteten Nacht das Programm immer noch abstürzt, oder b) es wider Erwarten zum Laufen gebracht werden konnte. Ich werde später noch auf diese beiden Möglichkeiten zurückkommen...

BDOS, BIOS und der Rest der Welt

Sicher haben Sie den einen oder anderen Begriff schon einmal gehört, und da es fast schon ein Muß für einen CP/M-Artikel ist, sei auch hier kurz erklärt, was sich hinter den Abkürzungen verbirgt.

Im Grunde besteht jedes CP/M-Betriebssystem aus drei Hauptteilen: dem CCP, dem BDOS und dem BIOS. Der CCP (engl. Abk. für Console Command Processor) bildet, wie der Name schon sagt, die Benutzeroberfläche des CP/Ms. Er ermöglicht es, residente Befehle einzugeben und transiente Programme (.COM-Files) aufzurufen. Das BDOS (engl. Abk. für BASIC Disk Operating System) ist (sollte) bei allen CP/M-Rechnern gleich (sein) und bildet zusammen mit dem CCP den Hardwareunabhängigen Teil des Betriebssystems. Er stellt die Ein-/Ausgabefunktionen und hauptsächlich die Dateiverwaltungsoperationen zur Verfügung. Das BIOS (engl. Abk. für Basic Input Output System) ist der Hardware-abhängige Teil des CP/Ms. Er ist auf die jeweilige Hardware des Rechners zugeschnitten und stellt wiederum dem BDOS die grundlegenden Ein-/Ausgaberoutinen zur Verfügung.

Von den Funktionen...

Auch hier muß gleich von vornherein erwähnt werden, daß aus Platzgründen nicht alles angesprochen werden kann. Es sei in diesem Zusammenhang nochmal auf Fachliteratur für CP/M hingewiesen, die in großer Auswahl erhältlich ist.

Bevor Sie weiterlesen, sollten Sie einmal einen Blick in Tabelle 1 werfen. Dort sind die BDOS-Funktionen und deren Operationen aufgelistet. "Eine ganze Menge", werden Sie vermutlich denken. Naja, ganz brauchbar. "Und wie werden die aufgerufen?", werden Sie sich nun fragen. Nun, grundsätzlich wird das BDOS immer mit einem CALL 5 erreicht, dort steht

dann ein Jump in das BDOS. Im C-Register muß dann die Funktionsnummer stehen, damit das BDOS die entsprechende Routine aufrufen kann. So erreicht man über einen Vektor eine Reihe von Betriebssystemfunktionen.

"Und die Parameter?" Stimmt, da haben Sie aufgepaßt. Eintrittsparameter werden immer im DE-Register übergeben, ist nur ein Acht-Bit-Wert gefordert, so wird logischerweise nur das E-Register benutzt. Liefern die Routinen irgendwelche Resultate zurück, so werden 16-Bit-Resultate im HL-Register und Acht- Bit-Resultate im Akku zurückgeliefert. Wie Sie sehen, ist es ganz einfach.

Aber leider gibt es noch viel mehr, was für die Programmierung wichtig ist. So gibt es z.B. wie in jedem Betriebssystem bestimmte Speicherbereiche, die für einem bestimmten Zweck reserviert sind, man nennt sie Systemvariablen. Und auch hierzu kann aus der Fülle nur das Wesentliche dargestellt werden. (Siehe dazu Bild 1 und 2.)

...zum Programm

Wie bereits anfangs angedeutet, gibt es bei der Programmierung zwei grundverschiedene Methoden. Die eine besteht darin, daß man sich zuerst überlegt, welche Unterprogramme gebraucht werden und sich alles aufschreibt. Die andere ist nur für den 'echten' Programmierer geeignet: Er schreibt sich nichts auf, dazu hat er einen Kopf, und programmiert einfach 'drauf los'. Das hat meistens zur Folge, daß die Entwicklung mit a) (siehe dort) abgebrochen wird...

Wir wollen also hier die "informatische Methode" (Ausdruck des Autors) anwenden. Fragen wir uns somit zuerst, wozu das Programm eigentlich dienen soll. Nun, um Ihnen das Denken ein wenig abzunehmen und keine Monster-Programme (das sind keine PacMan-Spiele, sondern wild programmierte Programme, Anm. d. Autors) zu entwickeln, wollen wir ein Programm schreiben, welches eine Datei auf dem Bildschirm ausgibt. "Das gibt's ja schon!", werden Sie jetzt vermutlich sagen. Das weiß ich auch, aber man kann als Einsteiger ja nicht mit einem Diskettenmonitor anfangen...

Überlegen wir also, welche Unterprogramme wir dazu brauchen. Zuerst einmal geht man den Ablauf kurz im Kopf (oder auf dem Papier) durch und sollte dann auf Folgendes kommen:

1. Öffnen der Datei
2. Einlesen des nächsten Records
3. Ausgeben des aktuellen Records
4. Kein EOF, dann 2.
5. Schließen der Datei

Haben Sie sich vorhin die Tabelle 1 mit den BDOS Funktionen angesehen, so werden Sie feststellen, daß man diese Programmteile ziemlich leicht mit den BDOS Funktionen lösen kann.

Tabelle 2 – Die BIOS- Einsprungtabelle

Adresse	Name	Funktion
AD00	BOOT	Laden des kompletten CP/M Systems
AD03	WBOOT	Laden des BDOS und des CCP
AD06	CONST	A: = FF, wenn eine Taste gedrückt wurde A: = 00, wenn keine Taste gedrückt wurde
AD09	CONIN	A: = Zeichencode - auf eine Taste warten oder sofort zurückkehren,wenn bereits eine Taste gedrückt wurde
AD0C	CONOUT	Zeichen im C Register ausgegeben
AD0F	LIST	Zeichen im C Register an den Drucker senden
AD12	PUNCH	Zeichen im C Register an die V.24 senden
AD15	READER	A: = Zeichen von der V.24 - wenn kein Zeichen empfangen wurde, gilt A: = 1A (EOF)
AD18	HOME	Schreib-/Lesekopf auf Spur 0 positionieren
AD1B	SELDSK	Laufwerk im C Register selektieren, HL: = DPH Adresse oder HL: = 0000,wenn das Laufwerk nicht existiert
AD1E	SETTRK	Spurnummer im C Register setzen
AD21	SETSEC	Sektornummer im C Register setzen
AD24	SETDMA	DMA Adresse im BC Register setzen
AD27	READ	Einen Record (128 Byte) nach DMA Adresse lesen, A: = 00, wenn die Operation erfolgreich gewesen ist
AD2A	WRITE	Einen Record von der DMA-Adresse schreiben, A: = 0, wenn die Operation erfolgreich gewesen ist, das C-Register enthält die Allocation Information:C = 0, normale Operation C = 1, Directory-Zugriff, Record sofort zurückschreiben C = 2, der zugehörige Block ist leer. (Da in einen physikalischen Sektor 4 logische Records passen, muß, bevor ein Record geschrieben werden kann, der komplette Sektor gelesen werden, um die anderen Records nicht zu überschreiben. Soll der Sektor nicht eingelesen werden, so ist C = 2 zu setzen.)
AD2D	LISTST	A: = 00, wenn der Drucker offline ist A: = FF, wenn der Drucker online ist
AD30	TRANS	HL: = BC, normalerweise Umrechnung der Sektornummern, wird aber bei AMSDOS nicht benötigt
AD33	SYSTEM	Aufruf (JUMP!)der Betriebssystemroutine, deren Adresse nach dem CALL folgt (diese Routine gehört nicht zum CP/M Standard, kann aber dazu benutzt werden, um Routinen aus dem Betriebssystem aufzurufen). Um z.B. die TXT OUTCHAR Routine aufzurufen, muß folgendes programmiert werden: CALL AD33 DW BB5A Zu beachten ist dabei,daß der CALL wie ein JP behandelt wird und daß das Programm nicht (!) hinter dem DW fortgeführt wird.

Abb. 2: Anhand der BIOS-Einsprungtabelle lassen sich Systemroutinen erkennen und schnell für eigene Projekte nutzen.

Tabelle 1 – Die BDOS Funktionen

0 = Systemzurücksetzen und BDOS sowie CCP laden
1 = Konsoleingabe
2 = Konsolausgabe
3 = ReaderEingabe
4 = Punch Ausgabe
5 = List Ausgabe
6 = Direkte Konsoleinausgabe
7 = I/O Byte holen
8 = I/O Bytesetzen
9 = String ausgeben
10 = Konsolpuffer lesen
11 = Konsolstatus holen
12 = Versionsnummerholen
13 = Disksystem zurücksetzen
14 = Default Laufwerk auswählen
15 = Datei öffnen
16 = Datei schließen
17 = Directory nach erstem Eintrag durchsuchen
18 = Directory nach nächstem Eintrag durchsuchen
19 = Datei löschen
20 = sequentiell lesen
21 = sequentiell schreiben
22 = Datei erzeugen
23 = Datei umbenennen
24 = Login Vektor holen
25 = Default Laufwerk holen
26 = DMA Adresse setzen
27 = Belegungsverzeichnis holen
28 = Schreibschutz setzen
29 = R/O Vektor holen
30 = Dateiattribute setzen
31 = Diskparameter holen
32 = Usernummer setzen / holen
33 = wahlfrei lesen
34 = wahlfrei schreiben
35 = Dateigröße berechnen
36 = wahlfreien Block setzen
37 = Laufwerk zurücksetzen
38 = unbenutzt
39 = unbenutzt
40 = wahlfrei schreiben und leere Blöcke mit Nullen füllen

Abb. 1: Tabelle 1 zeigt Ihnen die BDOS-Funktionen in übersichtlicher Form.

Bild 1 - Die Speicheraufteilung auf dem CPC unter CP/M

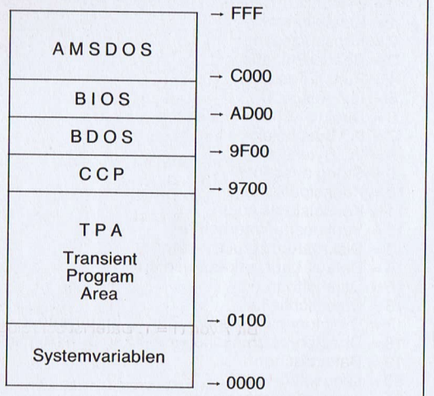


Abb. 3: Der Speicher des CPC wird unter CP/M so aufgeteilt, wie es diese Abbildung zeigt.

Bild 2 - Die Systemvariablen

0000 Jump auf den BIOS Warm Boot
– um ein Programm zu beenden, sollte ein Jump auf diese Adresse ausgeführt werden
0003 IO-Byte
– über diese Bitmaske werden die physikalischen Geräte den logischen zugeordnet
0004 Drive/User Default-Einstellung
– im unteren Halbbyte steht das Laufwerk, im oberen Halbbyte die Usernummer
0005 Jump in das BDOS
– Haupteinsprung für BDOS Funktionen
005C Erster FCB (FileControlBlock)
– wird ein Programm aufgerufen und ein Dateiname angegeben, so wird dessen FCB hier gespeichert
006C Zweiter FCB falls angegeben
0080 Default DMA Bereich
– wird ein Programm aufgerufen, so enthält 0080 die Länge der Eingabezeile, die ab 0081 gespeichert ist

Bei größeren Bereichen ist jeweils nur die erste Adresse angegeben. Nicht angegebene Bereiche sind reserviert und werden vom CPC Betriebssystem benutzt.

Abb. 4: In dieser Abbildung finden Sie die System-Variablen und die dazugehörigen Erläuterungen.

Bild 3 – Die Programierung

```
; ***                                   DISPLAY.COM Hauptpro-
                                        gramm
                org 100h                ; Code nach 100h (TPA Anfang)
                                          legen
                ld (old__stack),sp      ; alten Stackpointer sichern
                ld sp,stack__area       ; und neuen Stack festlegen
                call set__default       ; Default DMA Adresse vor dem
                                          Open setzen
                call open__file         ;  angegebene Datei öffnen
                jr nc,exit              ; ein Fehler ist aufgetreten

loop:           call read__record       ; nächsten Record einlesen
                jr nc,exit              ; Datei ist zu Ende
                call display__record    ; Record-Inhalt auf dem Bild-
                                          schirm anzeigen
                jr c,loop               ; weiter, wenn kein EOF aufge-
                                          treten ist

exit:           call close__file        ; Datei schließen
                ld sp,(old__stack)      ; alten Stackpointer zurückholen
                ret                     ; und zum CCP zurückkehren

; ***           Unterprogramme
set__default:   ld de,80h               ; Default DMA Area
                ld c,26                 ; BDOS DMA Adresse setzen
                jp 5                    ; das BDOS aufrufen

open__file:     ld de,5ch               ; Adresse des 1. FCBs
                ld c,15                 ; BDOS Datei öffnen
                xor a
                ld (5ch+32),a           ; cr auf 0 setzen
                call 5
                inc a                   ; A=FF ?
                scf                     ; OK Flag setzen
                ret nz                  ; Operation war erfolgreich,
                                          wenn A<>FF
                and a                   ; Carry löschen
                ret
close__file:    ld de,5ch               ; Adresse des 1. FCBs
                ld c,16                 ; BDOS Datei schließen
                jp 5

read__record:   ld de,5ch               ; Adresse des 1. FCBs
                ld c,20                 ; BDOS sequentiell lesen
                call 5
                or a                    ; A=0 bedeutet die Operation
                                          war ok
                ret nz                  ; das Carry ist schon durch den
                                          OR A gelöscht
                scf                     ; OK Flag setzen
                ret

display__record: ld hl,80h              ; Default DMA Area
                 ld b,128               ; 128 Zeichen pro Record aus-
                                          geben

display__loop:  ld a,(hl)               ; aktuelles Zeichen nach A holen
                cp 1ah                  ; EOF ?
                ret z                   ; das Carry ist schon durch den
                                          CP 1AH gelöscht
                call print__char        ; Zeichen im Akku ausgeben
                inc hl                  ; nächstes Zeichen im Buffer
                                          adressieren
                djnz display__loop      ; und auslesen
                scf                     ; Signal "weiterlesen"
                ret

print__char:    push hl
                push bc                 ; Register retten
                ld e,                   ; das Zeichen nach E holen
                ld c,2                  ; BDOS Zeichen ausgeben
                call 5
                pop bc
                pop hl                  ; Register zurückholen
                ret

; ***           Variablen
old__stack:     dw 0
                ds 16                   ; 16 Bytes für den Stack
                                          reservieren

stack__area     equ $
```

Hinweis
Die Notation in diesem Assemblerprogramm entspricht der des M80 von Microsoft. Andere Assembler dürften eine ähnliche Syntax haben, um aber Schwierigkeiten vorzubeugen, sind hier kurz die Besonderheiten angegeben:
Hexadezimale Zahlen werden durch ein nachgestelltes 'h' kenntlich gemacht (z.B. 1ah), alle anderen Zahlen sind Dezimalzahlen.
'dw' steht für 'define word' und sollte ebenso wie 'ds' ('define space') von anderen Assemblern angenommen werden.
Der 'equ $' weist dem vorstehenden Label (hier 'stack__area') den aktuellen Program Counter zu. Alternativ kann man auch schreiben: stack__area ds 0

Abb. 5: Das vollständige Beispielprogramm zum anschauen und nachvollziehen.

Bild 4 – Das IO-Byte

Unter CP/M werden vom BIOS vier logische Geräte unterstützt: CON: die Konsole, RDR: der Reader, PUN: der Puncher und LST: der Drucker.
Jedem dieser logischen Geräte können wiederum eines von vier physikalischen Geräten zugeordnet werden. Diese Zuordnung kann z.B. mit dem SETUP.COM geändert werden. Während des Betriebes holt sich das BIOS das IO-Byte aus Adresse 0003 und führt die entsprechende Routine aus, um z.B. ein Zeichen auf den Bildschirm auszugeben. Nachfolgend finden Sie die Zuordbarkeit der Geräte in Verbindung mit dem IO-Byte:

IO-Byte 76543210	Zuordnung	Gerätebezeichnung
xxxxxx00	CON: = TTY:	SIO Kanal A
xxxxxx01	CON: = CRT:	Tastatur und Bildschirm
xxxxxx10	CON: = BAT:	RDR: Eingabe und LST: Ausgabe
xxxxxx11	CON: = UC1:	SIO Kanal B
xxxx00xx	RDR: = TTY:	SIO Kanal A
xxxx01xx	RDR: = PTR:	EOF Eingabe
xxxx10xx	RDR: = UR1:	SIO Kanal B
xxxx11xx	RDR: = UR2:	Tastatur
xx00xxxx	PUN: = TTY:	SIO Kanal A
xx01xxxx	PUN: = PTP:	kurzgeschlossene Ausgabe
xx10xxxx	PUN: = UP1:	SIO Kanal B
xx11xxxx	PUN: = UP2:	Bildschirm
00xxxxxx	LST: = TTY:	SIO Kanal A
01xxxxxx	LST: = CRT:	Bildschirm
10xxxxxx	LST: = LPT:	Centronics Port
11xxxxxx	LST: = UL1:	SIO Kanal B

Mit Sicherheit haben Sie jetzt schon einmal in Adresse 0003 nachgesehen und festgestellt, daß das IO-Byte default-mäßig auf 81 steht. Binär wäre das 10000001 und deutet auf folgende Standard-Zuordnung:
CON: = CRT:, RDR: = TTY:, PUN: = TTY:, LST: = LPT:

Abb. 6: Diese Tabelle zeigt Ihnen die möglichen Belegungen des I/O-Bytes auf

Bild 5 – Der FCB-Aufbau

Ganz zu Anfang wurde der Begriff FCB in Verbindung mit dem Dateinamen bei den Dateioperationen verwendet. FCB steht für File Control Block und enthält alle diejenigen Informationen, die das BDOS zum Bearbeiten eines Files braucht. Unter CP/M können beliebig viele Dateien gleichzeitig geöffnet sein, es muß nur für jede geöffnete Datei ein FCB existieren.
Nachfolgend finden Sie den Inhalt des FCBs:

Byte	Name	Bedeutung
0	dr	Laufwerksnummer (0-16) 0 = Default- Laufwerk 1 = Laufwerk A 2 = Laufwerk B . . 16 = Laufwerk P
1-8	f1-f8	Dateiname in Großschrift (Bit 7 ist 0)
9-11	t1-t3	Dateityp in Großschrift (Bit 7 ist 0) Bit 7 von t1 = 1: Datei ist schreibgeschützt Bit 7 von t2 = 1: SYS-Datei, wird bei DIR nicht gelistet
12	ex	aktuelle Extensionsnummer
13	s1	reserviert
14	s2	reserviert
15	rc	Blockzähler für die aktuelle Extension
16-31	d0-dn	reserviert
32	cr	Blocknummer des aktuellen Blocks
33-35	r0-r2	optionaler Anhang für den wahlfreien Zugriff in r0 steht das LSB der 16Bit Record-Nummer in r1 steht das MSB der 16Bit Record-Nummer r2 ist für das System reserviert

Wie Sie sehen, ist ein FCB mindestens 33 Byte lang, soll auf eine Datei wahlfrei zugegriffen werden, ist der FCB 36 Byte lang. Wird ein FCB zum erstenmal vom BDOS benutzt, muß dr, f1-f8 und t1-3 angegeben werden und die Variable cr auf 0 gesetzt werden.

Abb. 7: In dieser Beschreibung finden Sie den Aufbau des FCB.

Fühlen Sie sich jetzt genau richtig, um sich das Programm anzusehen, dann werfen Sie einmal einen Blick auf Bild 3. Dort finden Sie das komplette Programm.
Wie Sie gesehen haben, kann man auch unter Assembler übersichtlich und strukturiert programmieren. Ein Label mit dem Namen 'set__default__dma' sagt eben mehr aus als eines mit dem Namen 'sdd', und so liest sich ein Assemblerprogramm fast wie ein Buch…

Als nächstes…

Da wären wir nun: ein fertiges Programm, ein fertiger Leser und ein glühender Rechner. Nein, soweit sollten Sie es wirklich nicht treiben. Jetzt ist es an der Zeit, daß Sie eigene Veränderungen in das Programm einbauen. So könnten Sie zum Beispiel anstatt der Bildschirmausgabe eine Druckerausgabe einbauen. Die Funktionsnummer finden Sie in der Tabelle 1 und den Rest wissen Sie bereits. Versuchen Sie es ruhig…

Am Ende…

…angekommen hoffe ich, daß Sie Gefallen an der Programmierung unter CP/M gefunden haben und wünsche Ihnen viel Spaß dabei. Und damit Sie ohne Fremdliteratur für's Erste nicht ganz so hilflos darstehen, finden Sie in Bild 4 und 5 noch Informationen zu dem IO-Byte und dem FCB-Aufbau.

(Marco Pries/jb)

FÜR NIX GIBT'S NIX

Für den Programmhit des Monats
1.000,—

Und für den Top-Tip des Monats
500,—

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

DMV Daten & Medien Verlagsgesellschaft mbh — Fuldaerstr 6 – 3440 Eschwege

CPC-Listingservice

Eingabekontrolle für Listings

für 464-664-6128

Die Fehlersuche von abgetippten Listings ist manchmal eine lästige und nervenaufreibende Sache. Wenn Sie auch dieser Meinung sind, dann können wir Ihnen eine Hilfe anbieten.

Auf dieser Seite befinden sich drei Programme, die für jeden CPC-Rechner abgestimmt sind. Der "Checksummer" hilft Ihnen bei jeder Eingabe von Programmlistings.
Bevor Sie sich das Programm, das für Ihren Rechner geeignet ist, vornehmen. Sollten noch einige "Dinge" beachtet werden:

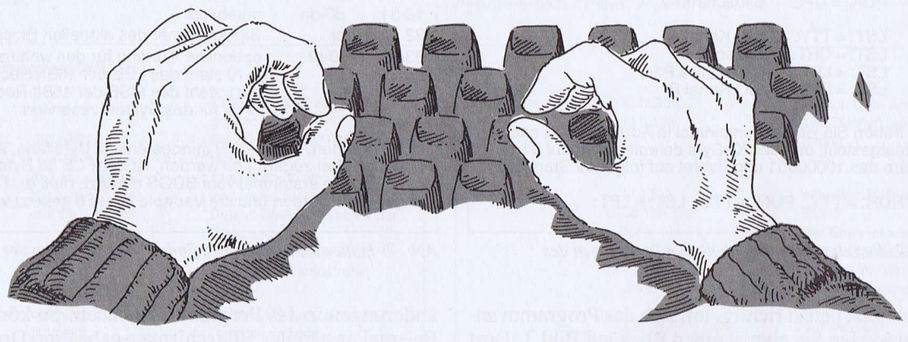
1. Das abgetippte Programm vor dem ersten Start abspeichern, da der Checksummer Maschinensprache enthält.

2. Um den Checksummer benutzen zu können, genügt es, das Programm einmal durch RUN zu starten. Es wird selbständig der Maschinencode eingelesen und aufgerufen.

3. Jedesmal, wenn die ENTER oder RETURN-Taste gedrückt wird, erscheint eine Quersumme der eingegebenen Zeile. Z.B. wenn ohne Zeile die ENTER-Taste gedrückt wird, erhalten Sie in eckigen Klammern, am Ende der Zeile eine <0>. Dadurch lassen sich eingetippte Listings gleich auf Fehler überprüfen.

4. Weitere Möglichkeiten: Möchten Sie ein Listing mit dem Checksummer ausdrucken, so läßt sich auch das bewerkstelligen. Z.B. mit **l**check,x,y. Wobei x = Ausgabekanal und y = Zeilennummer ist. 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 Checksummer ein- bzw. ausgeschaltet werden.

(Thomas Fippl/cd)



Version 464

```
10 REM *** CHECKSUM 464 ***                [1071]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,98,aa,c3           [799]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 06,dd,11,3a,bd,01,03,00           [923]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a2           [1665]
220 DATA c1,cd,a3,e7,e5,4e,23,46           [1294]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,3c,c4,e5,09,e3,cd,63           [1318]
250 DATA e1,21,a4,ac,cd,7a,a5,e1           [1223]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,96,f2,e3,cd,f6           [973]
280 DATA a5,cd,4e,c3,e1,7e,a7,c8           [738]
290 DATA cd,98,a5,cd,4e,c3,18,f5           [1601]
300 DATA 3a,24,ac,d6,08,47,7e,a7           [1761]
310 DATA c8,cd,45,e1,23,10,f7,c9           [1394]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,8a,ff,4f           [1826]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,5c,c3,3e,5b,cd,5c,c3           [1002]
430 DATA cd,79,ee,3e,5d,c3,5c,c3           [1170]
```

Version 664

```
10 REM *** CHECKSUM 664 ***                [1262]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,02,ac,c3           [333]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 4c,cb,11,5b,bd,01,03,00           [1095]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a9           [1682]
220 DATA c1,cd,69,e8,e5,4e,23,46           [1382]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,75,c4,e5,09,e3,cd,59           [798]
250 DATA e2,21,8a,ac,cd,7a,a5,e1           [1228]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,58,f3,e3,cd,f6           [1104]
280 DATA a5,cd,9b,c3,e1,7e,a7,c8           [1523]
290 DATA cd,98,a5,cd,9b,c3,18,f5           [1069]
300 DATA 3a,09,ac,d6,08,47,7e,a7           [1172]
310 DATA c8,cd,22,e2,23,10,f7,c9           [1848]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,ab,ff,4f           [1559]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,a3,c3,3e,5b,cd,a3,c3           [1480]
430 DATA cd,49,ef,3e,5d,c3,a3,c3           [1389]
```

Version 6128

```
10 REM *** CHECKSUM 6128 ***               [945]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,02,ac,c3           [333]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 49,cb,11,5e,bd,01,03,00           [1424]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a6           [1661]
220 DATA c1,cd,64,e8,e5,4e,23,46           [812]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,72,c4,e5,09,e3,cd,54           [1100]
250 DATA e2,21,8a,ac,cd,7a,a5,e1           [1228]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,53,f3,e3,cd,f6           [1107]
280 DATA a5,cd,98,c3,e1,7e,a7,c8           [1357]
290 DATA cd,98,a5,cd,98,c3,18,f5           [1651]
300 DATA 3a,09,ac,d6,08,47,7e,a7           [1172]
310 DATA c8,cd,1d,e2,23,10,f7,c9           [1782]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,ab,ff,4f           [1559]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,a0,c3,3e,5b,cd,a0,c3           [1906]
430 DATA cd,44,ef,3e,5d,c3,a0,c3           [1650]
```


SPORTLICH! SPORTLICH!

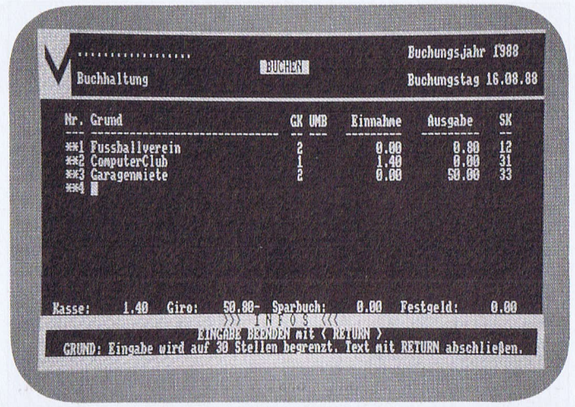
Vereins-Verwaltung

Wer ist heutzutage nicht Mitglied in irgendeinem Verein, wie Schulverein, Gesangsverein, Kulturverein oder Sportverein? Überall wirkt - meist im Hintergrund - der Schatzmeister oder Kassenwart. Solch ein Ehrenamt hat weniger mit Ehre, als vielmehr mit Arbeit zu tun, insbesondere Rechner-Arbeit. Wen erstaunt es da, wenn versucht wird, einen Teil der Abrechnungen auf den Kollegen Computer abzuwälzen. Dies hat auch der Kassenwart eines Sportvereins mit 400 Mitgliedern getan. Um die Kassengeschäfte bei jährlich ca. 1000 Buchungen zu erleichtern und die Abschlußarbeiten zu vereinfachen, schrieb er ein Buchhaltungsprogramm, das auf die speziellen Belange seines Vereins abgestellt ist.

Die hier abgedruckte Version wurde überarbeitet und überall dort geändert, wo das Verfahren zu sehr auf diesen Verein zugeschnitten war.
Das Programm ist mit dem CPC 6128 für 1 Laufwerk, Monochrom-Monitor und Epson-Drucker geschrieben. Änderungen sind auch für den Nichtprofi leicht möglich, da auch der Verfasser nur Hobby-Hacker ist und alles in BASIC geschrieben hat.
Doch nun ein paar Angaben zum Programm:

Grundsätzliches

1. Das Startprogramm (Listing 1) enthält das Eröffnungsbild, den deutschem Zeichensatz und Arbeitsmaske, und lädt dann das Haupt-(Buchungs-) Programm nach.
2. Listings
Das Hauptprogramm muß nach dem Abtippen unbedingt unter dem Namen gesichert werden, der in Zeile 920 des Startprogrammes festgelegt wurde.
3. Der Programmablauf basiert auf dem Doppelbuchungsverfahren mit den 4 Gegenkonten:
Kasse / Giro / Sparbuch / Festgeld
Die 36 Sachkonten unterteilen sich zunächst einmal in je 18 Einnahme- und 18 Ausgabekonten. Die Einzelkonten sind wiederum in die drei Sachkontengruppen
Sport / Veranstaltungen / Sonstiges
gegliedert, für die in den Übersichten und Abschlüssen Zwischenwerte ausgewiesen werden. Die Bezeichnungen der Sachkonten stehen in den DATA-Zeilen 200 - 220. Änderungen sind durch Ersetzen der obigen Konten oder Kontengruppen möglich. Die neuen Begriffe müssen jedoch dann noch, soweit erforderlich, entsprechend in die Zeilen 300,1110 und 2010 - 2070 übernommen werden.
4. Mit der Variablen in Zeile 110 wird die Höchstzahl der Buchungen festgelegt. Das Programm sichert danach automatisch, beginnt dann aber nicht von vorn, sondern arbeitet weiter. Dadurch entstehen neue Buchungsdateien, mit denen man eine fast beliebig große Zahl von Buchungen erfassen kann. Das Programm fordert nach dem Start auf, eine Buchungsdiskette an- und einzulegen. Dies ist zwar aus Platzgründen und zur Verwaltung mehrerer Jahre sinnvoll, muß aber nicht sein.
Die Buchhaltung arbeitet mit einer Grunddatei und den Buchungsdateien. Beide Dateiarten werden automatisch erzeugt und fortgeschrieben.
5. Das ganze Verfahren ist mit einer durchgängigen Benutzerführung ausgestattet, die den Anwender ständig begleitet. Die Informationen zum Weiterarbeiten stehen immer im INFO-Window.
6. Es gibt zwei Menü-Ebenen:
- Das Hauptmenü (Optionen)
- Das Auswahlmenü "Buchungen", mit dem die anzuzeigenden oder auszudruckenden Buchungen ausgewählt werden können.



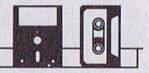
Solange das Hauptprogramm noch nicht richtig läuft, sollten die Zeilen 230 und 240 als REM-Zeilen erfaßt bleiben, um einen möglichen Totalverlust des Programmes auszuschließen und Fehler finden zu können.

Das Buchungsverfahren

1. Über das Menü – hier Optionen genannt – gelangt man zum Buchen. Hier wird nach Buchungsjahr und Buchungstag gefragt. Am Jahresanfang, das heißt, bis März eines Jahres, fragt das Verfahren, ob eine neue Jahresrechnung beginnen soll. Wenn ja, dann werden alle Sachkonten auf 0 gestellt und der Jahresabschluß vom Vorjahr zum Jahresanfangsbestand. Man ist beim Buchen nicht an eine monatliche Verarbeitung gebunden.
2. Nach einer Sicherheitsabfrage wird die aktuelle Buchungsdatei geladen. Danach erscheint das Buchungsbild mit Buchungsjahr und Buchungstag, sowie dem Info-Window. Die fortlaufende Nummer des Buchungssatzes wird automatisch vergeben. Soweit vorhanden, erscheinen die letzten beiden Buchungen. Zunächst wird nach dem Buchungs-GRUND gefragt. Die Eingabe darf 30 Zeichen lang sein. Längere Texte werden vom Verfahren gekürzt. Wird kein Text eingegeben, sondern nur die RETURN-Taste gedrückt, so beendet das Verfahren das Buchen, sichert und geht in das Menü zurück.
3. Die Gegenkonteneingabe dürfte nicht weiter problematisch sein. Neben den o.g. Konten ist hier noch eine 0 für Umbuchungen – aber nur zwischen den Gegenkonten – vorgesehen.
Eine solche Buchung ist z.B. immer dann erforderlich, wenn Geld aus der Kasse entnommen und auf das Sparbuch eingezahlt wird.
4. Die Beträge der Einnahmen und Ausgaben sind ohne Komma einzugeben, z.B. 12345 für DM 123,45. Man gewöhnt sich sehr schnell an diese Vereinfachung beim Buchen. Es sind auch keine Leerstellen oder führende Nullen einzugeben, einfach den Pfennig-Betrag eintippen und RETURN drücken. Die Eingabe wird dann in DM umgerechnet und in die richtige Spalte gesetzt.
Bei Umbuchungen ist nur die Einnahme einzugeben. Das Programm bucht sie automatisch auch als Ausgabe. Ansonsten schließt die Eingabe einer Einnahme eine gleichzeitige Ausgabebuchung durch Überspringen dieses Feldes aus. Ein RETURN bei Einnahme setzt den Cursor auf das Ausgabefeld.
Für Korrekturen können die Beträge auch in Minus gebucht werden. Sie erhöhen dadurch nicht das Einnahme- oder Ausgabesaldo.
5. Nach Eingabe von Sachkontengruppe und Sachkonto wird abgefragt, ob die Buchung in Ordnung ist. J = RETURN-Taste , nein = eine beliebige andere Taste drücken.
6. Übersichten und Ausdruck werden über das Menü angesteuert. Alle Anzeigen lassen sich auch ausdrucken. Das Programm ist an einen Epson-Drucker angepaßt.
Alle Erklärungen sind auch in der Benutzerführung enthalten. Sie können allerdings für den möglichen Anwender vor dem Abtippen eine wertvolle Entscheidungshilfe sein.

(Eckhard Bergandy/jb)

für 664-6128



464 mit Emulator

```
10 '************************************    [1460]
20 '*         Startprogramm zu         *    [1519]
30 '*         Vereinsverwaltung        *    [972]
40 '*          - Buchhaltung -         *    [1251]
50 '*                                  *    [175]
60 '*        v. Eckhard Bergandy       *    [2145]
70 '*                 3/88             *    [250]
80 '************************************    [1460]
90 MODE 0:BORDER 22                         [637]
100 INK 0,13:PAPER 0                        [1816]
110 'Deutscher Zeichensatz"                 [1526]
120 KEY DEF 24,1,39,39,39                   [1421]
130 KEY DEF 71,1,121,89,89                  [1309]
140 KEY DEF 43,1,122,90,90                  [1502]
150 KEY DEF 39,1,44,59,59                   [1542]
160 KEY DEF 31,1,46,58,58                   [906]
170 KEY DEF 30,1,45,95,95                   [1307]
180 KEY DEF 29,1,124,92,92                  [721]
190 KEY DEF 28,1,123,91,91                  [1169]
200 KEY DEF 19,1,35,39,39                   [1230]
210 KEY DEF 17,1,43,42,42                   [1030]
220 KEY DEF 41,1,55,47,47                   [1363]
230 KEY DEF 32,1,48,61,61                   [1414]
240 KEY DEF 25,1,126,63,63                  [800]
250 KEY DEF 57,1,51,166,166                 [1355]
260 KEY DEF 26,1,125,93,93                  [614]
270 KEY DEF 22,1,60,62,62                   [1090]
280 SYMBOL AFTER 32                         [1296]
290 SYMBOL 91,&66,&3C,&66,&66,&7E,&66,&66,  [1754]
&0
300 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C,  [2304]
&0
310 SYMBOL 93,&66,&0,&66,&66,&66,&66,&3C,&  [2316]
0
320 SYMBOL 123,&66,&0,&78,&C,&7C,&CC,&76,&  [2474]
0
330 SYMBOL 124,&66,&0,&3C,&66,&66,&66,&3C,  [2094]
&0
340 SYMBOL 125,&0,&66,&0,&66,&66,&66,&3E,&  [1627]
0
350 SYMBOL 126,&38,&6C,&6C,&6C,&66,&76,&6C  [2642]
,&60
360 INK 4, 26                               [288]
370 PEN 4                                   [562]
380 INK 7,0                                 [57]
390 PLOT 110,350,7                          [591]
400 DRAW 120,340                            [638]
410 DRAW 130,330                            [496]
420 DRAW 150,300                            [614]
430 DRAW 180,220                            [79]
440 DRAW 210,300                            [427]
450 DRAW 230,330                            [434]
460 DRAW 240,340                            [469]
470 DRAW 250,350                            [499]
480 MOVE 115,350                            [664]
490 DRAW 125,340                            [371]
500 DRAW 135,330                            [640]
510 DRAW 155,300                            [442]
520 DRAW 180,225                            [74]
530 DRAW 205,300                            [402]
540 DRAW 225,330                            [476]
550 DRAW 236,340                            [187]
560 DRAW 246,350                            [350]
570 LOCATE 8,5                              [340]
580 PRINT "ereins-"                         [1143]
590 LOCATE 7,9                              [324]
600 PRINT "erwaltung"                       [1891]
610 LOCATE 5,13                             [691]
620 PRINT"BUCHHALTUNG"                      [1529]
630 PRINT TAB(9)"f}r"                       [453]
640 vn$="................":'-hier statt   [2926]
... Vereinsnamen eintragen-
650 LOCATE INT (21-LEN(vn$))/2,18           [1952]
660 PRINT CHR$(24)vn$CHR$(24)               [1493]
670 PEN 13                                  [607]
680 LOCATE 1,23                             [733]
690 PRINT CHR$(24)" eb-soft      vv 2.2 " C [3827]
HR$(24)
```

Listing "Vereins-Verwaltung"

```
700 OPENOUT "dummy":MEMORY HIMEM-1:CLOSEOU  [2303]
T
710 ' *** Maske ***                         [1364]
720 PAPER 0                                 [816]
730 MODE 2:BORDER 8:INK 0,0:INK 1,26        [2310]
740 WINDOW#1,1,80,1,5                       [1007]
750 WINDOW#2,1,80,6,8                       [1123]
760 WINDOW#3,1,80,21,21                     [1238]
770 WINDOW#4,1,80,22,25                     [1489]
780 WINDOW#5,1,80,6,20                      [1492]
790 WINDOW 2,79,23,24                       [993]
800 INK 3,9:INK 4,0:PAPER#4,3:PEN#4,4:CLS#  [951]
4
810 PRINT#4,TAB(30)">>>  I N F O S  <<<"    [1904]
820 CLS                                     [91]
830 PRINT #1,CHR$(24);"      ";CHR$(24);STR [2834]
ING$(75,208);
840 PRINT #1,CHR$(24);" ";CHR$(215);"  ";C  [3550]
HR$(214);CHR$(24);" "+vn$
850 PRINT #1,CHR$(24);" ";CHR$(213);CHR$(2  [4804]
15);CHR$(214);CHR$(212);CHR$(24)
860 PRINT #1,CHR$(24);"  ";CHR$(213);CHR$(  [4147]
212);" ";CHR$(24);" Buchhaltung"
870 PRINT #1,CHR$(24);"      ";CHR$(24);STR [3035]
ING$(75,210);
880 LOCATE #1,60,2:PRINT#1,"Buchungsjahr 1  [2699]
9  "
890 LOCATE #1,60,4:PRINT#1,"Buchungstag "   [1883]

900 CLS:CLS#5                               [232]
910 LOCATE 27,2:PRINT"Moment bitte! Ich la  [3748]
de."
920 RUN"vvbu.com"                           [1192]
```

```
10 '************************************    [1460]
20 '*         Hauptprogramm zu         *    [1395]
30 '*         Vereinsverwaltung        *    [972]
40 '*          - Buchhaltung -         *    [1251]
50 '*                                  *    [175]
60 '*        v. Eckhard Bergandy       *    [2145]
70 '*                 3/88             *    [250]
80 '************************************    [1460]
90 CLS:CLS#5:PRINT TAB(15)"Bitte die BUCHU  [4687]
NGSDISKETTE einlegen! Danach ..."
100 CLEAR                                   [217]
110 z=300                                   [449]
120 DIM btn$(z),nr(z),g$(z),gk(z),um$(z),e  [3158]
in(z),aus(z),sk(z)
130 DIM kt$(3,6),kte(3,6),kta(3,6)          [1849]
140 FOR d1=0 TO 3                           [1079]
150 FOR d2=0 TO 6                           [799]
160 READ kt$(d1,d2)                         [699]
170 NEXT d2                                 [454]
180 NEXT d1                                 [451]
190 DATA Bestand am 01.01.,Kasse,Giro,Spar  [3673]
buch,Festgeld,Bestand am,Summen
200 DATA SPORT,Allgemein,Fu~ball,Handball,  [5247]
Leichtathletik,Judo,Turnen
210 DATA VERANSTALTUNGEN,Allgemein,Basar,F  [8335]
asching,Herbstball,Stiftungsfest,Theaterab
end
220 DATA SONSTIGES,Allgemein,Beitr{ge,Geb{  [7337]
ude,Jugend,Kantine,Zusch}sse
230 ON BREAK CONT                           [323]
240 ON ERROR GOTO 2880                      [1499]
250 GOSUB 4540                              [829]
260 ' *** menue ***                         [958]
270 FOR i=0 TO 9                            [492]
280 READ men$(i)                            [306]
290 NEXT i                                  [375]
300 DATA Optionen,Buchen,Gesamtabschlu~,Sp  [9084]
ort,Veranstaltungen,Sonstiges,Buchungen -
Bild,Buchungen - Druck,Abschlu~-Druck,Prog
rammende
```

Listing "Vereins-Verwaltung"


```
310 GOSUB 2000:KEY DEF 0,1,240:KEY DEF 2,1  [1470]
,241
320 WINDOW#5,1,80,9,19                      [1373]
330 WA=0:GOSUB 1620                         [1107]
340 FOR I=1 TO 9                            [461]
350 X=INT(80-LEN(men$(i)))/2                [2037]
360 PRINT#5,TAB(x-1)" "; MEN$(I)            [779]
370 NEXT                                    [350]
380 LOCATE 23,2:PRINT"BITTE MIT CURSORTAST  [3695]
EN W[HLEN !"
390 Y1=VPOS(#5)-9:WA=1                      [1020]
400 tt=240                                  [179]
410 WHILE tt<>13                            [821]
420 X=INT(80-LEN(men$(wa)))/2-1             [2241]
430 LOCATE #5,x,Y1                          [1053]
440 PRINT#5,CHR$(24);" ";MEN$(WA);" ";CHR$  [2427]
(24)
450 ta$=INKEY$:IF ta$="" THEN 450           [819]
460 tt=ASC(ta$)                             [719]
470 IF tt=241 THEN LOCATE#5,x,y1:PRINT#5,"  [3389]
 ";men$(wa);" ":y1=y1+1:wa=wa+1
480 IF y1>9 THEN y1=1:wa=1                  [939]
490 IF tt=240 THEN LOCATE#5,x,y1:PRINT#5,"  [1734]
 ";men$(wa);" ":y1=y1-1:wa=wa-1
500 IF y1<1 THEN y1=9:wa=9                  [1752]
510 WEND                                    [390]
520 ON wa GOTO 540,1560,1470,1470,1470,321  [2783]
0,4640,4640,1880
530 CLS#5:GOSUB 1620:FOR i=1 TO 2000:NEXT:  [2803]
GOTO 1880
540 KEY DEF 0,0,32:KEY DEF 2,0,32           [2044]
550 GOSUB 1620                              [855]
560 IF q2>0 THEN 630                        [577]
570 GOSUB 2000                              [875]
580 WINDOW#5,1,80,9,19                      [1373]
590 GOSUB 2900                              [903]
600 GOSUB 2980                              [823]
610 GOSUB 1890                              [873]
620 IF qq$<>CHR$(13)THEN CLS#5:CLS:GOTO 59  [1967]
0
630 IF q4=1 THEN 680                        [950]
640 CLS                                     [91]
650 IF nbj$="J" THEN bj=VAL(bj$)-1:bj$=RIG  [3697]
HT$(STR$(bj),2)
660 GOSUB 2300                              [863]
670 IF nbj$="J" THEN GOSUB 3100:bj$=RIGHT$  [5368]
(STR$(bj+1),2):GOTO 720
680 IF q3=1 THEN 720                        [1072]
690 PRINT TAB(12)"     Ich lade jetzt die   [5453]
Daten vom ";bta$;" bis Nr.";nr(0)
700 PRINT TAB(12)"     Die letzten Daten s  [4269]
ind in der ";dat$;". Buchungsdatei."
710 GOSUB 2440                              [819]
720 GOSUB 2000                              [875]
730 dr1=2:GOSUB 3180                        [573]
740 IF nr(0)>1 THEN FOR i=b-1 TO b:dr2=5:G  [2614]
OSUB 2230:NEXT
750 GOSUB 1680                              [907]
760 nr(0)=nr(0)+1                           [537]
770 PRINT#5,"    ";USING "**#";nr(0);       [1811]
780 CLS:PRINT TAB(25)"EINGABE BEENDEN mit   [4189]
< RETURN >"
790 PRINT"   GRUND: Eingabe wird auf 30 St  [7131]
ellen begrenzt. Text mit RETURN abschlie~e
n."
800 LINE INPUT #5,;" ",gr$                  [1341]
810 IF gr$="" THEN nr(0)=nr(0)-1:CLS:PRINT  [6225]
:PRINT TAB(32)"Eingabe beendet !":GOSUB 26
00:GOSUB 2730:GOTO 310
820 IF LEN(gr$)<31 THEN 850                 [1642]
830 LOCATE #5,9,VPOS(#5):PRINT #5,SPC (LEN  [4816]
(gr$));:gr$=LEFT$(gr$,30)
840 LOCATE #5,9,VPOS(#5):PRINT#5,CHR$(7);g  [3316]
r$;
850 CLS:PRINT" GEGENKONTEN: 1 = Kasse   2   [7076]
= Giro   3 = Sparbuch   4 = Festgeld
                    0 = Umbuchung von ..
. > nach ..."
860 CURSOR#5,1                              [885]
870 LOCATE #5,42,VPOS(#5)                   [821]
880 gk$=INKEY$:IF gk$="" THEN 880           [1052]
890 IF gk$<CHR$(48) OR gk$>CHR$(52)THEN PR  [3958]
INT#5,CHR$(7);:GOTO 860
900 gk(0)=VAL(gk$)                          [1775]
910 PRINT#5,gk$;                            [328]
```

Listing "Vereins-Verwaltung"

```
920 IF gk(0)=0 THEN GOSUB 2100              [1480]
930 CLS:PRINT"      Betrag hier ohne Leer   [11486]
stellen und ohne Dezimalpunkt eingeben!
          Den Rest erledige ich dann
schon, wenn Du <RETURN> dr}ckst."
940 LOCATE #5,49,VPOS(#5)                   [908]
950 INPUT #5,;" ",ein$:ein(0)=VAL(ein$)/10  [1617]
0
960 FOR i=1 TO LEN(ein$):qq$=MID$(ein$,i,1  [1866]
)
970 IF qq$<CHR$(45) OR qq$>CHR$(57) OR qq$  [7420]
=CHR$(46) OR qq$=CHR$(47) THEN LOCATE #5,5
0,VPOS(#5):PRINT#5,CHR$(7);SPC(LEN(ein$)):
GOTO 940
980 NEXT i                                  [375]
990 LOCATE #5,50,VPOS(#5)                   [821]
1000 PRINT#5,;USING "##,###.##-";ein(0);    [2471]
1010 IF gk(0)=0 THEN aus(0)=ein(0):GOTO 10  [2030]
80
1020 IF ein(0)<>0 THEN 1080                 [690]
1030 LOCATE #5,61,VPOS(#5)                  [867]
1040 INPUT#5,;" ",aus$:aus(0)=VAL(aus$)/10  [2218]
0
1050 FOR i=1 TO LEN(aus$):qq$=MID$(aus$,i,  [3325]
1)
1060 IF qq$<CHR$(45) OR qq$>CHR$(57) OR qq  [7791]
$=CHR$(46) OR qq$=CHR$(47) THEN LOCATE #5,
62,VPOS(#5):PRINT#5,CHR$(7);SPC(LEN(aus$))
:GOTO 1030
1070 NEXT i                                 [375]
1080 LOCATE #5,62,VPOS(#5)                  [496]
1090 PRINT#5,;USING "##,###.##-";aus(0);    [2274]
1100 IF gk(0)=0 THEN 1250                   [987]
1110 CLS:PRINT:PRINT" SACHKONTENGRUPPEN:    [5877]
1 = Sport   2 = Veranstaltungen   3 = Sons
tiges"
1120 LOCATE #5,74,VPOS(#5)                  [1442]
1130 CURSOR #5,1                            [885]
1140 sg$=INKEY$:IF sg$="" THEN 1140         [1443]
1150 IF sg$<CHR$(49) OR sg$>CHR$(51)THEN P  [3444]
RINT#5,CHR$(7);:GOTO 1120
1160 sg=VAL(sg$)                            [1543]
1170 PRINT#5,sg$;                           [331]
1180 CLS:PRINT" EINZELKONTEN: 1 = Allgemei  [3469]
n   ";
1190 ON sg GOSUB 2010,2030,2050             [1394]
1200 CURSOR #5,1                            [885]
1210 SE$=INKEY$:IF se$="" THEN 1210         [1612]
1220 IF SE$<CHR$(49) OR se$>CHR$(54)THEN P  [2282]
RINT#5,CHR$(7);:GOTO 1200
1230 PRINT#5,se$;                           [404]
1240 se=VAL(se$)                            [676]
1250 sk(0)=VAL(sg$+se$)                     [1436]
1260 CURSOR#5,0:GOSUB 1890                  [1285]
1270 IF qq$<>CHR$(13) THEN nr(0)=nr(0)-1:L  [6486]
OCATE #5,5,VPOS(#5):PRINT#5,TAB(79):LOCATE
 #5,1,VPOS(#5):ein(0)=0:aus(0)=0:GOTO 760

1280 b=b+1:dat=(z+nr(0)-b)/z:dat$=STR$(dat  [3231]
)
1290 btn$(b)=btn$(0):nr(b)=nr(0):g$(b)=gr$  [6207]
:gk(b)=gk(0):um$(b)=um$(0):ein(b)=ein(0):a
us(b)=aus(0):sk(b)=sk(0)
1300 q1=1:q3=0:q4=0:gr$="":um$(0)="":ein$(  [3591]
0)="":aus$(0)="":sg$="":sk$=""
1310 GOSUB 1740                             [863]
1320 ein(0)=0:aus(0)=0                      [1130]
1330 GOSUB 1680                             [907]
1340 IF b=z THEN GOSUB 2730:b=0             [1990]
1350 PRINT#5:GOTO 760                       [590]
1360 '*************** Uebersichten          [933]
1370 GOSUB 2000                             [875]
1380 GOSUB 1620                             [855]
1390 IF q4>0 THEN 1430                      [781]
1400 GOSUB 2900                             [903]
1410 GOSUB 2300                             [863]
1420 GOSUB 2000                             [875]
1430 IF btn$(0)="" THEN bt$=bta$ ELSE bt$=  [2712]
btn$(0)
1440 WINDOW#5,1,80,6,20                     [1492]
1450 LOCATE #1,72,4:PRINT#1,bt$             [1894]
1460 RETURN                                 [555]
1470 GOSUB 1370                             [921]
1480 dr=5:ta=4:wl=wa-2                      [1387]
1490 GOSUB 4180                             [837]
```

Listing "Vereins-Verwaltung"

```
1500 GOSUB 2070                                    [869]
1510 GOSUB 1910                                    [893]
1520 IF UPPER$(qq$)="D" THEN dr=8:IF wa=2          [2378]
THEN 1580 ELSE 1490
1530 qq=VAL(qq$):IF qq<2 OR qq>9 THEN PRIN         [2840]
T CHR$(7):GOTO 1500
1540 IF qq=9 THEN 310                              [677]
1550 wa=qq:GOTO 520                                [2193]
1560 GOSUB 1370                                    [921]
1570 dr=5:ta=4                                     [261]
1580 GOSUB 4280                                    [867]
1590 GOSUB 4330                                    [1063]
1600 GOSUB 4370                                    [1055]
1610 GOTO 1500                                     [399]
1620 LOCATE #1,20,3 :'******************           [2511]
**** ]berschrift
1630 PRINT#1,SPC(35)                               [459]
1640 x=INT(80-LEN(men$(wa)))/2                     [2059]
1650 LOCATE #1,x-1,3                               [585]
1660 PRINT#1,CHR$(24);" ";UPPER$(men$(wa))         [3357]
;" ";CHR$(24)
1670 RETURN                                        [555]
1680 PRINT#3: '************************            [5116]
**** Gegenkonten anzeigen
1690 FOR i=1 TO 4                                  [450]
1700 PRINT#3,"  ";kt$(0,i);":"; USING"##,#          [1656]
##.##-";KTE(0,i);
1710 NEXT i                                        [375]
1720 RETURN                                        [555]
1730 '**********************************           [1871]
**** Berechnungen
1740 IF gk(b)=0 THEN kte(0,u1)=kte(0,u1)-e         [2410]
in(b):kte(0,u2)=kte(0,u2)+ein(b):GOTO 1870

1750 kte(0,gk(b))=kte(0,gk(b))+ein(b)-aus(         [2497]
b)
1760 kte(sg,se)=kte(sg,se)+ein(b)                  [1509]
1770 kta(sg,se)=kta(sg,se)+aus(b)                  [1350]
1780 kte(sg,0)=kte(sg,0)+ein(b)                    [2057]
1790 kta(sg,0)=kta(sg,0)+aus(b)                    [1809]
1800 kte(0,6)=kte(1,0)+kte(2,0)+kte(3,0)           [2708]
1810 kta(0,6)=kta(1,0)+kta(2,0)+kta(3,0)           [1601]
1820 FOR i=1 TO 3                                  [435]
1830 kta(0,i)=kte(i,0)-kta(i,0)                    [1950]
1840 NEXT i                                        [375]
1850 kta(0,0)=kte(0,6)-kta(0,6)                    [1724]
1860 kte(0,5)=kte(0,0)+kta(0,0)                    [1280]
1870 RETURN                                        [555]
1880 WINDOW 1,80,1,25:MODE 2:END                   [1688]
1890 CLS:LOCATE 26,2: '******************          [1734]
 Abfrage
1900 PRINT" Eingabe richtig ?("+CHR$(242)+         [2083]
CHR$(153)+"/N) "
1910 CLEAR INPUT                                   [184]
1920 QQ$=INKEY$:qq$=UPPER$(qq$)                    [2021]
1930 IF qq$="" THEN 1920                           [776]
1940 RETURN                                        [555]
1950 nbj$="" :' ************************           [3246]
***** Neues Buchungsjahr?
1960 LOCATE#5,14,10:PRINT#5,"Soll eine NEU         [6191]
E Jahresrechnung begonnen werden? (J/N)";

1970 nbj$=INKEY$:IF nbj$="" THEN 1970              [1268]
1980 nbj$=UPPER$(nbj$):IF nbj$="J" OR nbj$         [3901]
="N" THEN PRINT#5,nbj$:RETURN
1990 PRINT CHR$(7):GOTO 1970                       [1894]
2000 CLS#2:CLS#3:CLS#5:CLS:RETURN :' *****         [3472]
**** Windows l¦schen
2010 PRINT"      2 = Fu~ball    3 = Handba         [5365]
ll"' ********************** Info Sachkon
ten
2020 PRINT TAB(16)"4 = Leichtathletik    5         [4732]
 = Judo      6 = Turnen":RETURN
2030 PRINT" 2 = Basar          3 = Faschin         [2124]
g"
2040 PRINT TAB(16)"4 = Herbstball   5 = St         [5373]
ifungsfest   6 = Theaterabend":RETURN
2050 PRINT"2 = Beitr{ge   3 = Geb{ude   4          [2755]
= Jugend"
2060 PRINT TAB(16)"5 = Kantine     6 = Zus         [3899]
ch}sse": RETURN
2070 PRINT"Auswahl: 2 = Abschlu~   3 = Spo         [10916]
rt   4 = Veranstaltungen  5 = Sonstiges":'
 ************************ Wahl Uebersic
hten
```

Listing "Vereins-Verwaltung"

```
2080 PRINT"         6 = Buchungen-Bild   7         [7161]
 = Buchungen-Druck   9 = Optionen   D = Dr
uck"
2090 RETURN                                        [555]
2100 CURSOR#5,1:'************************          [3347]
******** Umbuchung
2110 LOCATE #5,44,VPOS(#5)                         [852]
2120 u1$=INKEY$:IF u1$="" THEN 2120                [1394]
2130 IF u1$<CHR$(49) OR u1$>CHR$(52)THEN P         [2583]
RINT#5,CHR$(7);:GOTO 2100
2140 PRINT#5,u1$+CHR$(62);:u1=VAL(u1$)             [1463]
2150 CURSOR#5,1                                    [885]
2160 LOCATE #5,46,VPOS(#5)                         [1370]
2170 u2$=INKEY$:IF u2$="" THEN 2170                [1531]
2180 IF u2$<CHR$(49) OR u2$>CHR$(52) OR u2         [3963]
$=u1$ THEN PRINT#5,CHR$(7);:GOTO 2150
2190 PRINT#5,u2$;:u2=VAL(u2$)                      [865]
2200 um$(0)=u1$+CHR$(62)+u2$                       [1285]
2210 RETURN                                        [555]
2220 '**********************************           [2224]
* Buchungen anzeigen
2230 PRINT#dr2,"    ";USING"###";nr(i);            [2446]
2240 PRINT#dr2," ";g$(i);TAB(41)gk(i);um$(         [2382]
i);
2250 PRINT#dr2,TAB(50)USING"##,###.##-";ei         [2916]
n(i);
2260 PRINT#dr2,TAB(62)USING"##,###.##-";au         [2978]
s(i);
2270 PRINT#dr2,TAB(73)sk(i)                        [2091]
2280 RETURN                                        [555]
2290 ' **********************************          [2488]
** LADEN  Grunddatei
2300 OPENIN "vv"+bj$+".su"                         [2017]
2310 INPUT #9,bta$                                 [975]
2320 INPUT #9,nr(0)                                [1011]
2330 INPUT #9,dat$                                 [1060]
2340 FOR d1=0 TO 3                                 [1079]
2350 FOR d2=0 TO 6                                 [799]
2360 INPUT #9,kte(d1,d2)                           [1278]
2370 INPUT #9,kta(d1,d2)                           [1024]
2380 NEXT d2                                       [454]
2390 NEXT d1                                       [451]
2400 CLOSEIN                                       [752]
2410 q4=1                                          [327]
2420 RETURN                                        [555]
2430 ' **********************************          [3189]
** LADEN Buchungsdatei
2440 '                                             [117]
2450 OPENIN "vvbu"+bj$+"."+dat$                    [2216]
2460 INPUT #9,b                                    [694]
2470 FOR i=1 TO b                                  [420]
2480 INPUT #9,btn$(i)                              [1425]
2490 INPUT #9,nr(i)                                [1125]
2500 INPUT #9,g$(i)                                [711]
2510 INPUT #9,gk(i)                                [883]
2520 INPUT #9,um$(i)                               [944]
2530 INPUT #9,ein(i)                               [1150]
2540 INPUT #9,aus(i)                               [1174]
2550 INPUT #9,sk(i)                                [948]
2560 NEXT                                          [350]
2570 CLOSEIN:q3=1                                  [731]
2580 RETURN                                        [555]
2590 ' **********************************          [1794]
** SICHERN Grunddatei
2600 IF q1=0 THEN 310 ELSE OPENOUT "vv"+bj         [4127]
$+".su"
2610 WRITE #9,btn$(b)                              [1207]
2620 WRITE #9,nr(0)                                [997]
2630 WRITE #9,dat$                                 [1242]
2640 FOR d1=0 TO 3                                 [1079]
2650 FOR d2=0 TO 6                                 [799]
2660 WRITE #9,kte(d1,d2)                           [1486]
2670 WRITE #9,kta(d1,d2)                           [1232]
2680 NEXT d2                                       [454]
2690 NEXT d1                                       [451]
2700 CLOSEOUT                                      [902]
2710 RETURN                                        [555]
2720 ' **********************************          [2272]
** SICHERN Buchungsdatei
2730 OPENOUT "vvbu"+bj$+"."+dat$                   [1884]
2740 WRITE #9,b                                    [1011]
2750 FOR i=1 TO b                                  [420]
2760 WRITE #9,btn$(i)                              [1150]
2770 WRITE #9,nr(i)                                [984]
2780 WRITE #9,g$(i)                                [1203]
```

Listing "Vereins-Verwaltung"


```
2790 WRITE #9,gk(i)                                [1490]
2800 WRITE #9,um$(i)                               [1111]
2810 WRITE #9,ein(i)                               [1393]
2820 WRITE #9,aus(i)                               [1316]
2830 WRITE #9,sk(i)                                [953]
2840 NEXT                                          [350]
2850 q1=0                                          [248]
2860 CLOSEOUT                                      [902]
2870 RETURN                                        [555]
2880 CLS:PRINT TAB(12),"Hmmm...! Da stimmt         [5274]
 was nicht! Ich mache gleich weiter.":RESU
ME NEXT
2890 '************************ Abfrage Buc         [1380]
hungsjahr
2900 LOCATE #5,46,2                                [590]
2910 PRINT#5,"          "                          [949]
2920 LOCATE #5,30,2                                [558]
2930 INPUT #5,;"Buchungsjahr: 19",bj$              [1880]
2940 IF LEN(bj$)<>2 THEN PRINT CHR$(7):GOT         [1622]
O 2900
2950 LOCATE #1,75,2:PRINT#1,bj$                    [1832]
2960 RETURN                                        [555]
2970 '************************ Abfrage Buc         [1209]
hungstag
2980 LOCATE #5,48,5                                [591]
2990 PRINT#5,"                "                    [1291]
3000 LOCATE #5,25,5                                [667]
3010 INPUT #5,;"Buchungstag (TT.MM.JJ):",b         [3183]
t$
3020 IF LEN(bt$)<>8 OR MID$(bt$,3,1)<>"."          [9331]
OR MID$(bt$,6,1)<>"."OR VAL(LEFT$(BT$,2))>
31 OR VAL(MID$(BT$,4,2))>12 THEN PRINT CHR
$(7):GOTO 2980
3030 LOCATE #1,72,4                                [748]
3040 PRINT#1,bt$                                   [349]
3050 IF VAL(MID$(bt$,4,2))<3 AND RIGHT$(bt         [4474]
$,2)=bj$ THEN GOSUB 1950
3060 q2=1                                          [303]
3070 btn$(0)=bt$                                   [822]
3080 RETURN                                        [555]
3090 ' ************************ Arrays auf          [2550]
0 setzen f}r Jahresanfang
3100 kte(0,0)=kte(0,5):kte(0,6)=0                  [2109]
3110 FOR d1=1 TO 3                                 [1311]
3120 FOR d2=0 TO 6                                 [799]
3130 kte(d1,d2)=0:kta(d1,d2)=0:kta(0,d2)=0         [2046]
3140 NEXT d2                                       [454]
3150 NEXT d1                                       [451]
3160 nr(0)=0:b=0                                   [346]
3170 RETURN                                        [555]
3180 PRINT#dr1:PRINT #dr1,"    Nr. Grund           [6241]
                    GK UMB    Einnahm
e    Ausgabe    SK"
3190 PRINT #dr1,"    --- ------------------        [4156]
-------------  -- ---   ---------   ------
---   --"
3200 RETURN                                        [555]
3210 wa=6:GOSUB 2000:ve$="angezeigt":dr1=2         [3289]
:dr2=5
3220 WINDOW#5,1,80,9,21                            [1391]
3230 GOSUB 1620                                    [855]
3240 IF q4>0 THEN 3270                             [781]
3250 GOSUB 2900                                    [903]
3260 GOSUB 2300                                    [863]
3270 PRINT TAB(16)"Im Buchungsjahr 19";bj$         [4280]
;" wurde bis Nr.";nr(0);"gebucht."
3280 GOSUB 4540                                    [829]
3290 CLS#5:CLS#2                                   [871]
3300 PRINT#5,TAB(5)"Welche Buchungen solle         [5060]
n ";ve$;" werden?"
3310 PRINT#5:PRINT#5,TAB(30)"alle"                 [2305]
3320 PRINT#5,TAB(30)"alle von... bis..."           [1216]
3330 PRINT#5,TAB(30)"alle eines Sachkontos         [1922]
"
3340 PRINT#5,TAB(30)"alle eines Gegenkonto         [1642]
s"
3350 PRINT#5,TAB(30)"alle Umbuchungen"             [1778]
3360 PRINT#5:PRINT#5,TAB(30)"OPTIONEN"             [1901]
3370 PRINT#5,TAB(30)"PROGRAMMENDE"                 [1368]
3380 CLS:LOCATE 23,2:PRINT"BITTE MIT CURSO         [4084]
RTASTEN W{HLEN !"
3390 Y1=VPOS(#5)-8:WA=1                            [988]
3400 tt=240                                        [179]
3410 WHILE tt<>13                                  [821]
3420 LOCATE #5,28,y1                               [364]
3430 PRINT#5,CHR$(243)                             [1010]
3440 ta$=INKEY$:IF ta$="" THEN 3440                [1018]
3450 tt=ASC(ta$)                                   [719]
```

Listing "Vereins-Verwaltung"

```
3460 IF tt=241 THEN LOCATE #5,28,y1:PRINT#         [5092]
5," ":y1=y1+1:wa=wa+1:IF wa=6 THEN y1=y1+1
3470 IF y1>10 THEN y1=3:wa=1                       [1165]
3480 IF tt=240 THEN LOCATE #5,28,y1:PRINT#         [3388]
5," ":y1=y1-1:wa=wa-1:IF wa=5 THEN y1=y1-1

3490 IF y1<3 THEN y1=10:wa=7                       [1915]
3500 WEND:GOSUB 2000:j=0:btn$(0)=""                [2123]
3510 ON wa GOTO 3530,3780,3880,4040,3530,3         [2496]
10,1880
3520 ' *************** Gesamt}bersicht             [462]
3530 GOSUB 2000:GOSUB 3180                         [1265]
3540 da=VAL(dat$):z1=0:                            [1456]
3550 FOR dat=1 TO da:dat$=STR$(dat)                [1479]
3560 IF dat=1 AND q3>0 THEN 3590                   [775]
3570 GOSUB 4510                                    [835]
3580 GOSUB 2440                                    [819]
3590 i=0                                           [420]
3600 GOSUB 4620                                    [1069]
3610 WHILE INKEY(47)<>0 AND i<b                    [2389]
3620 i=i+1                                         [444]
3630 ON wa GOSUB 4630,3850,4020,4140,4160          [2343]
3640 IF j=1 AND btn$(i)<>btn$(i-1) THEN GO         [3773]
SUB 4490:z1=z1+1:PRINT CHR$(7)
3650 IF j=1 THEN GOSUB 2220:z1=z1+1                [2072]
3660 IF z1=60 AND dr2=8 THEN GOSUB 4870            [2580]
3670 WEND:IF i=b THEN 3740                         [1999]
3680 GOSUB 4570                                    [839]
3690 GOSUB 1910                                    [893]
3700 IF qq$="W" THEN GOSUB 4620:GOTO 3610          [747]

3710 IF qq$="N" THEN 3740                          [458]
3720 IF qq$="S" THEN dat=da:dat$=STR$(dat)         [1614]
:GOTO 3290
3730 GOTO 3690                                     [670]
3740 NEXT dat                                      [547]
3750 PRINT#5,CHR$(13);TAB(5)CHR$(24)"ENDE          [4291]
der Datei"CHR$(24):PRINT#5
3760 IF z1=0 AND b1=0 THEN GOSUB 4590 ELSE         [2778]
 GOSUB 4530
3770 GOTO 3290                                     [686]
3780 ' ********************** VON-BIS              [750]
3790 LOCATE #5,30,1:INPUT #5,"von Nummer:"         [1619]
,vo
3795 IF vo=VAL(CHR$(13)) THEN GOTO 3290            [2215]
3800 IF vo<1 OR vo>nr(0)-1 THEN PRINT#5,CH         [2544]
R$(7);CHR$(19):GOTO 3790
3810 PRINT#5,CHR$(20):LOCATE #5,30,3:INPUT         [4293]
 #5,"bis Nummer:",bi
3820 IF bi<vo OR bi>nr(0) THEN PRINT#5, CH         [3403]
R$(7);CHR$(11);CHR$(11):GOTO 3810
3830 CLS#5                                         [385]
3840 GOTO 3530                                     [503]
3850 IF nr(i)>=vo AND nr(i)<=bi THEN j=1 E         [3124]
LSE j=0
3860 IF nr(i)=bi THEN dat=da:dat$=STR$(dat         [2415]
)
3870 RETURN                                        [555]
3880 GOSUB 2000                                    [875]
3890 PRINT#5,TAB(30)CHR$(24);" S a c h k o         [3388]
 n t e n ";CHR$(24)
3900 PRINT#5                                       [314]
3910 PRINT#5,TAB(6)kt$(1,0)TAB(31)kt$(2,0)         [1965]
TAB(56)kt$(3,0)
3920 PRINT#5                                       [314]
3930 FOR d2=1 TO 6                                 [1021]
3940 PRINT#5,TAB(5)10+d2;kt$(1,d2)TAB(30)2         [3281]
0+d2;kt$(2,d2)TAB(55)30+d2;kt$(3,d2)
3950 NEXT d2                                       [454]
3960 PRINT#3,SPC(18):INPUT #3,"Welches Sac         [4590]
hkonto soll angezeigt werden";qq$
3965 IF qq$="" THEN 3290                           [734]
3970 qq=VAL(LEFT$(qq$,1)):IF qq<1 OR qq>3          [2885]
THEN CLS#3:GOTO 3960
3980 qq=VAL(RIGHT$(qq$,1)):IF qq<1 OR qq>6         [3492]
 THEN CLS#3:GOTO 3960
3990 IF LEN(qq$)<>2 THEN CLS#3:GOTO 3960           [3110]
4000 qq=VAL(qq$)                                   [531]
4010 GOTO 3530                                     [503]
4020 IF sk(i)=qq THEN j=1 ELSE j=0                 [1138]
4030 RETURN                                        [555]
4040 GOSUB 2000                                    [875]
4050 PRINT#5,TAB(30)CHR$(24)" G e g e n k          [2446]
o n t e n "CHR$(24)
4060 PRINT#5                                       [314]
4070 FOR d1=1 TO 4                                 [1302]
4080 PRINT#5,TAB(30)d1;kt$(0,d1)                   [1333]
4090 NEXT d1                                       [451]
```

Listing "Vereins-Verwaltung"

```
4100 PRINT#3,SPC(18):INPUT#3,"Welches Gege [3725]
nkonto soll angezeigt werden";qq$
4105 IF qq$="" THEN 3290                   [734]
4110 qq=VAL(qq$)                           [531]
4120 IF qq<1 OR qq>4 THEN CLS#3:GOTO 4100  [1807]
4130 GOTO 3530                             [503]
4140 IF gk(i)=qq THEN j=1 ELSE j=0         [2215]
4150 RETURN                                [555]
4160 IF gk(i)=0 THEN j=1 ELSE j=0          [535]
4170 RETURN                                [555]
4180 PRINT#dr:PRINT#dr,TAB(ta)kt$(wl,0);TA [6420]
B(27)" Einnahmen","Ausgaben","Salden"
4190 PRINT#dr,TAB(ta)STRING$(56,&2D);      [2997]
4200 PRINT#dr                              [836]
4210 FOR i=1 TO 6                          [448]
4220 PRINT#dr,TAB(ta)kt$(wl,i);TAB(27)USIN [5937]
G"###,###.##-";kte(wl,i),kta(wl,i),kte(wl,
i)-kta(wl,i)
4230 NEXT i                                [375]
4240 PRINT#dr,TAB(ta)"-------------------- [2850]
   ---------- ---------- ----------"
4250 PRINT#dr,TAB(ta)kt$(0,6),,USING"###,# [5004]
##.##-";kte(wl,0),kta(wl,0),kte(wl,0)-kta(
wl,0)
4260 PRINT#dr,TAB(ta)"-------------------- [2850]
   ---------- ---------- ----------"
4270 RETURN                                [555]
4280 PRINT#dr:PRINT#dr,TAB(ta)"Abschluˆ "; [2160]
4290 PRINT#dr,TAB(27)" Einnahmen","Ausgabe [4136]
n","Salden"
4300 PRINT#dr,TAB(ta)STRING$(73,&2D)       [2771]
4310 PRINT#dr,TAB(ta)kt$(0,0)+bj$;TAB(66)U [4420]
SING"###,###.##-";kte(0,0):PRINT#dr
4320 RETURN                                [555]
4330 FOR i=1 TO 3                          [435]
4340 PRINT#dr,TAB(ta)kt$(i,0);TAB(27)USING [4474]
"###,###.##-";kte(i,0),kta(i,0),kta(0,i)
4350 NEXT i                                [375]
4360 RETURN                                [555]
4370 PRINT#dr,TAB(ta)"-------------------- [2850]
   ---------- ---------- ----------"
4380 PRINT#dr,TAB(ta)"Gesamtsummen",USING" [5211]
###,###.##-";kte(0,6),kta(0,6),kta(0,0);

4390 PRINT#dr,TAB(66)USING"###,###.##-";kt [2114]
a(0,0)
4400 PRINT#dr,TAB(ta)"================== [5272]
====================================";TAB(
66)"----------"
4410 PRINT#dr,TAB(ta)kt$(0,5);" ";bt$      [2051]
4420 PRINT#dr,"         Kasse          Gi [4008]
ro       Sparbuch       Festgeld"
4430 FOR i=1 TO 4                          [450]
4440 PRINT#dr,USING"#######,###.##-";kte(0 [1849]
,i);
4450 NEXT i                                [375]
```

Listing "Vereins-Verwaltung"

```
4460 PRINT#dr,TAB(66)USING"###,###.##-";kt [2014]
e(0,5)
4470 PRINT#dr,TAB(ta)STRING$(73,&3D)       [2735]
4480 RETURN                                [555]
4490 IF dr2=5 THEN PRINT#5,CHR$(13);TAB(5) [4731]
CHR$(24)btn$(i)CHR$(24):RETURN
4500 PRINT#dr2,CHR$(13);TAB(5)btn$(i):RETU [2581]
RN
4510 CLS:LOCATE 27,2:PRINT"Moment, bitte!  [2715]
Ich lade."
4520 RETURN                                [555]
4530 CLS                                   [91]
4540 LOCATE 27,2:PRINT"W E I T E R  mit Ta [2780]
ste !"
4550 CALL &BB18                            [389]
4560 RETURN                                [555]
4570 CLS:PRINT:PRINT TAB(20)"(w)eiter      [3382]
 (s)top     (n){chste Datei"
4580 RETURN                                [555]
4590 CLS                                   [91]
4600 PRINT TAB(20)"Gesuchte Daten sind nic [4609]
ht in der Datei."
4610 GOTO 4540                             [317]
4620 CLS:PRINT:PRINT TAB(22)"Unterbrechen  [5063]
mit der < Leertaste >"
4630 j=1:RETURN                            [1077]
4640 '********************** Drucken von   [2009]
Buchungen und Abschluˆ
4650 GOSUB 2000                            [875]
4660 WINDOW#5,1,80,9,19                    [1373]
4670 GOSUB 1620                            [855]
4680 dr1=8:dr2=8:dr=8:ta=4                 [480]
4690 ve$="gedruckt"                        [1440]
4700 PRINT#5,TAB(30)"Ist der Drucker berei [2752]
t?"
4710 GOSUB 4530                            [847]
4720 CALL &BB18                            [389]
4730 GOSUB 2000                            [875]
4740 IF wa=7 THEN 3230                     [976]
4750 GOSUB 1370                            [921]
4760 PRINT#8,TAB(25)"- D e r   K a s s e n [1923]
 w a r t -"
4770 PRINT#8:PRINT#8:PRINT#8:PRINT#8       [1873]
4780 PRINT#8,TAB(29)"Abschluˆ vom "+bt$    [2179]
4790 PRINT#8:PRINT#8                       [1314]
4800 GOSUB 4300                            [1069]
4810 wl=1:GOSUB 4180                       [1617]
4820 wl=2:GOSUB 4180                       [1098]
4830 wl=3:GOSUB 4180                       [1573]
4840 GOSUB 4380                            [1053]
4850 PRINT#8,CHR$(12)                      [1074]
4860 GOTO 310                              [510]
4870 z1=0                                  [640]
4880 PRINT#8,CHR$(12)                      [1074]
4890 GOSUB 3180                            [1148]
4900 RETURN                                [555]
```

Listing "Vereins-Verwaltung"

DMV informiert

Wir möchten sie darauf hinweisen, daß sich unsere Telefonummer geändert hat!

Die Telefonzentrale erreichen Sie unter:.....**0 56 51/80 09-0**

Telefax:.....**0 56 51/80 09-33**

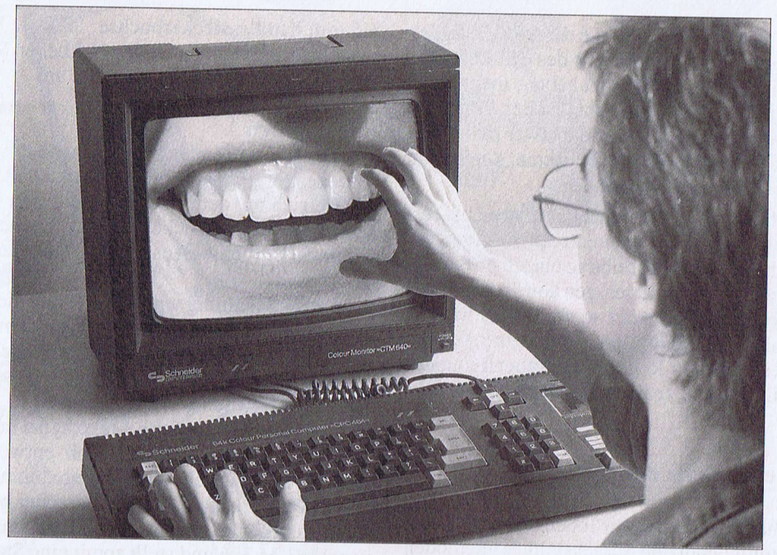
Telex:.....**9 93 210 dmv d**

Für unseren HOTLINE-Service, wo Ihnen unsere Redakteure zu Ihren aktuellen Computerproblemen Rat und Hilfen geben, haben wir spezielle "heiße Drähte" installieren lassen.

Die PC-Amstrad-Redakteure sind für Sie jeden Mittwoch von 17.00 – 20.00 unter folgenden Rufnummern erreichbar:

Herr Daschner (CPC).. 0 56 51/80 09-16
Herr Borngießer (CPC/Hardware).. 0 56 51/80 09-17
Herr Schößler (PCW).. 0 56 51/80 09-18
Herr Ebbrecht (PCW/PC)..... 0 56 51/80 09-18

DMV Verlag · Daten & Medien Verlagsges. mbH · Postfach 2 50 · Fuldaer Straße 6 · 3440 Eschwege



Der Laberkasten

Sprach-Synthesizer für die CPC's

"Guten Morgen! Heute ist Mittwoch, der 11.9.; draußen sind 10 Grad Celsius; es ist kurz vor Sieben... Guten Morgen! Heute ist Mittwoch,...". Mit freundlicher Stimme wiederholt ein Computer den Text zum Aufwecken - auf die Mitteilung "Du hast nur noch 20 Minuten Zeit! Beeil Dich!" steht der übernächtigte Computerfreak auf und stürzt sich in den neuen Tag. Science Fiction meinen Sie? Nein, Sprechi, der Sprach-Synthesizer.

Bevor wir jedoch zum konkreten Aufbau und Test von SPRECHI kommen, möchten wir Sie mit den elementaren Grundlagen der Sprachsynthese vertraut machen:
Die intensive Beschäftigung, wie und wo die Sprache entsteht, ist aber auch den Entwicklern heutiger Sprachchips zugute gekommen, auch wenn sie die Sprache total anders ablegen und wiedergeben: Die Speicherung geschieht nämlich digital, also in Nullen und Einsen, und nicht analog wie auf Schallplatte oder Kassette.
Da die Sprache selbst jedoch analog und nicht digital ist, denn sie hat ja große Dynamikschwankungen, feine Nuancen, etc. und nicht nur "an" und "aus", muß die analoge Information zur Speicherung irgendwie ins digitale System übersetzt werden, denn die Sprache kann nur auf digitaler Basis (zumindest halbwegs) verständlich reproduziert werden. Diese Digitalisierung der Sprache bringt, besonders wegen der großen Dynamikschwankungen und schnellen Übergängen zum Teil erhebliche Probleme mit sich, weshalb man spezielle Methoden zur Sprachdigitalisierung für Sprachsynthesizer anwendet:

1. Die "Wellenform-Codierung":

Bei dieser Methode wird die Amplitude des Sprachsignals in bestimmten Zeitabständen abgetastet und der zu diesem Zeitpunkt vorliegende Wert gespeichert. Logischerweise wird die Genauigkeit, mit der das ursprüngliche, analoge Sprachsignal wiedergegeben wird, mit steigender Abtastfrequenz ebenfalls größer, weil einfach mehr Werte vorliegen. Aber auch die Datenmenge nimmt (sogar extrem) zu, weshalb viel Speicherplatz nötig ist. Wenn der Synthesizer also immer dasselbe sagen soll, z.B. "Sie haben Ihr Licht noch an", "Der Tank ist bald leer", ist diese Methode gut geeignet (wenn genug Speicher vorhanden ist), denn die Sprache wird quasi fotografiert. Fast zwangsläufig denkt man da an die CD, die die Musik ja auch "abfotografiert". Tatsächlich wird die Musikinformation mit der Wellenform-Codierung gespeichert, wobei das Signal 44100mal pro Sekunde abgetastet wird und so auf jeder CD ca. 16 GigaBit gespeichert sind...

2. Die "Formantenfrequenz- Analyse":

Diese Methode mit dem langen Namen wird für die heutzutage erhältlichen Sprachchips benutzt: sie beruht auf der genauen Kenntnis der Grundelemente der Sprache, die PHONEME genannt werden. Ein Phonem ist also die kleinste Einheit, in die die Sprache zerlegt werden kann. Diese Phoneme werden nun einzeln synthetisiert, also schließlich digital abgespeichert, und können dann in beliebiger Reihenfolge aneinandergereiht werden. So können neue Worte einfach programmiert werden, auch wenn sie nie synthetisiert wurden, denn sie entstehen nur noch durch das Aneinanderreihen der Phoneme. Deshalb kann SPRECHI auch jeden beliebigen Text deutsch, englisch, oder sonstwie sprechen, wenn ein entsprechendes Treiberprogramm den eingegebenen Text in die entsprechenden Phoneme umwandelt und an SPRECHI weitergibt.

Zur Programmierung von SPRECHI (ohne die RSX-Befehle) ist also die Kenntnis der Phonemnummern des SSI 263 (der Sprachchip, den wir in SPRECHI benutzt haben; mehr dazu später) unbedingt nötig, weshalb eine Tabelle aller Phoneme mit Beispielworten natürlich nicht fehlen darf (wer mal einen Blick riskieren will: in Tabelle 1 sind alle Phoneme, die SPRECHI zur Verfügung stehen, aufgelistet).

Der eigentlich unendlich große Wortschatz wird aber bei der Phonem-Methode (bzw. bei der Formantenfrequenz-Analyse) dadurch erkauft, daß bei Aneinanderreihung einzelner Phoneme eine Stimme entsteht, die sehr derjenigen ähnelt, die allgemein als "Roboterstimme" bezeichnet wird. Dieses Manko wird beim SSI 263 (also dem Chip, den wir benutzen) durch die vom Benutzer frei setzbaren Register für Filterfrequenz, Tonhöhe, Geschwindigkeit, etc. (nähere Erläuterung s.u.) fast behoben.

3. Mathematische Modelle der Sprache:

Der Vollständigkeit halber möchten wir Ihnen noch die wahrscheinlich in der Zukunft benutzten Sprachsynthesemethoden kurz nennen, die letztlich bessere Ergebnisse als die Phonemmethode liefern werden: die "Linear Predictive Coding" (LPC) Methode und die schon etablierte "Fourieranalyse". Beide Methoden benutzen zur Beschreibung der Sprache komplizierte mathematische Formeln, die auch "Schuld daran" sind, daß diese Technik nicht schon heute verfügbar ist: die heutigen Chips rechnen einfach immer noch nicht schnell genug!

Nach diesen allgemeinen Grundlagen der Sprachsynthese kommen wir nun zur Praxis, nämlich zum Aufbau, Test und schließlich der Programmierung von SPRECHI. Doch beginnen wir mit dem Aufbau des Sprachsynthesizers…

Der Aufbau von SPRECHI

Der Hardwareaufbau besteht aus drei Gruppen:

1. PIO (Ein- und Ausgabe-Schnittstelle)
2. Sprachchip
3. Verstärker

An Bauteilen benötigen Sie Folgendes:
Widerstände:
R1a,b = 4.7 K

R1a ist in den Bestückungsplan mit eingebaut, R1b sollte eingebaut werden, falls der Synthesizer nicht richtig funktioniert.
R2 = 3k9 (3.9 Kiloohm)
R3 = 10 Ohm
Poti TR1 = 25 kOhm/linear
Kondensatoren:
C1 = 100uF/16V
C1 wird parallel zur externen Spannungsversorgung geschaltet.
C2 = 0.22uF
C3 = 220 uF/16V
C4 = 0.068uF

(Die Bezeichnung uF steht für Mikrofarad)
Halbleiter:
IC 1 = Z80A PIO
IC 2 = SN 74LS32
IC 3 = SSI 263
IC 4 = SN 74LS74
IC 5 = LM 386
IC- Sockel
Expansionport-Stecker (bei 464 und 664 Platinenstecker, beim 6128 CENTRONICS-Adapter), dünner isolierter Draht.
3.5 mm Klinkensteckerbuchse, falls Sie Sprechi an Ihre Stereoanlage anschließen wollen, schließen Sie die Buchse direkt an den Ausgang des Synthesizers an.
8 Ohm-Lautsprecher 0.5 W
Lötstifte 1mm

Das Ein-/Ausgabeinterface besteht hauptsächlich aus der mittlerweile sehr günstig zu erhaltenden Z80A PIO, die speziell für den Betrieb mit dem Z80 Prozessor (der ja auch im CPC seinen Dienst tut) entwickelt wurde. Sie bietet 16 frei programmierbare Ein- bzw. Ausgänge, die in die Ports A und B aufgeteilt sind.
In unserer Schaltung wurde Port A an die Datenleitungen des Sprachchips gelegt, Port B Bit 7 mit dem A/R Eingang, Bit 6 mit CS0 und die Bits 3 bis 5 mit den Anschlüssen RS0 bis RS2 verbunden. Die Bits 0,1 und 2 sind frei, so daß an sie noch Geräte angeschlossen werden können.
Das vom Chip kommende Audiosignal (also die Sprache bzw. Musik oder Musikeffekte) kann entweder direkt über die 3.5mm Klinkensteckerbuchse abgenommen oder über den internen Verstärker auf den Lautsprecher gegeben werden. Dieser Verstärker besteht aus einer Beschaltung des allgemein bekannten LM 386 und stellt somit eine Standardschaltung dar.
Kommen wir nun zum Erstellen der Platine:
Nach dem Fotopositiv-Verfahren wird auf übliche Weise das Layout auf die Platine gebracht. (Sie können vom Layout eine Folie herstellen lassen, und diese zum Belichten der Platine benutzen, erfahrene Hobby-Elektroniker können diese Schaltung jedoch auch auf einer Lochraster-Platine aufbauen.)
Wenn nach dem Ätzen die Leiterbahnen fehlerfrei übriggeblieben sind und die Platine gereinigt worden ist, werden an allen Lötaugen Bohrungen mit einem 0.8mm-Bohrer vorgenommen.
Damit sind die Arbeiten an der Platine beendet und wir können selbige nun bestücken:

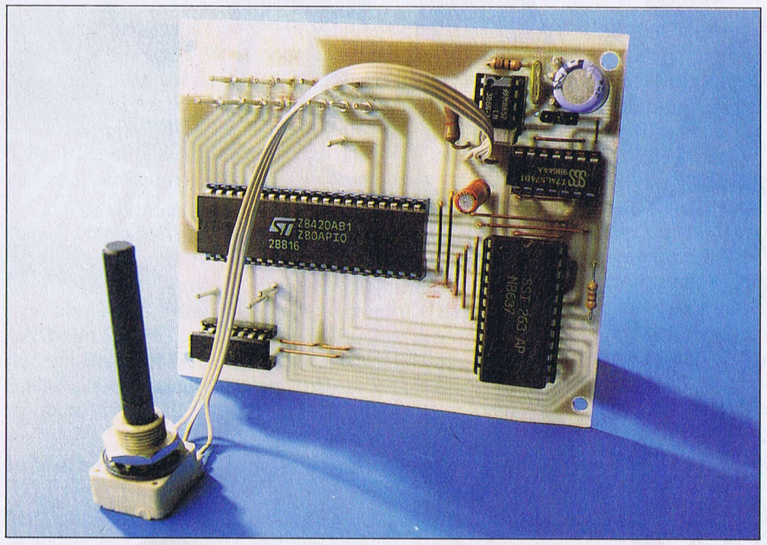
Das Einsetzen der Bauteile

Zuerst setzen Sie die Brücken, die Widerstände und Kondensatoren ein (in dieser Reihenfolge), die Brücken zuerst, da einige unter den ICs liegen. Danach löten Sie die Sockel der ICs ein und überprüfen die Schaltung auf fehlerhafte Lötstellen. Nun bringen Sie das Poti und den Lautsprecher mit flexiblem Draht an die Lötnägel.
Haben Sie diese "Tortur" hinter sich, können Sie zum Schluß die ICs in ihre Fassungen setzen, wobei Sie darauf achten müssen, die beiden LS-Typen nicht zu vertauschen!
Jetzt sollten Sie die fertig bestückte Platine noch einmal überprüfen und auch darauf achten, daß alle ICs ihre Kerbe auf der richtigen Seite haben.
Wenn Sie alles richtig gemacht haben, und das Listing 1 ("SPRTEST.BAS") eingegeben und gestartet haben, müßte Ihr SPRECHI Sie nun mit einem sonoren "Hallo, User!" begrüßen.
Nachdem Sie nun mit SPRECHI einen leistungsfähigen Sprachsynthesizer aufgebaut und an Ihrem CPC angeschlossen haben, können wir zur Programmierung des sprechenden Genossen kommen.

Die Programmierung

Wichtigster Bestandteil jedes Sprachsynthesizers ist ganz offensichtlich der eingesetzte Sprachchip. Bei SPRECHI haben wir den SSI 263 von Silicon Systems, einen Sprachchip der dritten Generation benutzt: Er beherrscht sogar die deutschen Umlaute und kann auch ein (immer noch etwas amerikanisch klingendes) deutsches "R" sprechen. Außerdem kommt er mit einer sehr niederigen Daten-Rate aus (12 Byte pro Sekunde; bei höch-

Seit dem 10. 8. im Handel
Joystick
ABONNEMENT
Endlich eine Zeitschrift, die Ihnen hilft, mit all den Monstern in Ihrem Computer fertig zu werden! Spannend wie ein Comic, unterhaltend wie ein Computerspiel
Yippie
Hilft mir denn keiner?
Toll
Joystick – denn Computerspielen ohne Joystick ist wie Pacman ohne Geister
DMV Verlag · Postfach 250 · 3440 Eschwege
Jetzt neu



Die fertig aufgebaute Platine des Sprachsynthesizers

ster Sprachqualität kann sich dieser Wert verfünffachen), so daß er auch von BASIC angesteuert werden kann.

Ein weiterer wichtiger Vorteil des SSI 263 ist, daß dem Programmierer nahezu alle Möglichkeiten der Sprachmanipulation offenstehen. Sowohl die Lautstärke als auch die Tonhöhe sind in weiten Grenzen recht fein einstellbar, weshalb man Sprache mit großer Dynamik oder sogar Gesangsstücke mit dem SSI 263 produzieren kann (ein Beispiel ist das Demoprogramm "7BR.BAS", das SPRECHI dazu veranlaßt, den Refrain von Peter Maffays "Über sieben Brücken mußt du geh'n…" zu singen !). Auch Szenen, in denen man z.B. ein junges Mädchen, einen jungen Mann und einen alten Mann sprechen hört, sind relativ leicht zu verwirklichen. Durch weitere Manipulationen z.B. an Sprachgeschwindigkeit und -artikulation kann man auch eine Micky-Maus-ähnliche Stimme aus SPRECHI hervorzaubern oder einen leicht angetrunkenen Herren simulieren (s. Demoprogramm "SPRDEMO.BAS"). Der SSI 263 stellt auch die einzelnen Phoneme nicht einfach hintereinander, sondern verbindet sie mit einer vom Programmierer festlegbaren Geschwindigkeit, steuert z.B. beim Heben der Stimme die neue Frequenz schrittweise an, und kann noch vieles mehr, so daß die menschliche Sprache (bei genügend Programmieraufwand) ganz gut verständlich synthetisiert werden kann…

Sie merken sicher, daß man bei den Möglichkeiten, die der SSI 263 bietet, ins Schwärmen geraten kann; um SPRECHI hundertprozentig auszunutzen zu können, muß man ihn (hauptsächlich wegen der Geschwindigkeit) auf Assemblerebene programmieren, da einem erst dort alle Möglichkeiten (auch in bezug auf die Geschwindigkeit) offenstehen.

Weil es aber sicher nicht jedermanns Sache ist, SPRECHI (und damit den SSI 263) direkt mit Phonem- und Register-Daten zu "füttern" (was letztlich die bessere Qualität bringt), haben wir eine RSX- Befehlserweiterung entwickelt, die es auch Einsteigern ermöglicht, schnell und einfach tolle Sachen mit SPRECHI zu machen.

Oft wird es passieren, daß - nach einigem "Einspielen" mit den RSX-Befehlen - eigene Ideen verwirklicht werden sollen (z.B. ein Treiber-programm, das einen eingegebenen Text englisch ausspricht, ein neues Lied, …). Dann kann es sein, daß die Umsetzung mit den RSX-Befehlen nicht ganz gelingt, weshalb eine genaue Kenntnis des Innenaufbaus und der direkten Programmierung SPRECHIs (z.B. von Assembler aus) sehr von Vorteil ist, da man durch sie eigene, etwas kompliziertere Ideen meistens besser umsetzen kann. Zu diesem wichtigen Teil kommen wir später; erstmal wollen wir Ihnen Gelegenheit geben, sich mit den RSX-Befehlen "auszutoben" und erste Erfahrungen mit SPRECHI zu machen. Wenn Sie dann an dem oben beschriebenen Punkt angelangt sind, Sie also eine Idee nicht zur vollen Zufriedenheit mit den RSX-Befehlen umsetzen können, lesen Sie sich die Beschreibung der "direkten" SPRECHI-Programmierung durch und sollten dann in der Lage sein, Ihre Idee selber optimal verwirklichen zu können.

Wir wollen Sie nun nicht länger auf die Folter spannen und gleich Beschreiben, wie Sie SPRECHI mit den RSX-Befehlen das Sprechen beibringen: Tippen Sie zuerst das BASIC-Programm "SPRECHI.BAS" ab, das aus den DATAs das Binärfile "SPRECHI.RSX" generiert, wenn alle DATAs in Ordnung waren. Dieses "SPRECHI.RSX" enthält viele leistungsfähige RSX-Befehle zur leichten Ansteuerung SPRECHIs von BASIC aus. (Einige dieser Befehle werden in den Demoprogrammen "ROTKAEPP.BAS" und "SPRDEMO.BAS" benutzt, weshalb diese Demos nur funktionieren, wenn dieses File "SPRECHI.RSX" von Diskette bzw. Kassette nachgeladen werden kann!)

Um die neuen Befehle zur Verfügung zu haben, gehen Sie so vor: Setzen Sie zunächst Ihren CPC mit <CTRL>+<SHIFT>+<ESC> zurück und laden und initialisieren Sie hierauf "SPRECHI.RSX" und mit folgendem Einzeiler:

```
MEMORY &7FFF:LOAD"SPRECHI.RSX":C
ALL &89ED
```

Nun stehen Ihnen, wie schon oft gesagt, zahlreiche RSX-Befehle zur Verfügung, die wir nun beschreiben wollen (denken Sie bitte immer daran, daß alle RSX-Befehle mit dem "senkrechten Strich", den man mit <SHIFT+Klammeraffe> erzeugt, beginnen).

Die Befehle

Der wichtigste Befehl ist natürlich

|SAG,@a$ bzw. |SAY,@a$,

denn dieser schickt den Text, den die Textvariable a$ enthält, zu SPRECHI, der diesen Text dann ausspricht. (CPC 664/6128- und Besitzer eines VORTEX-Laufwerkes können statt '|SAG,@a$' auch '|SAG,"Text"' eingeben). Wenn Sie sich also mit "Guten Tag, Boss!" von SPRECHI begrüßen lassen wollen, gehen Sie so vor:

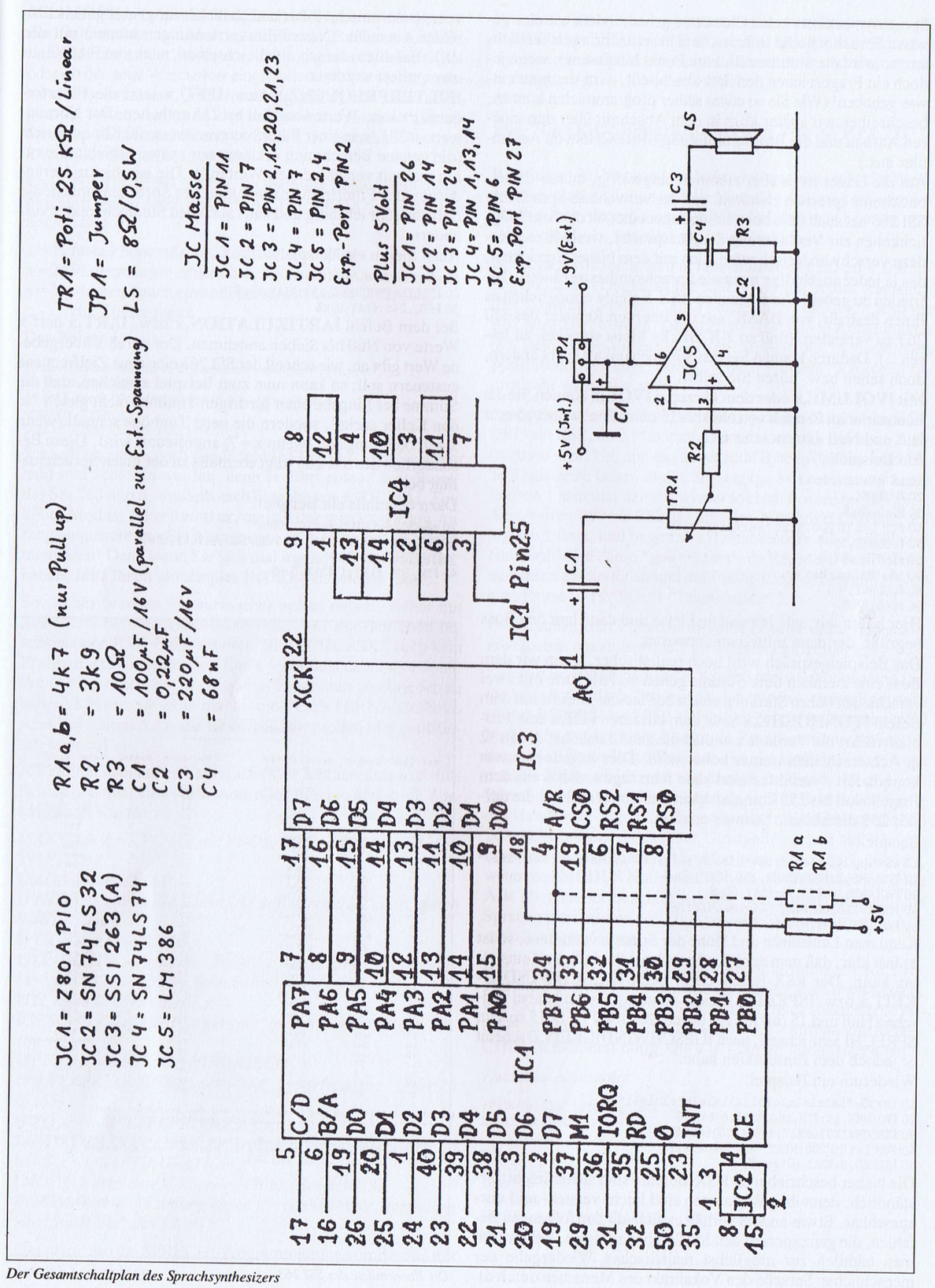
```
10 a$="GutenTag Boss!"
20 |SAY,@a$
```

Auch längere Texte sind kein Problem:

```
10 text$="Hia ist SPRECHI, der PE TSE INTERNATSIONAL Sprachsyntee
   saiser guten Tag und viil Spass beim Äxperimentieren."
20 |SAG,@a$
```

Wie Sie am zweiten Beispiel sehen (besser: hören), ist SPRECHI oft leichter und besser zu verstehen, wenn man die Rechtschreibungsregeln etwas vernachlässigt und dafür mehr auf die Aussprache achtet.

Sicher haben Sie außerdem bemerkt, daß wir versucht haben, dem berühmt-berüchtigten eintönigen "Gebrabbel" jedes



Der Gesamtschaltplan des Sprachsynthesizers

Sprachsynthesizers keine Chance zu geben, indem wir eine gewisse Sprachmelodie in jeden Satz hereinzubringen versuchten: so wird die Stimme z.B. zum Punkt hin gesenkt; wenn jedoch ein Fragezeichen den Satz abschließt, wird sie immer etwas gehoben! (Wie Sie so etwas selber programmieren können, beschreiben wir später kurz in dem Abschnitt über den inneren Aufbau und die Programmierung SPRECHIs von Assembler aus.)

Auf die Dauer ist es aber ziemlich langweilig, immer dieselbe Stimme sprechen zu hören, und der verwendete Sprachchip SSI 263 hat auch (wie oben schon angedeutet) noch viele Möglichkeiten zur Veränderung der Aussprache, weshalb es geradezu verschwenderisch wäre, sich mit dem bisher Erreichten, das ja jeder anständige normale Sprachsynthesizer bietet, zufrieden zu geben! Die folgenden RSX-Befehle ermöglichen es Ihnen deshalb, von BASIC aus die internen Register des SSI 263 zu verändern (und so z.B. Micky Maus sprechen zu lassen...). Dadurch können Sie den Stimmcharakter leicht ändern, doch sehen bzw. hören Sie selbst:

Mit **|VOLUME,x** oder dem kürzeren **|VOL,x** können Sie die Lautstärke im Bereich von Null bis 15 einstellen, wobei 15 sehr laut und Null extrem leise ist.

Ein Beispiel:

```
10 a$="Hallo, Boss."
20 |VOLUME,4
30 |SAG,@a$
40 FOR i=1 TO 200:NEXT 'Warteschleife
50 |VOLUME,15
60 |SAG,@a$
70 b$="Hallo, Mai er
80 |VOLUME,9
90 |SAG,@b$
```

Hier hören wir, wie jemand erst leise und dann laut den Boss begrüßt, der dann mittellaut antwortet.

Das Beispielgespräch wird noch realistischer, wenn wir dem Boss eine ziemlich tiefe Stimme geben und die Rufe mit zwei verschieden hohen Stimmen erschallen lassen. Dies ist mit dem Befehl **|TONHOEHE,x** bzw. dem kürzeren **|TH,x** kein Problem mehr; die Variable x enthält die neue Tonhöhe, die ab 32 in Achterschritten immer höher wird. Dies ist jedoch etwas kompliziert - vereinfachend kann man sagen, daß x aus dem Breich Null bis 255 kommen kann und der Wert Null die tiefste; 255 die höchste Stimme erzeugt.

Beispiel:

```
10 a$="Hallo, Boss.":b$="Bist du es?":c$="Alles klar, ich bins."
20 |VOLUME,7:|TONHOEHE,150:|SAY, @a$
30 |VOLUME,10:|TH,56:|SAY,@b$
40 FOR i=1TO200:NEXT 'Warteschleife
50 |VOLUME,12:|TH,0:|SAY,@c$
```

Kann man Lautstärke und Höhe der Stimme verändern, so ist es fast klar, daß man auch die Sprechgeschwindigkeit einstellen kann. Der RSX-Befehl dafür heißt **|GESCHWINDIGKEIT,x** bzw. **|SPEED,x** (beides ist gültig), x muß dabei zwischen Null und 15 (inklusive) liegen. Nach |SPEED,15 spricht SPRECHI sehr schnell; nach |GESCHWINDIGKEIT,0 scheint er jedoch dem Einschlafen nahe.

Wiederum ein Beispiel:

```
10 text$="Lalalalalololililalelelelolali"
20 |VOLUME,13:|TH,160:|SAY,@text$
30 |GESCHWINDIGKEIT,1:|SAY,@text$
40 FOR i=1 TO 200:NEXT 'Warteschleife
50 |SPEED,8:|SAY,@text$
```

Die bisher beschriebenen RSX-Befehle sind sicher leicht verständlich, denn ihre Wirkungen sind leicht versteh- und voraussehbar. Etwas anders verhält es sich mit den folgenden Befehlen, die ganz speziell den SSI 263 beeinflussen. Bei ihm hat man nämlich zur möglichst realistischen Wiedergabe der menschlichen Sprache den Vokaltrakt des Menschen durch diverse elektronische Filter und ähnliches möglichst gut nachzubilden versucht. Diese Filtervorrichtungen können mit den RSX-Befehlen, die wir nun beschreiben, auch von BASIC aus manipuliert werden:

|FILTERFREQUENZ,x bzw. **|FFQ,x** setzt die Filterfrequenz; x kann Werte von Null bis 255 enthalten. Der Normalwert ist 231, was einer Filterfrequenz von ca. 20 Khz entspricht (die genaue Berechnung erklären wir später - spielen Sie ruhig mal mit verschiedenen Werten !). Die geschickte Veränderung der Filterfrequenz während des Sprechens macht die Sprache sehr lebendig und kann auch den Stimmcharakter verändern.

Auch hierzu ein Beispiel:

```
10 a$="Hallo du da !"
20 |FILTERFREQUENZ,131:|SAY,@a$
30 |FFQ,241:|SAY,@a$
```

Bei dem Befehl **|ARTIKULATION,x** bzw. **|ART,x** darf x Werte von Null bis Sieben annehmen. Der durch x übergebene Wert gibt an, wie schnell der SSI 263 eine neue Zielfrequenz ansteuern soll; so kann man zum Beispiel erreichen, daß die Stimme bei Eingabe einer niedrigen Tonhöhe nicht gleich "in den Keller sackt", sondern die neue Tonhöhe schnell (wenn x=0) oder langsam (wenn x=7) angesteuert wird. Diese Besonderheit des SSI 263 trägt ebenfalls zu der guten Sprachqualität bei.

Dazu ebenfalls ein Beispiel:

```
10 a$="Dummdididummdidi damm"
20 |TH,200:|ARTIKULATION,7:|SPEED,9:|SAY,@a$
30|TH,32:|SAY,@a$
40 |TH,200:|ART,1:|SAY,@a$
```

PHONEMLISTE SSI 263

Hex-Code	Symbol	Beispielwort (in Englisch)	Hex-Code	Symbol	Beispielwort (in Englisch)
00	PA	(Pause)	20	L	lift
01	E	meet	21	L1	play
02	E1	bent	22	LF	fall (End-L)
03	Y	before	23	W	water
04	Y1	year	24	B	bag
05	AY	please	25	D	paid
06	IE	any	26	KV	tag
07	I	six	27	P	pencil
08	A	made	28	T	tart
09	AI	care	29	K	kit
0A	EH	nest	2A	HV	(Vokal halten)
0B	EH1	belt	2B	HVC	d(h)oubt
0C	AE	dad	2C	HF	heart
0D	AE1	after	2D	HFC	p(h)ound
0E	AH	got	2E	HN	(Nasal halten)
0F	AH1	father	2F	Z	zero
10	AW	office	30	S	same
11	O	store	31	J	pleasure
12	OU	boat	32	SCH	sheep
13	OO	look	33	V	very
14	IU	you	34	F	five
15	IU1	could	35	THV	there
16	U	tune	36	TH	with
17	U1	moon	37	M	most
18	UH	wonder	38	N	nine
19	UH1	love	39	NG	song
1A	UH2	what	3A	:A	Mädchen (*)
1B	UH3	hut	3B	:OU	Möwe (*)
1C	ER	bird	3C	:U	Tür (*)
1D	R	roof	3D	:UH	menu (+)
1E	R1	rug	3E	E2	Bitte (*)
1F	R2	Kutter.(*)	3F	LB	il (+)

(*) bedeutet, daß das Beispielwort aus dem Deutschen kommt;
(+), daß es sich um ein französiches Wort handelt.

(Beachten Sie bitte, daß den Hex-Codes in BASIC ein "&" (SHIFT+6) vorangestellt werden muß, wenn sie direkt mit dem OUT-Befehl an SPRECHI gesendet werden sollen.)

Die Phonemliste des SSI 263

Mit **|MODE,x** (x von Eins bis Drei) können Sie einstellen, ob die Zielfrequenz, wie oben beschrieben, gemäß der Artikulationsrate mehr oder weniger schnell angesteuert werden soll oder ob der neue Wert sofort eingestellt wird. Die sofortige Ansteuerung der neuen Tonhöhe ermöglicht es z.B., Lieder singen zu lassen, während die schrittweise Ansteuerung die Sprachqualität entscheidend verbessert. Je nach dem Wert von x wird der Modus ausgewählt:

x=0: NICHT BENUTZEN, da dann die Schnittstellenlogik blockiert wird (durch |NORMAL (s.u.) kann man sich aber noch retten !))

x=1: feste Phonemdauer, Zielfrequenz sofort
x=2: Phonemdauer einstellbar, Zielfrequenz sofort
x=3: Phonemdauer einstellbar, schrittweiser Übergang zur Zielfrequenz

Das Beispiel:

```
10a$="Dammdududumm dididommdomm bammbumm":|TH,32
20 |MODE,3:|TH,320
30 |SAY,@a$ 'Stimme hebtsich schrittweise
40|MODE,1:TH,32
50|SAY,@a$'Stimme faellt sofort
```

Schließlich gibt es noch den Befehl **|NORMAL**, den Sie sicher bald sehr schätzen werden, denn er führt einen "Soft-Reset" des SSI 263 durch, weshalb nach Eingabe von |NORMAL Tonhöhe, Modus, Filterfrequenz, und all das wieder wie am Anfang eingestellt ist - Sie können also nach Herzenslust experimentieren! Denn wenn Sie sich mal irgendwo "festgefahren" haben, hilft Ihnen ein simples |NORMAL wieder raus!

So, sicher brennen Sie inzwischen schon darauf, selber mit SPRECHI herumzuexperimentieren, was mit den eben beschriebenen RSX-Befehlen von "SPRECHI.RSX" auch kein Problem ist; wir wünschen Ihnen jedenfalls dabei viel Spaß. Wenn Sie meinen, ein interessantes Programm geschrieben zu haben, können Sie es ja mal an die PC INTERNATIONAL schicken, damit dann alle Leser von Ihrer tollen Idee profitieren können!

Als kleine, aber äußerst nützliche, Gedächtnisstütze und -hilfe hier nochmal eine Auflistung aller Befehle und ihrer Wirkungen in Kurzform:

|SAY,@a$ (bei CPC 464 mit VDOS, CPC 664/6128 geht auch |SAY,"Text")
|SAG,@a$ wie |SAY
|VOLUME,x stellt die Lautstärke softwaremäßig ein (Bereich 0-15)
|VOL,x wie |VOLUME,x
|TONHOEHE,x setztdie Tonhöhe (Bereich0-255, effektiv ber"nur" 32 bis 255 in Achterschritten)
|TH,x wie |TONHOEHE,x
|GESCHWINDIGKEIT,x setztdie Sprachgeschwindigkeit (Bereich 0-15)
|SPEED,x wie|GESCHWINDIGKEIT,x
|FILTERFREQUENZ,x setzt die Filterfrequenz (Bereich 0-255)
|FFQ,x wie |FILTERFREQUENZ,x
|ARTIKULATION,x setzt die Artikulationsrate (Bereich 0-7)
|ART,x wie |ARTIKULATION,x
|MODE,x setzt den Modus des Chips (Bereich1-3)
|NORMAL setzt die Anfangsdaten für Tonhöhe,etc., sozusagen ein "SoftReset" des Chips

Das wäre also die SPRECHI-Programmierung von BASIC mit den RSX-Befehlen.

Für kompliziertere Programme (z.B. ein Programm, das SPRECHI auf ENGLISCH reden läßt) ist jedoch die Kenntnis des inneren Aufbaus und der "direkten" Programmierung unbedingt nötig ...

Die "direkte" Programmierung von SPRECHI

Bitte haben Sie Verständnis dafür, daß wir hier nicht alle Fachworte erklären können und die Materie ziemlich komprimiert beschreiben, denn für ausführliche Erklärungen braucht man schon ein Buch, um alles unterzubekommen.

Betrachten wir zunächst den Innenaufbau des SSI 263: Die anliegenden acht Datenbits D0 bis D7 werden in fünf Registern (für Tonhöhe, Phonem, etc.) gespeichert, die durch RS0 bis RS2 adressiert werden. Die Sprache wird nun in Abhängigkeit der Register in einem analogen Modell des menschlichen Sprachtraktes erzeugt, wobei - je nach Phonem - Rauschen zugemischt wird oder nicht.

Das anliegende Datenbyte kann also, je nach Zustand von RS0, RS1 und RS2, eine Phonemnummer oder z.B. die neue Zielfrequenz sein. Um nun das anliegende Byte in das entsprechende Register zu laden, muß CS0 (wenige Mikrosekunden lang) logisch 1 sein und danach wieder logisch 0 werden. Beispiel: Adressieren RS0 bis RS2 das Register 0, während CS0 kurz auf logisch 1 liegt, und liegen alle Datenbits derweil ebenfalls auf Null, wird eine Pause "gesprochen", da Register 0 für Phonemnummern zuständig ist und das Phonem mit der Nummer Null eine Pause ist (s. Tab. 1 "Phonemliste").

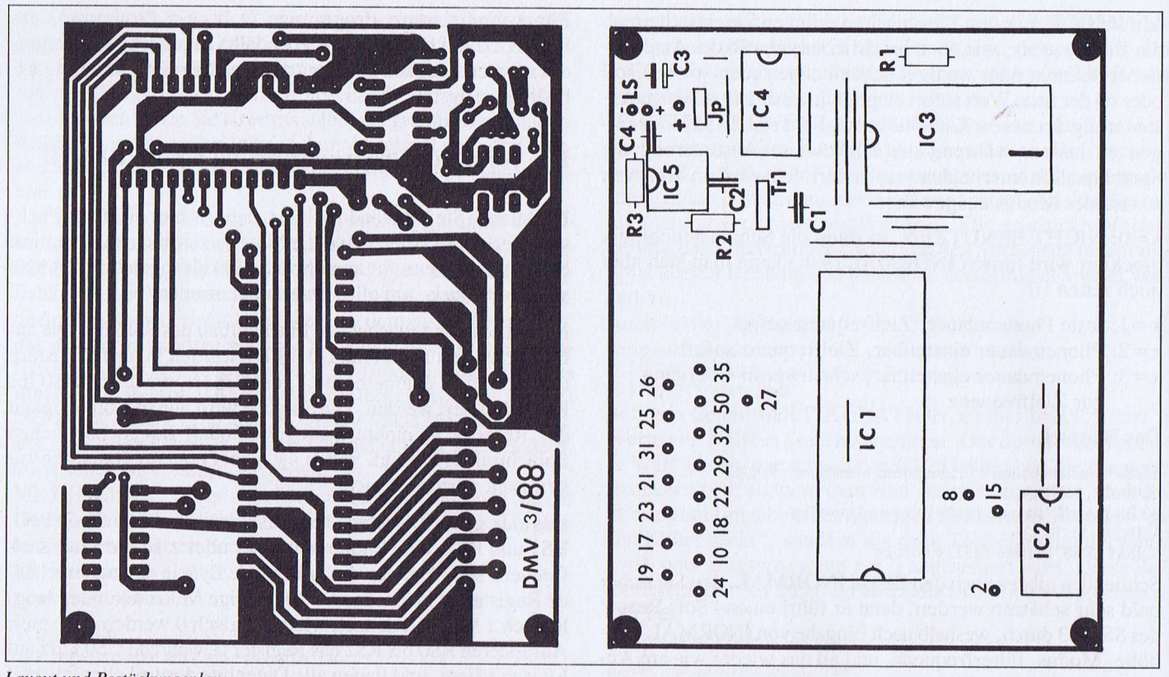
Die Nummern der Phoneme und die Phoneme selbst finden Sie, jeweils mit einem Beispiel, in Tabelle 1 aufgelistet.

Neben dem CS0-Eingang des SSI 263 (zum Schreiben von Daten kurz auf logisch 1 zu legen), den nun bekannten Registern (durch RS0, RS1 und RS2 andressiert) und dem Datenbus (D0 bis D7) ist nur noch ein Bit für die Programmierung wichtig: das A/R (Acknowledge/Request)-Bit, das logisch 1 ist, solange der SSI 263 beschäftigt ist und danach auf 0 geht. Über dieses Bit erfolgt also das "Händeschütteln". Eine Berücksichtigung des Handshake-Signals A/R ist vor der Initialisierung des SSI 263 nicht ratsam, da es unter Umständen schon auf 1 liegen kann und so nie initialisiert werden könnte (was auch passiert, wenn man MODUS Null wählt!!); nach der Initialisierung sollte A/R aber immer beachtet werden, wenn man verständliche Sprache produzieren will (für spezielle Effekte kann man A/R natürlich ignorieren).

Jetzt kennen Sie bereits alle wichtigen Grundlagen zur Programmierung des SSI 263, aber Sie müssen auch noch wissen, wie Sie mit dem "Ding" in Verbindung treten: dazu ist die Parallel-Schnittstelle mit der Z80A PIO (s. "Aufbau von SPRECHI", Schaltplan) nötig, die wie folgt zu initialisieren ist:

Zuerst in Assembler

```
;PIO-Initialisierung LD A ,207
LD BC ,#F8F2
OUT (C),A
SUB A
OUT (C),A
LD BC ,#F8F3
LD A ,207
OUT (C),A ;    Port A(&F8F0): Datenbyte
LD A ,128 ;
               Port B(&F8F1):
               Register und Handshake
OUT (C),A
```



Layout und Bestückungsplan

Das Ganze nun in BASIC:

Achtung Hardware-Freunde!!
In letzter Minute hat sich ein Fehler gezeigt!! Die Stromversorgung des CPC reicht nicht aus!! Trennen Sie bitte den Anschluß vom Rechner an PIN 27 auf (+ 5 V intern) und legen Sie eine externe 5 Volt-Spannung an diesen PIN.

```
1'PIO-Initialisierung
10 OUT &F8F2,207
20 OUT &F8F2,0
30 OUT &F8F3,207
40 OUT &F8F3,128
```

Nach Ausführung dieser Initialisierungsroutine ist die PIO bereit: Port A mit der Adresse &F8F0 ist mit dem Datenbus des SSI 263 verbunden; PORT B (Adresse &F8F1) erhält im siebten Bit das A/R-Signal für den Handshake. Bit 6 liegt am CS0-Eingang des SSI 263 und wird als Schreibsingnal benötigt. Die Bits 3,4 und 5 sind mit RS0, RS1 und RS2 des SSI 263 verbunden, wählen also das Register des SSI 263 an, in das das Datenbyte geladen werden soll. An die noch freien Bits 0,1 und 2 können Sie eigene Erweiterungen anschließen, abfragen und Steuerimpulse ausgeben.

Mit diesem Wissen können wir nun – nach der Initialisierung der PIO –den SSI 263 (und damit unseren lieben kleinen SPRECHI) voll ausreizen, und wir wollen Ihnen nun zum schnellen und leichten Einstieg elementare Routinen für Programme, die mit SPRECHI arbeiten sollen vorstellen. Selbstverständlich werden wir diese Prograämmchen erklären, so daß Sie diese Routinen selber verändern und in Ihren Programmen einsetzen können. (Anregungen und Verbesserungsvorschläge sind natürlich immer willkommen...)

Die wichtigste Routine ist sicher die, die ein Datenbyte in ein Register des SSI 263 lädt, denn nur so kann man Sprache oder Musikeffekte erzeugen. Wir haben dafür das folgende Programm entwickelt, bei dem - je nach Einsprungadresse - mit oder ohne Handshake (also mit oder ohne Warten, bis der SSI 263 nicht mehr beschäftigt ist) ein Datenbyte in ein Register geladen wird. (Normalerweise muß das Busy-Signal (Handshake) beachtet werden, bei der Initialisierung jedoch sollte man es sicherheitshalber ignorieren... (s.o.)). Die BASIC- und die Assembler-Routine haben genau dieselben Übergabeparameter:

H enthält die Registerauswahl (s. Tabelle 2) und L enthält das Datenbyte, das in das ausgewählte Register geladen werden soll.

Ein "CALL daus" in Assembler bzw. "GOSUB 1020" in BASIC bewirken, daß das Datenbyte OHNE Rücksicht auf das Handshake-Signal an den SSI 263 geschickt wird (was bei der Initialisierung sehr ratsam ist). Mit "GOSUB 1010" in BASIC bzw. "CALL aus" in Assembler wird das Byte erst dann in das Register geschrieben, wenn der Chip bereit ist (im normalen Betrieb benutzt man diese Übergabemethode). Da die Sprache bekanntlich über Phoneme (s.Tabelle 1) synthetisiert wird, muß man zur Ausgabe eines Phonems also H=0 und L="Phonemnummer" setzen. Wie wir alle wissen, sind Programmierer faule Leute und deshalb bauten wir in Assembler die Routine "paus" ein (in BASIC mit "GOSUB 1000" zu erreichen), die H automatisch auf Null setzt, so daß man sich in Assembler das lästige "LD H,0" und in BASIC das "h=0" bei jedem Phonem spart, was sowohl Zeitvorteile als auch Speicherplatzgewinn bringt. Doch nun zur Routine:

Übergabe eines Datenbytes in ein Register des SSI263

Zuerst in Assembler:

```
                        ;Datenbyte (L) in
                        ;h=Datenbyte,
                        ;SSI 263-Register (H) laden
                        ;l=Register-Selekt
paus:
LD H,0                  ;PhonemAUSgeben, Register 0:
LD BC,&F8F1             ;Handshake über Port B
IN A,(C)                ;Zustandeinlesen
```


CPC DATABOX

— Das ist die Software zum CPC Sonderheft 7/88 —

DATABOX:

— mehr als der übliche Softwareservice·

— bringt alle Programme des Sonderheftes auf Kassette oder 3" Diskette

— Die Programme sind, durch den auf dem Datenträger enthaltenen CPC-Emulator, auf allen drei CPC-Rechnern lauffähig.

— trägt das Titelbild des gleichzeitig erscheinenden Sonderheftes.

– über 400 KByte Software für Ihren CPC!



Für alle CPC's als Kassette und 3"-Diskette

Inhalt der Databox zu Sonderheft 7/88

Diskette/Kassette 1:

Amsmon plus	Basic 765
Auxilia	Domingo
H'Cardin	Musicus
Screen Pack plus	Checksummer
Mission Wega	Acceleration
Impuls	Taschenrechner
□ Filetransfer	● Emulator
Page	Huelle

Diskette/Kassette 2

3D-Tron	Mueller
Cosmic	Chemystical
Komplex	Pascalsche Dreiecke
Biorythmus	Backup
Basic Programmcode-Komprimierer	Copy all Disks
CP/M Beispielprogramm	● Ein Stein
● Zeilen-Generator	Sprechi-Der Laberkaste
Labyrinth	Vereinsverwaltung
Kompos	● CP/M-Verbesserungen

● Nur für 464 □ Nur für 6128

Einzelbezugspreise für DATABOX:

je Diskette 3"		24,– DM	je Kassette		14,– DM
Kombi-Pack (3"Diskette 1 + 2)	nur	38,– DM	Kombi-Pack (Kassette 1 + 2)	nur	25,– DM

Coupon ausschneiden, auf eine Postkarte kleben oder per Brief senden an:

DMV-Verlag, Postfach 250, 3440 Eschwege

»CPC-Bestellservice«

Ich bestelle hiermit aus Ihrem Angebot:

Anz.	Best.-Nr.	Bezeichnung	Preis
	201	Copyshop (Hardcopy-Programm) Cass.	59,– DM
	202	Copyshop 3" Disk.	69,– DM
	203	Copyshop (Vortex) 5,25" Disk.	69,– DM
	101	Power-Spiele, 4 Cass.	50,– DM
	102	Power-Spiele, 4 3" Disk.	70,– DM
	103	Startest, Cass.	24,– DM
	104	Startest 3" Disk.	29,– DM
	106	Know 3" Disk.	29,– DM
	204	Special Offer No.1 3" Disk.	69,– DM
	205	Special Offer No.2 3" Disk.	69,– DM
	107	Special Offer No.3 3" Disk.	49,– DM
	206	Context CPC Cass.	49,– DM
	207	Context CPC 3" Disk.	59,– DM
	209	FAstBAsicCOMpiler 3" Disk.	69,– DM
	1011	Fantastic Four	49,– DM
	307	CPC Sonderheft 1/86	14,– DM
	308	CPC Sonderheft 2/86	14,– DM
	309	CPC Sonderheft 3/86	14,– DM
	310	CPC Sonderheft 4/87	14,– DM
	311	CPC Sonderheft 5/87	14,– DM
	312	CPC Sonderheft 6/88	14,– DM
	3130	Databox zum CPC Sonderheft 7/88 Kass. 1	14,– DM
	3131	Databox zum CPC Sonderheft 7/88 Kass. 2	14,– DM
	3132	Databox zum CPC Sonderheft 7/88 Kombi-Kass.	25,– DM
	3133	Databox zum CPC Sonderheft 7/88 Disk. 1 3"	24,– DM
	3134	Databox zum CPC Sonderheft 7/88 Disk. 2 3"	24,– DM
	3135	Databox zum CPC Sonderheft 7/88 Kombi-Disk.	38,– DM

Porto/Verpackung (Unabhängig von der Bestellmenge Inland 3,– DM/Ausland 5,– DM) _____

Gesamtbetrag _____

□ Den Betrag bezahle ich mittels des beigefügten Verrechnungsschecks
□ Ich bitte um Lieferung per Nachnahme, zuzügl. der Nachnahmegebühr (nur innerhalb der BRD)

Datum _____ Unterschrift (bei Minderjährigen des gesetzl. Vertreters) _____

Absender: *(Bitte genaue Anschrift angeben)*

Name

Vorname

Str./Nr.

PLZ/Ort

```
CP128           ;SSI 263 busy ?
JR Z,wait       ;wenn ja, weiter abfragen
daus:
LDBC,&F8F0      ;Port A=Datenbyte
LD A,L
OUT (C),A       ;Datenbyteausgeben
LD BC,&F8F1     ;PortB=Reg.+Handshake
LD A,H
OR 64           ;CS0 auf logisch 1
OUT (C),A       ;Register auswählen
LDA,H
AND56           ;CS0 wieder aufNull
OUT (C),A
RET
```

Und hier in BASIC:

```
1000 h=0
1010 IF INP(&F8F1)=128 THEN 1010 :'Handshakewait
1020 OUT&F8F0,l
1030 OUT &F8F1,hOR 64
1040 OUT &F8F1,h AND 56
1050 RETURN
```

Soll der Busy-Zustand NICHT beachtet werden, so ist in Assembler "daus" und in BASIC Zeile 1020 anzuspringen. Nachdem wir jetzt mit Hilfe der Datenbyte-Übergabe-Routine recht einfach mit dem SSI 263 kommunizieren können, ist die Initialisierung des Chips kein Problem mehr: Nach dem Einschalten des CPC befindet sich der SSI 263 in irgendeinem, nicht vorhersehbaren Zustand; um nun Daten austauschen zu können und somit Sprache bzw. Musikeffekte zu produzieren, muß sich jener SSI 263 in einem bestimmten Zustand befinden, den die folgende Initialisierungsroutine herstellt. Dazu wird (wie zu allem, was den SSI 263 betrifft) die oben beschriebene Datenbyte-Übergabe-Routine wird benötigt. Da der Zustand des SSI 263, wie schon gesagt, unbestimmt ist, wird das Busy-Signal nicht berücksichtigt, da es sonst passieren kann, daß man "ewig" darauf wartet, daß der Chip frei ist!

Assembler:

```
;Initialisierung des SSI 263
LD L,128      ;CTL=1
LD H, 24      ;Ampl.,Art.,Control
CALL daus     ;direkt ausgeben
LD L,192      ;Modus 3
LD H,0        ;Datenbus
CALL daus     ;"Busy" NICHT abfragen
LD L,127      ;CTL=0
LD H, 24      ;Ampl.,Art.,Control
CALL daus     ;direkt ausgeben
              ;fertig !!
```

BASIC:

```
1 'Initialisierung des SSI 263
10 l=128:h=24:GOSUB 1020
20 l=192:h=0:GOSUB1020
30 l=127:h=24:GOSUB 1020
40 'SSI263 ist initialisiert
50 'jetzt kann man z.B. Tonhoehe,
60 'setzen (s."HALLO.BAS" !)
```

Zum Schluß möchten wir Ihnen ein kleines Modell unseres "Text to Speech"-Programms, das SPRECHI deutsch reden läßt, vorstellen (da solche Programme mit der Zeit recht kompliziert und relativ lang werden, sollte man sie gleich in Assembler programmieren, was dann auch Vorteile wie größere Geschwindigkeit und weniger Speicherplatz bringt):

Zuerst muß der vom Benutzer eingegebene Text in Phoneme übersetzt werden. Diese Phoneme sollten hintereinander stehen, so daß am Ende des Übersetzungs- bzw. Compiliervorgangs der eingegebene Text als Phonemfolge vorliegt. Bei deutschem Text muß man natürlich berücksichtigen, daß "SCH" nicht als "S","C","H", sondern eben als "SCH" (Phonemnummer &32) vorliegen muß, daß einem Vokal folgendes "H" nicht gesprochen wird, und, und, und ... (jede Sprache hat da ihre Eigenheiten). Allein die Erkundung von Regeln in einer Sprache macht schon Spaß (und Mühe); am schönsten ist es aber, wenn ein selbst geschriebenes Programm einen Text so super in Phoneme übersetzt, daß fast jeder eingegebene Text richtig (also halbwegs verständlich) ausgesprochen wird! Hat man einmal so ein Programm, kann man sich daran machen, weitere Sprachcharakteristika wie Sprachmelodie, Sprachgeschwindigkeit, usw. mit zu berücksichtigen, so daß die Stimme zum Beispiel zum Punkt hingesenkt, zum Fragezeichen hin jedoch gehoben wird!

Sie sehen also, daß das Übersetzungsprogramm am wichtigsten (und damit auch am kompliziertesten und längsten) ist, weshalb wir hier kein konkretes Beispiel geben können (der Assembler-Quellcode unseres Übersetzungsprogramms ist knapp 12K lang!).

Die Ausgabe einer bereits compilierten Phonemfolge an den SSI ist dagegen nicht allzu schwer und auch als Demonstration dafür geeignet, wie man SPRECHI zum Reden bringen kann: (Die Datenbyte-Übergabe-Routine wird natürlich auch hier benötigt)

```
talk:
LD HL,phonl     ;HL zeigt auf den Anfang
                ;der Phonemliste speak:
LD A,(HL)
CP #FF          ;fertig ?
RET Z
LD L,A          ;L enthält die Phonemnummer
CALL paus       ;über "paus" an den SSI 263 ausgeben
INCHL           ;HLauf nächstes Phonem
JRspeak
phonl:
DEFB 0,#2C,#D,  ;zu sprechende Phoneme
#21,#12
DEFB #FF        ;#FF kennzeichnet das Ende
```

Nach Anspringen von "talk" mit "CALL talk" bzw. "JP talk" werden die ab "phonl" abgelegten Phoneme unter Berücksichtigung des Busy-Signals an den SSI 263 gesendet. (Dadurch wird sichertgestellt, daß erst dann ein neues Phonem gesprochen wird, wenn das aktuelle fertig ausgesprochen wurde). Ein weiteres besonderes Feature von SPRECHI darf nicht unerwähnt bleiben: die Möglichkeit, ihn SINGEN zu lassen (s. DEMO "Über sieben Brücken mußt Du geh'n ..."). Dazu wählt man Modus 1 oder 2 an, damit die Zielfrequenz sofort eingestellt wird - sich die Stimme also nicht, wie beim normalen Sprechen, schrittweise der Zielfrequenz annähert. Da man normalerweise die Noten des Liedes kennt und dadurch (mit Hilfe der Tabelle "Noten und Tobperioden" im Anhang des Handbuches) auch die entsprechenden Frequenzen ermitteln kann, muß man diese nur noch dem Tonhöhenregister des SSI 263 übermitteln. Dafür haben wir folgende Formel entwickelt:

SSIData=-62500/(Frequenz/7)+4096

In Zeile 240 des Demoprogramms "Über sieben Brücken..." ist die Anwendung dieser Formel im Zusammenhang zu sehen. Das "Filtern" der Bits 3 bis 10 für das Tonhöhenregister erledigt der lange Ausdruck mit "a=VAL("&X"+MID$(...))", denn die Bits 0,1,2 und 11 sind hierbei vernachlässigbar. (Wenn dies nicht der Fall ist, müßten sie gesondert in das Register für "Tonhöhe und Artikulation" geschrieben werden).

Wie die Tonhöhe kann auch die Filterfrequenz, die normalerweise bei 20 Khz liegt, verändert werden. Auch hier hat man eine Frequenz, die einem vorschwebt - sie muß nur dem SSI 263 irgendwie übermittelt werden ... dafür haben wir folgende Formel für Sie:

FF = -1E6/(2*Filterfrequenz)+256

"FF" ist der Wert, der in das Filterfrequenz-Register des SSI 263 geschrieben werden muß (von BASIC mit unserem RSX-Befehl "|FILTERFREQUENZ,FF" bzw. "|FFQ,FF" möglich). (Stören Sie sich nicht an dem "-1E6"; dies ist nur eine Kurzform für "-1000000", die sicher schneller und fehlerfreier eingegeben wird.)

So, jetzt wissen Sie eigentlich alles Wesentliche über SPRECHI und auch über Sprachsynthese, -programme und -probleme. Wahrscheinlich haben Sie schon irgendwelche Ideen (ein DATA-Zeilen-Vorleser, eine Brief-Vorleserin,...), die Sie sicher auch bald umsetzen werden.

Doch erfreuen Sie sich nicht allein an Ihren Kostbarkeiten, sondern lassen Sie die gesamte PC INTERNATIONAL-Leserschaft daran teilhaben, indem Sie Ihre SPRECHI-Programme an die Redaktion schicken!

Sollten Sie sich jedoch noch genauer über Sprachsynthese im allgemeinen und den SSI 263 im besonderen informieren wollen, so können wir Ihnen folgende Literatur empfehlen, die auch als Quelle der hier veröffentlichten Schaltung zu nennen sind:

1. Häder,D-P/Götzlaff,W:
"Sprechen mit dem Mikro" ausdem Vogel-Buchverlag, Würzburg1986 (erschienen in der Reihe"CHIP Wissen") ISBN 3-8023-0882-4
2. Silicon Systems: Datenblatt SSI 263, SSI263A Tustin,Kalifornien: SiliconSystems Inc., 1984 (oder Händler,die den SSI 263(A)führen)
3. Ciarcia,S.: "Buildathird generation phonetic speech synthesizer"BYTE, März 1984, S.28-42

Den einzigen Spezialchip, den SPRECHI enthält, ist der SSI 263; alle anderen Bauteile sind in jedem gut sortierten Elektronikladen (bzw. Versand) erhältlich. Diesen SSI 263(A) bekommen Sie bei:

Dr. Otto Soskuty
Reisstraße 16
6200 Wiesbaden

Atlantik Electronic GmbH
Fraunhoferstr. 11a
8033 Martinsried

Silicon Systems Inc.
14351 Myford Road
Tustin, CA 92680 Tel.(06121) 463822 Tel.(089)857208689 U.S.A.
(wenn möglich sollten Sie sich einen SSI 263A beschaffen, da dieser etwas besser klingt als der SSI 263, aber sehr groß ist der Unterschied wirklich nicht).
Die kleine Verstärkungs-Schaltung mit dem NF-Verstärker LM 386 wurde dem Buch
Siegfried Wirsum
Praktizierte Elektronik
Franzis' Verlag 1982
in Grundzügen entnommen.

(Jan-Mirko Maczewski/Jens Heimsoeth/jb)

Die Register des SSI 263

Register-Selektion				Register	Format der Daten							
RS0	RS1	RS2	PIO		D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	Phonem und -dauer	PD1	PD0	P5	P4	P3	P2	P1	P0
?	?	1	32	Filterfrequenz	F7	F6	F5	F4	F3	F2	F1	F0
0	1	0	16	Tonhöhe und Geschwindigkeit	G3	G2	G1	G0	T11	T2	T1	T0
1	0	0	8	Tonhöhe	T10	T9	T8	T7	T6	T5	T4	T3
1	1	0	24	Lautstärke, Control und Artikulation	CTL	A2	A1	A0	L3	L2	L1	L0

Datenbits	Bedeutung	Breich der Bits	Normalwert
P0 - P5	Phonemnummer	0 bis 63	-
G0 - G3	Sprachgeschwindigkeit	0 bis 15	10
PD0 & PD1	Dauer des Phonems (nur Modus 1 und 3)	0 bis 3	0
F0 - F7	Filterfrequenz	0 bis 255	233
L0 - L3	Lautstärke	0 bis 15	10
A0 - A2	Artikulationsrate	0 bis 7	5
T0 - T11	Tonhöhe		
	Modus 1 und 2: T0 - T11 Zielfrequenz	0 bis 4095	-
	Modus 3: T0 - T2 wie bei Modus 1/2	0 bis 7	-
	T3 - T5 Übergangs-geschwindigkeit	0 bis 7	4
	T6 - T10 Zielfrequenz	0 bis 31	-
	T11 "0"=tief, "1"=hoch	0 oder 1	-
CTL	Bestimmung des Modus' durch PD0 & PD1		

Wechselt das CTL-Bit von logisch 1 auf 0, so wird der Modus durch den Zustand von PD0 und PD1 festgelegt:

PD0 PD1

0 0 kein Einfluß auf den Modus, KEIN HANDSHAKE mehr !
0 1 Modus 1: Zielfrequenz sofort ansteuern, wobei PD0 & PD1 die Phonemdauer angeben
1 0 Modus 2: Zielfrequenz sofort ansteuern, Phonemdauer fest definiert
1 1 Modus 3: Zielfrequenz graduell ansteuern, wobei PD0 & PD1 die Phonemdauer angeben

Die Register des Synthesizer-Chips auf einem Blick.

für 464-664-6128

```
10 '                                              [117]
20 ' generiert: "SPRECHI.RSX"                     [2294]
30 '                                              [117]
40 ' 1987 by Jens Heimsoth   und                  [1183]
50 '          Jan-Mirko Maczewski                 [1141]
60 '                                              [117]
70 MODE 1:MEMORY &7FFF                            [1419]
80 r$=CHR$(30)+STRING$(5,CHR$(9)):PRINT"Ze        [3422]
ile";:z=102
90 FOR i=1 TO 3:READ adr,l:z=z+2                  [1107]
100 '                                             [117]
110 FOR a=adr TO adr+l STEP 10                    [1924]
120 PRINT r$;z;                                   [451]
130   FOR b=0 TO 4                                [951]
140     READ b$:x$=x$+b$                          [816]
150   NEXT b                                      [378]
160   FOR c=0 TO 9                                [818]
170     x=VAL("&"+MID$(x$,c*2+1,2))               [2211]
180     POKE c+a;x:sum=sum+x                      [610]
190   NEXT c:x$="":z=z+2                          [1058]
200   READ p$:psum=VAL("&"+p$)                    [1326]
210   IF sum<>psum THEN 330 ELSE sum=0            [1572]
220 NEXT a:z=z+2                                  [367]
230 '                                             [117]
240 PRINT CHR$(13)"Teil"i"OK.";CHR$(11);          [2195]
250 PRINT CHR$(13)"Zeile";                        [1556]
260 NEXT i                                        [375]
270 PRINT CHR$(30)SPC(10)                         [1182]
280 LOCATE 1,6:PRINT"Alle DATAs OK ..."           [2550]
290 PRINT:PRINT"Bitte Datentraeger einlege        [4699]
n und dann     eine Taste druecken !"
300 CALL &BB18                                    [389]
310 SAVE"SPRECHI.RSX",b,&8000,&1000               [1328]
320 END                                           [110]
330 PRINT CHR$(7)"DATAFEHLER!":END                [2054]
340 'Teil 1                                       [202]
350 DATA 32768,1129                               [721]
360 DATA FE01,C0EB,4678,FE00,C823,551             [575]
370 DATA 5E23,56EB,E578,320A,897E,462             [1857]
380 DATA FE61,3806,FE7B,3002,CBAE,4C1             [1619]
```

Listing. "SPRECHI"

```
390 DATA 2310,F23A,0A89,4F11,0090,2E2      [1614]
400 DATA E1ED,B001,F2F8,3ECF,ED79,6DC      [947]
410 DATA 97ED,7901,F3F8,3ECF,ED79,65C      [1693]
420 DATA 3E80,ED79,2E80,2618,CD10,3ED      [1699]
430 DATA 842E,C026,00CD,1084,2E7F,3A6      [851]
440 DATA 2618,CD10,84CD,5084,3ED6,454      [2147]
450 DATA 32CB,83ED,4B0A,8906,000D,35E      [1766]
460 DATA 2100,9009,E57E,21E6,8901,3AE      [1558]
470 DATA 0700,EDB1,E128,0823,3620,32F      [953]
480 DATA 233E,2018,057E,3620,2377,20C      [1161]
490 DATA FE20,2809,FE41,3807,FE5B,426      [864]
500 DATA 3003,2336,2E23,3600,2336,16C      [1465]
510 DATA 2E23,3600,2B2B,2B7E,FE21,2A5      [963]
520 DATA 2806,FE3F,2812,1825,3A02,21E      [920]
530 DATA 89F6,0F32,0289,6F26,18CD,3C5      [1091]
540 DATA 0F84,1815,3EC6,32CB,833A,37E      [1477]
550 DATA 0389,D608,3809,3203,896F,2D8      [2093]
560 DATA 2608,CD0F,8421,0090,2205,266      [1439]
570 DATA 8997,320B,893E,FE32,0989,3E6      [1320]
580 DATA 7EE5,21E6,8901,0700,EDB1,499      [1416]
590 DATA E128,0323,18F0,FE21,282E,3AC      [1350]
600 DATA FE2C,281A,FE2E,281C,FE3B,415      [1831]
610 DATA 2818,FE3A,280E,FE3F,201A,325      [1198]
620 DATA 3A03,89C6,08CD,CE83,1810,3DA      [1544]
630 DATA AF32,CC83,180A,3A03,89D6,3EE      [1980]
640 DATA 1038,03CD,CE83,3E08,32CC,3AD      [1253]
650 DATA 83CD,3684,2A05,897E,FE20,45E      [1750]
660 DATA 3823,FE5B,301F,D620,8721,3A1      [1356]
670 DATA 008F,4F06,0009,0146,817E,233      [1959]
680 DATA 0302,0323,7E02,CD5A,BB3A,2C7      [2000]
690 DATA 0489,FE01,CAA5,832A,0589,436      [1365]
700 DATA 2322,0589,3A0B,89B7,C2D3,3ED      [1501]
710 DATA 807E,FE41,30C5,E521,E689,5A7      [1386]
720 DATA 0107,00ED,B1E1,20B9,3E10,3AE      [1484]
730 DATA 320B,8923,7EB7,2863,2B7E,352      [1960]
740 DATA 18B4,CDB6,832A,0589,237E,42B      [1013]
750 DATA FE45,2805,2E0D,C301,8423,316      [1548]
760 DATA 7ECD,4884,FE55,282B,2E3A,425      [2074]
770 DATA C301,84CD,B683,2A05,8923,429      [1781]
780 DATA 7EFE,4928,0DFE,5528,16FE,489      [1942]
790 DATA 4828,1F2E,02C3,0184,2E0D,242      [1653]
800 DATA CD01,842E,06CD,0184,C348,3E3      [1216]
810 DATA 842E,11CD,0184,2E07,CD01,318      [888]
820 DATA 84C3,4884,2E02,CD01,842E,3C3      [1168]
830 DATA 02CD,0184,C348,842E,00C3,3D4      [986]
840 DATA 0184,2E24,C301,842A,0589,2D7      [1120]
850 DATA 237E,FE48,2806,FE4B,280A,390      [1070]
860 DATA 1846,2E2C,CD01,84C3,4884,399      [904]
870 DATA 2E29,CD01,84C3,4884,2E25,38B      [1777]
880 DATA C301,842E,34C3,0184,2E29,349      [1586]
890 DATA C301,842E,2CC3,0184,CDB6,46D      [2321]
900 DATA 832A,0589,237E,FE48,2804,34E      [1301]
910 DATA FE45,2008,2E01,CD01,84CD,3B9      [1363]
920 DATA 4884,2E01,C301,842E,04C3,338      [1821]
930 DATA 0184,2E29,CD01,84C3,1582,388      [1910]
940 DATA 2E21,C301,842E,37C3,0184,344      [1281]
950 DATA 2E38,C301,84CD,B683,2A05,3E3      [1525]
960 DATA 8923,7EFE,4528,062E,12CD,3A8      [1395]
970 DATA 0184,C92E,3ECD,0184,C348,417      [1672]
980 DATA 842E,27C3,0184,2E29,CD01,346      [1368]
990 DATA 842E,2CCD,0184,2E23,CD01,34F      [1827]
1000 DATA 84C3,4884,2EEC,CD01,842E,4AD     [1704]
1010 DATA DFCD,0184,2EEC,C301,842A,4BD     [1764]
1020 DATA 0589,237E,FE50,2814,FE43,3FA     [1232]
1030 DATA 2815,2E30,C301,84CD,4884,37C     [1159]
1040 DATA CD48,842E,32CD,0184,2E32,3AB     [1567]
1050 DATA C301,8423,7EFE,4828,EA18,459     [1331]
1060 DATA E32E,28C3,0184,CDB6,832A,4B1     [1174]
1070 DATA 0589,237E,FE45,2805,2E17,2E4     [1468]
1080 DATA C301,842E,3CCD,0184,C348,40F     [2554]
1090 DATA 842E,33C3,0184,2E29,CD01,352     [1834]
1100 DATA 842E,2BCD,0184,2E30,C301,351     [1542]
1110 DATA 842E,28CD,0184,2E30,C301,34E     [1505]
1120 DATA 842E,00CD,0184,2E00,CD01,300     [1087]
1130 DATA 842E,00CD,0184,2E00,C301,2F6     [956]
1140 DATA 8421,1289,187B,2117,8918,2AC     [1370]
1150 DATA 7621,1C89,1871,2121,8918,2A8     [1497]
1160 DATA 6C21,2689,1867,212B,8918,2A8     [1533]
1170 DATA 6221,3189,185D,2135,8918,2A9     [1485]
1180 DATA 5821,3C89,1853,2141,8918,2AC     [1293]
1190 DATA 4E21,4689,1849,215A,8918,2BB     [1420]
1200 DATA 4421,6989,183F,2170,8918,2E0     [1107]
1210 DATA 3A21,7889,1835,217C,8918,2E7     [1681]
1220 DATA 3021,8689,182B,2192,8918,2F7     [1473]
1230 DATA 2621,9D89,1821,21A1,8918,309     [1136]
1240 DATA 1C21,A689,1817,21AC,8918,309     [1950]
1250 DATA 1221,BA89,180D,21C6,8918,323     [1521]
1260 DATA 0821,D289,1803,21D9,893E,360     [845]
1270 DATA 0132,0489,E52A,0589,2207,286     [1621]
```

Listing. "SPRECHI"

```
1280 DATA 89E1,2205,897E,FE2E,2808,3F4     [1835]
1290 DATA C332,81CD,4884,18F3,3E00,458     [2046]
1300 DATA 3204,892A,0789,2205,89C9,2F2     [2220]
1310 DATA 3A09,89B7,C83C,3209,89FE,449     [1931]
1320 DATA FF28,14CD,5084,3A03,8937,3D9     [1196]
1330 DATA 3FD6,08D8,3203,896F,2608,350     [1289]
1340 DATA C30F,843A,0189,C602,6F26,377     [1800]
1350 DATA 20CD,0F84,3A00,89E6,0F47,37F     [1793]
1360 DATA 3A00,89E6,F00F,0F0F,0FFE,3D3     [2012]
1370 DATA 0338,02D6,030F,0F0F,0FB0,202     [1338]
1380 DATA 6F26,1018,0E26,0001,F1F8,2DB     [1973]
1390 DATA ED78,E680,FE80,CA06,8400,59D     [1030]
1400 DATA 01F0,F87D,ED79,01F1,F87C,632     [1347]
1410 DATA F640,ED79,7CE6,38ED,79C9,665     [1192]
1420 DATA 3A03,896F,2608,CD0F,843A,2FD     [1613]
1430 DATA 0289,6F26,18CD,0F84,3A01,2D3     [956]
1440 DATA 896F,2620,CD0F,843A,0089,361     [1335]
1450 DATA 6F26,10C3,0F84,2A05,8923,2D6     [2097]
1460 DATA 2205,89C9,2E98,26E8,2200,36F     [1951]
1470 DATA 892E,7F26,5022,0289,18C4,335     [2012]
1480 DATA C5D5,E5CD,5ABB,E1D1,C1C9,79D     [2205]
1490 'Teil 2                               [209]
1500 DATA 35072,515                        [690]
1510 DATA 0000,0000,98E8,7F50,0000,24F     [1515]
1520 DATA 0000,0000,0000,0000,0000,000     [1481]
1530 DATA 0000,4E55,4C4C,2E45,494E,245     [1286]
1540 DATA 532E,5A57,4549,2E44,5245,2C9     [2037]
1550 DATA 492E,5649,4552,2E46,5545,2BB     [2214]
1560 DATA 4E46,2E53,4558,2E53,4945,2C1     [1904]
1570 DATA 4245,4E2E,4143,4854,2E4E,29F     [2078]
1580 DATA 4555,4E2E,414E,4655,4555,2DA     [1465]
1590 DATA 4552,554E,4753,5354,5249,316     [1717]
1600 DATA 4348,452E,4E55,4D4D,4552,2D2     [1347]
1610 DATA 4E5A,4549,4348,454E,2E44,2C6     [963]
1620 DATA 4F4C,4C41,522E,5052,4F5A,2F3     [1634]
1630 DATA 454E,542E,554E,442E,4150,2BB     [1939]
1640 DATA 4F53,5452,4F46,462E,4B4C,2E8     [1800]
1650 DATA 414D,4D45,5220,4155,462E,29C     [1637]
1660 DATA 4B4C,414D,4D45,5220,5A55,2D8     [1741]
1670 DATA 2E4D,414C,2E50,4C55,532E,2A8     [2216]
1680 DATA 4D49,4E55,532E,4745,5445,2DF     [1464]
1690 DATA 494C,5420,4455,5243,482E,2AD     [1990]
1700 DATA 444F,5050,454C,5055,4E4B,302     [2209]
1710 DATA 542E,4B4C,4549,4E45,5220,2AC     [2369]
1720 DATA 414C,532E,474C,4549,4348,2BA     [1690]
1730 DATA 2E47,524F,4553,5345,5220,2B8     [2289]
1740 DATA 414C,532E,202E,2C3B,3A21,21E     [1744]
1750 DATA 3F21,F689,01FA,89C3,D1BC,5B3     [1321]
1760 DATA 0000,0000,298A,C300,80C3,2B9     [826]
1770 DATA 0080,C383,8AC3,838A,C394,577     [1182]
1780 DATA 8AC3,948A,C3A9,8AC3,A98A,657     [936]
1790 DATA C3B1,8AC3,B18A,C3CA,8AC3,6D6     [1003]
1800 DATA CA8A,C3D2,8AC3,D28A,C3E7,73C     [641]
1810 DATA 8A53,41D9,5341,C756,4F4C,443     [1578]
1820 DATA 554D,C556,4FCC,4152,5449,408     [2192]
1830 DATA 4B55,4C41,5449,4FCE,4152,37A     [1761]
1840 DATA D446,494C,5445,5246,5245,377     [1258]
1850 DATA 5155,454E,DA46,46D1,4745,3FC     [2057]
1860 DATA 5343,4857,494E,4449,474B,2EB     [2038]
1870 DATA 4549,D453,5045,45C4,544F,3F6     [1230]
1880 DATA 4E48,4F45,48C5,54C8,4E4F,3F0     [2418]
1890 DATA 524D,41CC,4ED2,4D4F,44C5,471     [1705]
1900 DATA 00FE,01C0,7BE6,0F47,3A58,408     [1221]
1910 DATA 84E6,F0B0,3258,84C9,FE01,5E0     [2058]
1920 DATA C07B,E607,1717,1717,473A,305     [1413]
1930 DATA 5884,E60F,B032,5884,C9FE,556     [2030]
1940 DATA 01C0,7B32,5384,C9FE,01C0,4CD     [1412]
1950 DATA 7BE6,0FCB,27CB,27CB,27CB,511     [2566]
1960 DATA 2747,3A51,84E6,0FB0,3251,3A5     [1649]
1970 DATA 84C9,FE01,C07B,325A,84C9,560     [1206]
1980 DATA 3E98,3251,843E,E832,5384,40C     [1053]
1990 DATA 3E7F,3258,843E,5032,5A84,369     [1542]
2000 DATA C9FE,01C0,7BB7,C8E6,03CB,636     [2986]
2010 DATA 27CB,27CB,27CB,27CB,27CB,4BA     [2077]
2020 DATA 2747,3A48,80B0,3248,80C9,3E3     [1490]
2030 'Teil 3                               [208]
2040 DATA 36608,117                        [565]
2050 DATA DD81,F982,3F83,4483,4983,52E     [1380]
2060 DATA 4E83,5383,5883,5D83,6283,447     [2403]
2070 DATA 6783,6C83,F982,7183,F982,5C3     [1688]
2080 DATA 7683,0D83,1283,1783,1C83,357     [869]
2090 DATA 2183,2683,2B83,3083,3583,366     [1253]
2100 DATA 3A83,7B83,F982,8083,8583,541     [2160]
2110 DATA 8A83,F982,F982,7E81,E281,665     [1414]
2120 DATA E781,0682,9D81,0B82,1082,42D     [2249]
2130 DATA 1582,1A82,3782,3C82,4482,370     [1088]
2140 DATA 4982,4E82,5382,6D82,7282,453     [1644]
2150 DATA 8482,9382,BD82,C282,0B82,52B     [1276]
2160 DATA DB82,E082,3782,EF82,0000,4E9     [1017]
```

Listing. "SPRECHI"


```
10 ' SPRECHI-Testprogramm "HALLO"          [2153]
20 '                                       [117]
30 ' 1987 by J.H. und J-M.M.               [1209]
40 '                                       [117]
50 daa=&F8F0:dab=daa+1:OUT daa+2,207:OUT d [4745]
aa+2,0:OUT dab+2,207:OUT dab+2,128
60 a=128:b=24:GOSUB 160                    [1380]
70 a=128:b=0:GOSUB 160                     [1708]
80 a=95:b=24:GOSUB 160                     [1052]
90 a=229:b=32:GOSUB 150                    [913]
100 a=127:b=16:GOSUB 150                   [1642]
110 a=40:b=8:GOSUB 150                     [1063]
120 FOR i=1 TO 15:READ a$:a=VAL("&"+a$)    [1039]
130 b=0:GOSUB 150:NEXT i:END               [2823]
140 DATA 2c,d,21,21,12,0,0,0,31,16,30,a,9f [2165]
,ec,00
150 IF INP (dab)=128 THEN 150              [1407]
160 OUT daa,a:OUT dab,b OR 64:OUT dab,b AN [2474]
D 56:RETURN
```

```
10 ' Demonstration von SPRECHI             [1927]
20 '                                       [117]
30 ' 1987 by J.H. und J-M.M.               [1209]
40 '                                       [117]
50 IF PEEK (&8000)<>254 THEN MEMORY &7FFF: [4657]
LOAD"sprechi.rsx":CALL &89ED
60 KEY 0,"¦SAY,"+CHR$(34):KEY 1,"¦TH,"     [1999]
70 KEY 2,"¦FFQ,":KEY 3,"¦SPEED,"           [1966]
80 KEY 4,"¦ART,":¦SPEED,12                 [1196]
90 MODE 0:PRINT" Demo fuer Sprechi"         [4221]
100 ¦TH,80:¦FFQ,232:¦ART,7                 [2053]
110 a$="HERZLICH WILKOMMEN ZU SPRECHI, DEM  [6958]
 HAATWAER SSUENTESSEISER FUER IHREN TSE PE
 TSE.":¦SAY,@a$
120 a$="DIESER SPRACH SSUENTESSEISER BIETE  [10344]
T VON HAUS AUS EINE MENGE VON MOEGLICHKEIT
EN, DIE WIR MIT UNSEREM PROGRAM VERSUCHT H
ABEN, ZUGENGLICH ZU MACHEN.":¦SAY,@a$
130 a$="DER TSCHIP ERLAUBT DEM BENUTZER EI  [8568]
NE MENGE VON EINSCHTELLMOEOEGLICHKEITEN, M
IT DENEN DIE SPRACHE STARK BEEINFLUSST WER
DEN KANN.  ":¦SAY,@a$
140 a$="MAN KANN ZUM BEISPIEL EINEN ALTEN   [7352]
MANN SPRECHEN LASSEN, WAS SICH ETWA SO AN
HOEREN KOENNTE.":¦SAY,@a$
150 ¦TH,40:¦FFQ,227:¦SPEED,10              [1903]
160 a$="SAG MAL ELSE, HAST DU MEINE REUMAT  [3009]
ABLETTEN GESEHEN?":¦SAY,@a$
170 ¦TH,80:¦FFQ,232:a$=" SIE KOENNTE DANN  [4187]
ANTWORTEN.":¦SAY,@a$
180 ¦SPEED,13:¦TH,200:¦FFQ,231             [3199]
190 a$="     ABER SCHATZ, DIE HABE ICH DIR  [6817]
 DOCH IN DEINEN NACHT TISCH GELEEKT.":¦SAY
,@a$
200 :¦TH,80:¦FFQ,232:a$="    MITLERWEILE K  [8285]
OENNTE DANN EIN BETRUNKENER VORBEIKOMMEN,
DER UM EINEN SCHLUCK BITTET.":¦SAY,@a$
210 ¦SPEED,8:¦TH,60:¦FFQ,230:¦ART,3         [2870]
220 a$=" SAAGEN SIE MAAL, JUNGE FRAU, HABE  [7321]
N SIE NICHT EINEN KLEINEN SCHLUCK FUER?":¦
SAY,@a$:¦TH,200:a$="MICH?":¦SAY,@a$
230 ¦SPEED,12:¦TH,80:¦FFQ,232:¦ART,7       [2521]
240 a$="ABER SPRECHI KANN NICHT NUR":¦SAY,  [3979]
@a$:¦TH,200:x$="  HOCH":¦SAY,@x$
250 ¦TH,80:a$="  UND":¦SAY,@a$:¦TH,30:a$=" [4478]
  TIEF":¦SAY,@a$
260 ¦TH,80:a$="  SPRECHEN, SONDERN AUCH":¦  [8238]
SAY,@a$:¦VOL,4:a$="LEISE":¦SAY,@a$:¦VOL,15
:a$="LAUT":¦SAY,@a$:¦VOL,12:a$="SCHNELL":¦
SAY,@a$
270 ¦SPEED,15:a$="ICH KANN SOGGAAR  SCHNAE  [4516]
LLER  SCHPRECHAEAENN.":¦SAY,@a$
280 ¦SPEED,4:a$="ODER SEHR LANGSAM.":¦SAY,  [3311]
@a$
```

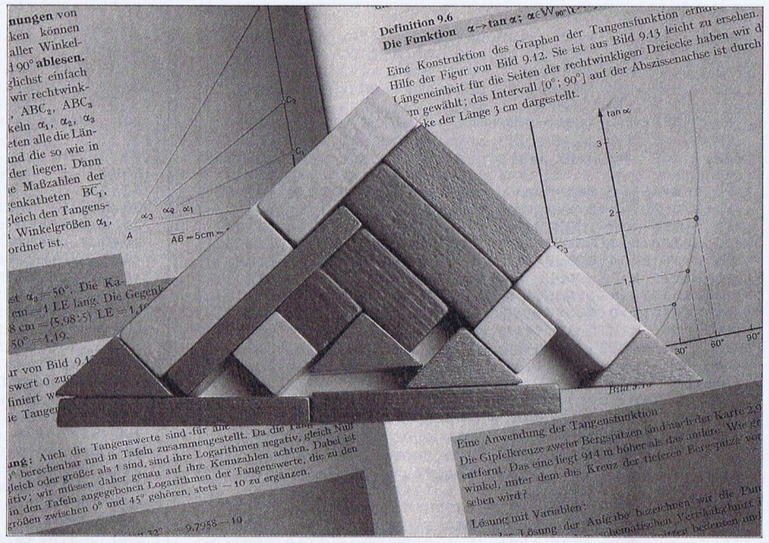
Listing. "SPRECHI"

```
20 ' Ueber Sieben Bruecken ...             [852]
30 '                                       [117]
40 ' 1987 by J. Heimsoth und               [1244]
50 '          J-M. Maczewski               [1357]
60 '                                       [117]
70 DEFINT a-z:ON ERROR GOTO 650            [1520]
80 daa=&F8F0:dab=daa+1                     [1762]
90 OUT daa+2,207:OUT daa+2,0:OUT dab+2,207 [3203]
:OUT dab+2,128
100 a=128:b=24:GOSUB 230                   [1382]
110 a=128:b=0:GOSUB 230                    [1535]
120 a=95:b=24:GOSUB 230                    [1070]
130 a=230:b=32:GOSUB 220                   [1311]
140 a=136:b=16:GOSUB 220                   [1392]
150 a=125:b=8:GOSUB 220                    [1576]
160 FOR i=1 TO 6000:READ a                 [1005]
170 IF a>63 THEN GOSUB 240:GOTO 190        [1739]
180 b=0:GOSUB 220                          [1796]
190 NEXT i                                 [375]
200 DATA 0,0,0,0,0                         [936]
210 END                                    [110]
220 IF INP (dab)=128 THEN 220              [1417]
230 OUT daa,a:OUT dab,b OR 64:OUT dab,b AN [2474]
D 56:RETURN
240 b=8:READ a:t=a:c=-62500/(a/7)+4096:a=V [4601]
AL("&X"+MID$(BIN$(c,12),2,8))
250 GOSUB 220:SOUND 129,62500/t,500:SOUND  [5142]
132,62500/(t/3),500:RETURN
260 READ a$:a=VAL("&"+a$):GOSUB 220        [846]
270 ' Ueber                                [179]
280 DATA 64,349,&3c,&2a,&2d,&3a,&1f        [1508]
290 ' sieben                               [445]
300 DATA 64,392,&30,&1,&1,&1,&24,&3A,&38   [988]
310 ' Bruecken                             [1072]
320 DATA 64,440,&24,&1e,&3c,&29,&29,&3A,&3 [2106]
8
330 '        musst          Du             [1215]
340 DATA &37,&13,&30,&28,    64,349,&17    [775]
350 ' geh'n                                [757]
360 DATA 64,392,&29,&2,&2,&38,&38,0,0,0,0, [1430]
0,0,0
370 ' sieben                               [445]
380 DATA &30,&1,&1,&24,&3A,&38             [1298]
390 ' dunkle                               [1307]
400 DATA 64,440,&25,&13,&13,&38,64,523,&29 [2784]
,&20,&0a
410 ' Jahre                                [170]
420 DATA 64,587,4,&d,&d,&d,&d,64,523,&1f,& [1894]
3e,&3e
430 ' uebersteh'n                          [479]
440 DATA 64,440,&3c,&2a,64,349,&2d,&3a,&1f [3106]
,64,392,&32,&28,2,2,&38,&38,&38,0,0,0,0,0,
0
450 ' siebenmal                            [1010]
460 DATA &30,&1,&1,&24,&3A,&38,&37,&d,&d,& [2659]
21,0,0
470 ' wirst                                [245]
480 DATA &23,1,&1f,&30,&28                 [849]
490 ' Du                                   [316]
500 DATA 64,440,&16,&16,0                  [1121]
510 '      die                Asche        [1239]
520 DATA &28,6,6,0,   &d,&32,64,330,&3a,&3a [1622]
,0
530 ' sein                                 [331]
540 DATA 64,294,&30,&d,3,&38,0,0,0,0,0     [990]
550 ' aber                                 [433]
560 DATA 64,392,&d,&d,64,440,&24,&3a,&1f   [1019]
570 ' einmal                               [1246]
580 DATA 64,466,&d,&6,&38,&37,&d,&21,0     [1695]
590 '       auch           der             [790]
600 DATA &d,&17,&2c,0,  64,523,&2d,2,&1f   [1163]
610 ' helle                                [280]
620 DATA 64,440,&2c,&3a,&3a,&3a,64,392,&3a [3258]
,&3a,64,350,&20,&3e,&3e
630 ' Schein.                              [906]
640 DATA &32,&d,3,3,&38,0,0,0,0            [602]
650 RUN                                    [243]
```

Listing. "SPRECHI"

Dreiecks-verhältnisse

Grafische Mathematik oder mathematische Grafik oder...?



Daß die Mathematik die treibende Kraft beim Computer ist, dürfte wohl jedem klar sein. Daß Mathematik das Hauptelement eines jeden Apfelmännchens ist, wissen viele. Daß man sich mit einem bißchen Mathematik schon wunderschöne Grafiken von seinem CPC erstellen lassen kann, wollen wir mit diesem kleinen Artikel beweisen. Es handelt sich hierbei um die grafische Berechnung eines PASCAL'schen Dreiecks. Nein, verehrte Leserinnen und Leser, Sie brauchen nicht gleich weiterzublättern, auch wenn sich Ihre mathematischen Kenntnisse auf das Errechnen des Gesamtpreises eines Big Mac's und einer Cola beschränken sollten. Die Formel zu diesem Dreieck ist recht schnell verstanden und dann soll doch möglichst mit Spaß und Freude selbst herumprobiert werden. Also stürzen wir uns ins Vergnügen und in die Welt der Grafik.

Die von uns allen so gefürchteten, von Lehrern zu Prüfungszwecken oft mißbrauchten, binomischen Formeln enthalten, wie wir hoffentlich alle noch wissen, Beizahlen, die (ACHTUNG!!) 'Binominalkoeffizienten'. Diese 'Binomikos' können in der Form eines Dreieckes angeordnet werden. Auf diese Weise entsteht also ein Pascalsch's Dreieck. Die Berechnung eines solchen Dreieckes ist anhand von zwei Regeln selbst nachvollziehbar. Die beiden Regeln lauten wie folgt:

1. Am linken und am rechten Rand des Dreiecks steht in jeder Zeile jeweils eine 1, und

2. jede andere Zahl ist die Summe der beiden schräg darüber stehenden Zahlen.

Damit das ganze nicht zu trocken wird, schauen wir es uns einmal in natura an:

Die Spitze des Dreiecks besteht, da kein rechter oder linker Rand vorhanden ist, aus einer 1. Eine Stufe tiefer kommen für rechts und links ebenfalls jeweils eine 1, aber immer eine Spalte versetzt:

```
      . 1 .
    . 1 . 1 .
  . 1 . ? . 1 .
. 1 . ? . ? . 1 .
```

Die 1 wird jeweils rechts und links vom Anfangspunkt (Dreiecksspitze) gesetzt, so daß sich die Dreieckskonstellation ergibt.

Dies war also die erste Regel, die wir zu beherzigen haben, kommen wir jetzt zur zweiten. Hier wollen wir die Zahlen, die im Augenblick noch durch die Fragezeichen repräsentiert werden, durch die Summe der beiden schrägoben stehenden Zahlen erzeugen. Nichts einfacher als das:

```
.1.                     Die Spitze des 'Eisberges'.
.1 . 1 .                Die nächste Reihe bilden die beiden 'Seiteneinsen',
. 1 . 2 . 1 .           die zusammen die 2 ergeben, mit der linken und rechten 1 wiederum kommt eine
. 1 . 3 . 3 . 1 .       3 heraus, die neuen Additionen ergeben dann die 4,
. 1 . 4 . 6 . 4 . 1 . 4 die 6, die 10 usw., die Reihe läßt sich beliebig
. 1 . 5 .10 .10 . 5 . 1 . lange fortsetzen.
```

Und wo bleibt der Computer?

Der kommt jetzt an die Reihe. Wie wir gemerkt haben, läßt sich die Berechnungsreihe immer weiter fortführen, und theoretisch könnten wir auf einem Block Millimeterpapier alle errechneten Punkte eintragen. Aber wir brauchen uns nur einmal den benötigten Zeitaufwand vorzustellen, und schon ist es vorbei mit unserem Elan. Jedenfalls bis uns dieses komische Brett mit den vielen Tasten einfällt, das ja deshalb gekauft wurde, um uns allerhand Rechenarbeit abzunehmen, falls man die BUSTOUT-Diskette gerade mal herausnimmt.
Wir müssen nur die geeignete Form finden, um den schlauen Gesellen auch zum Rechnen animieren zu können. Überlegen wir uns also erst einmal die Formel:
Wir wissen nun, daß eine Zahl an der Position X einer Zeile die Summe der beiden Zahlen ist, die an der der Position X und X-1 der vorherigen Zeile stehen. (Wer's nicht glaubt, soll sich noch einmal die obigen Beispiele zu Gemüte führen.)
Wir verdeutlichen dies, indem wir exemplarisch zwei Zeilen X und Y herausnehmen, wobei Y schon errechnet sein muß:

```
y(1)     y(2)     y(3)     y(4)     y(z-1)
  .   1   .   4   .   6   .   4   .   1   .
.   1   .   5   .  10   .  10   .   5   .   1   .
x(1)     x(2)     x(3)     x(4)     x(s)     x(z)
```

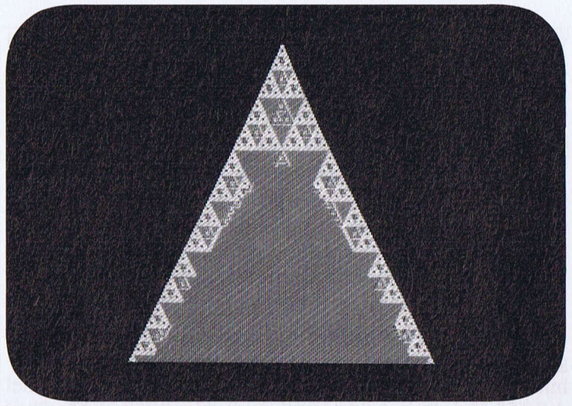



Abb. 1: Die Grafik wurde mit der ursprünglichen Formel aufgebaut. Hier sieht man die durch die ungenaue Berechnung entstandene freie Fläche...

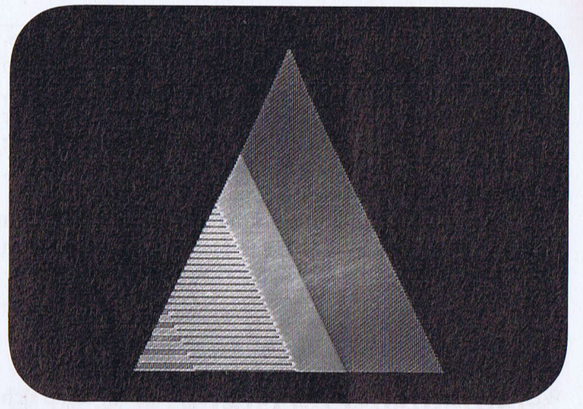


Abb. 2: ...während sich schon nach einer kleinen Änderung ein völlig anderes Bild zeigt.

Will man nun die Zahl x(s) errechnen, wobei die Formel 1 < s < z gelten muß, so addiert man y(s) und y(s-1). Dies gilt es nur noch, in geeignete Variablen für den Rechner umzusetzen.

Und wo bleibt die Grafik?

Gemach, gemach! Zuerst ist es doch wichtig, die Koordinaten zu haben, an denen der Rechner seine Grafik zeichnen darf, und die hat er mit oben genannter Rechenformel bekommen. Nun teilen wir ihm einfach drei kleine Forderungen mit, die an die errechneten Zahlen geknüpft werden, nämlich:

1. Durch 4 teilbare Zahlen sollen rot dargestellt werden,
2. gerade Zahlen bekommen die Farbe blau, und zu guter Letzt
3. alle ungeraden Zahlen werden zu gelben Pixeln.

So, das war eine ganze Menge ziemlich trockenen Stoffes. Jetzt ist es Zeit für eine kleine Erfrischung und für unser kleines Programm:

```
10 MODE 1:z=0:INK 0,0:INK 2,2:DIM x(999),y(999)
20 FOR s=1 TO z:y(s)=x(s):NEXT
30 z=z+1:x(z)=1:FOR s=2 TO 2*z-1:x(s/2)=y(s/2)+y(s/2-1)
40 IF x(s/2)/4=INT(x(s/2)/4) THEN C=3:GOTO 60
50 IF x(s/2)/2=INT(x(s/2)/2) THEN C=2 ELSE c=1
60 FOR j=1 TO 2:PLOT 320+z*2-s*2,400-z*4-j*3,c:NEXT:NEX
T:GOTO 20
```

Und das soll alles sein?

höre ich Sie schon entgeistert fragen. Schauen wir uns doch einmal das Programm genauer an, zuerst die Liste der verwendeten Variablen:

- z → die Anzahl der Zahlen in der gerade aktuellen Zeile; sowie die gerade aktuelle Zeile;
- x(s/2) → die Zahl x an der Position s/2 der gerade aktuellen Zeile
- y(s/2) → die Zahl y an der Position s/2 der vorherigen Zeile;
- c → die Farbe des Punktes, der gerade gesetzt wird oder wurde;
- j → bildet eine Schleife, die zur vergrößerten Darstellung des Dreiecks dient.

Soweit also die Variablen, kommen wir nun zur Programmbeschreibung:

Zeile / Ausführung
- 10 Dimensionierung der Felder x und y
- 20 die zu Ende berechnete Zeile x wird in das Feld der vorherigen Zeile y geschrieben;
- 30 Beginn der Schleife zur Berechnung einer neuen Zeile;
- 40 Überprüfung, ob die gerade neu errechnete Zahl durch 4 teilbar ist;
- 50 Überprüfung, ob die gerade neu errechnete Zahl gerade oder ungerade ist;
- 60 Setzen des Grafikpixels (durch die j Schleife werden 2 Pixel gesetzt);

Wie Sie sehen können, reichen tatsächlich ein paar Zeilen aus, um eine mathematische Formel in eine hübsche Computergrafik umzuwandeln.

Und wieso wird es ein Dreieck?

Dies ist einzig und allein die Wirkung der mathematischen Formel. Schauen wir uns die Variablen etwas genauer an.

Die Variablen s und z werden, außer zur Berechnung der Zahlen, auch zur Positionierung des Dreiecks auf dem Bildschirm genutzt (Abb.3). Zu x werden 2*z addiert, der Punkt, den man erhält, liegt damit auf dem rechten Rand des Dreiecks. Von diesem Punkt, nennen wir ihn der Einfachheit halber p, werden innerhalb der Schleife s 2*z abgezogen. Am Anfang der Schleife hat 2*s dann den Wert 4, d.h., der Punkt, der innerhalb der Schleife als erstes gesetzt wird, liegt vier Pixel weiter links als p. Dies hat aber keine weiteren Auswirkungen, da die Schleife zu Anfang immer den Wert 2 (2*s hat also den Wert 4) hat,

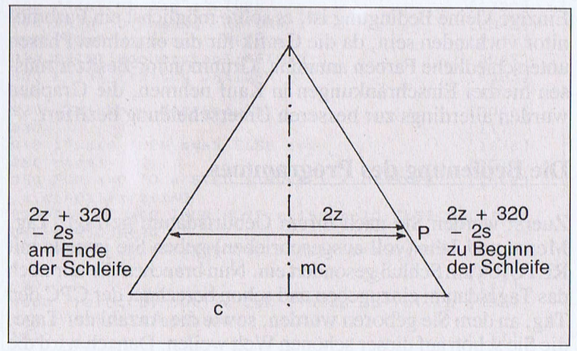


Abb. 3: Die Verhältnisse der Variablen im Dreieck zueinander.

und diese Verschiebung in jeder Zeile auftritt. Diese Verschiebung tritt auch jeweils am Ende der Schleife auf, wenn s den Wert 2*z-1 hat. Die Variable s wird dann mit 2 multipliziert, somit befindet sich der Punkt zwei Pixel weiter rechts als der Punkt, den man durch die Rechnung 320-2*z erhält. 320-2*z wäre dann der Spiegelpunkt von p, wenn mc (Mittelsenkrechte, siehe Abb.3) die Spiegelachse wäre.
Die beiden Ungenauigkeiten, die durch die Schleife bedingt sind, gleichen sich bis auf zwei Pixel aus, diese fallen jedoch so wenig ins Gewicht, daß Sie keinerlei optische Auswirkungen haben.

Und wie sieht es auf dem Bildschirm aus?

Wenn Sie das Programm korrekt abgetippt haben, sollte sich auf dem Monitor ein Dreieck abzeichnen, wie in Abb.1 mit der Spitze nach oben zeigend.
Dieses Dreieck besteht aus drei kleineren Dreiecken, die wiederum in drei kleinere Dreiecke aufgeteilt sind, die wiederum in...
Wenn der Bildschirm zu ungefähr einem Drittel mit dem Dreieck gefüllt ist, sind nun grobe Unregelmäßigkeiten in dem folgenden Abschnitt zu erkennen. Eine nach unten breiter werdende Fläche des Dreiecks wird ausschließlich rot gezeichnet. Dies hat seine Ursache darin, daß die errechneten Zahlen zu groß und nicht mehr bis zur letzten Stelle ausgerechnet werden, wie z.B. die Zahl 1.58726 E+09. Diese Zahl hat am Ende vier Nullen. Im Programm kommt es aber auf den Unterschied zwischen geraden und durch 4 teilbaren Zahlen an, d.h., die letzten beiden Stellen werden benötigt. Da diese bei zu großen Zahlen aber immer 0 sind, werden diese Zahlen als durch 4 teilbar angesehen und rot gezeichnet.

Da unser Programm aber nur als Anregung gedacht ist, um Sie zur Umsetzung eigener Ideen zu animieren, bleibt es Ihnen überlassen, was Sie zur Verbesserung einsetzen wollen. Anhand unserer Erklärungen sollte dies ein Leichtes sein, und vielleicht macht so manchem Schüler die Mathematik wieder Spaß, wenn man sehen kann, was der Computer daraus macht.

Die einfachste Änderung, die man vornehmen kann, ist die Vergrößerung der Vergleichszahl 4, das Ergebnis kann man in Abbildung 2 begutachten. Aber auch das Umstellen der Formel durch weitere Berechnungen bringt ganz erstaunliche Grafiken zustande.

Ein zwischendurch auftauchendes 'Overflow' sollte deshalb niemanden von weiteren Versuchen abhalten.

Und daß unser CPC mit seinen ausgefeilten Grafikbefehlen geradezu auserkoren ist, aus komplizierten, fast unverständlichen Formeln fantastische Grafiken zu erstellen, erleichtert die Entscheidung doch ungemein, oder?

(Jens Budinger/jb)

Rhythm and Soul

Mini-Biorhythmus mit grafischer Darstellung

In der heutigen Zeit, in der Gesundheit und Wohlbefinden so groß geschrieben werden, wie schon lange nicht mehr, werden Biorhythmus-Programme für alle Rechnertypen angeboten, mit den unterschiedlichsten Optionen und mit einer Menge Speicherverbrauch. Das dies auch anders geht, zeigt unser kleines Biorhythmus-Programm von 4 KByte Länge.

Einzige kleine Bedingung ist, es sollte möglichst ein Farbmonitor vorhanden sein, da die Grafik für die einzelnen Phasen unterschiedliche Farben annimmt. Grünmonitor-Besitzer müssen hierbei Einschränkungen in Kauf nehmen, die Graphen wurden allerdings zur besseren Unterscheidung beziffert.

Die Bedienung des Programmes

Zuerst werden Sie nach Ihrem Geburtsdatum gefragt, Tag, Monat und Jahr (voll ausgeschrieben) geben Sie jeweils mit RETURN am Schluß gesondert ein. Nun brauchen Sie nur noch das Tagesdatum einzugeben und schon berechnet der CPC den Tag, an dem Sie geboren wurden, sowie die Anzahl der Tage, die Sie schon auf dieser schönen Welt weilen. Danach wird die Tagesanalyse als Säulendiagramm dargestellt. Nach Betätigen

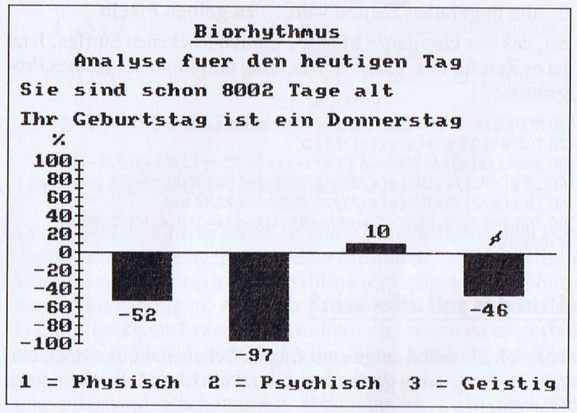


Abb. 1 zeigt Ihre körperliche Form als Balkendiagramm.

einer Taste wird der Verlauf Ihrer körperlichen und geistigen Form für die nächsten 31 Tage grafisch dargestellt. Durch Drücken der Ctrl H Taste wird zu einer Hardcopyfunktion gesprungen, die diesem Programm jedoch nicht nicht angeschlossen wurde. Ab Zeile 1000 kann jeder seine Hardcopy-Routine für den eigenen Drucker nachtragen, damit dürften alle Druckerwünsche erfüllt sein. Falls Sie gerne eigene Programme zur Berechnung der physischen, geistigen und psychischen Form erstellen wollen, können Sie die dazu benötigten Formeln aus dem Lexikon oder Medizinbüchern entnehmen.

(Dirk Haltermann/jb)

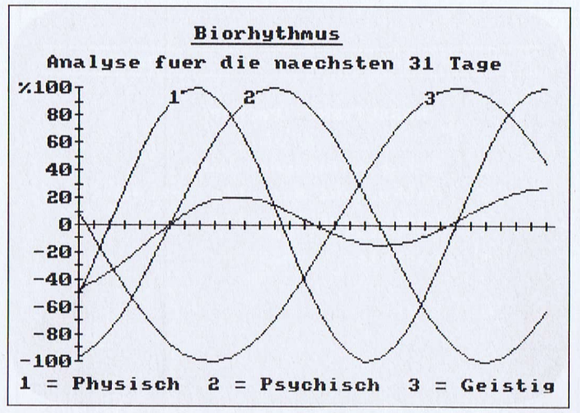


Abb. 2 mit den durch Zahlen kenntlich gemachten Kurven.

für 464-664-6128

```
100 '                                                    [117]
110 'Biorhythmus und Wochentagsbestimmung                [3788]
120 '                                                    [117]
130 '     1988 by Dirk Haltermann                        [747]
160 '                                                    [117]
170 MODE 1:INK 0,0:INK 1,26:INK 2,24:INK 3 [2718]
,18:BORDER 0:DEG
180 LOCATE 14,1:PRINT"Biorhythmus"                       [2094]
190 MOVE 204,380:DRAW 384,380,2                          [992]
200 PEN 3:LOCATE 1,3:PRINT"Geburtdatum:"                 [3446]
210 MOVE 0,348:DRAW 190,348,2                            [846]
220 PEN 2:LOCATE 1,5:PEN 1:INPUT"Tag  ";t:               [3254]
IF t>31 OR t<1 THEN 220
230 LOCATE 1,6:INPUT"Monat";m:IF m>12 OR m               [2678]
<1 THEN 230
240 LOCATE 1,7:INPUT"Jahr ";j:IF j<1582 TH               [3628]
EN 240
250 c=0:PEN 3:LOCATE 1,9:PRINT"Aktuelles D               [3512]
atum:":PEN 1
260 MOVE 0,252:DRAW 252,252                              [551]
270 LOCATE 1,11:INPUT"Tag  ";t1:IF t1>31 O               [3495]
R t1<1 THEN 270
280 LOCATE 1,12:INPUT"Monat";m1:IF m1>12 O               [3865]
R m1<1 THEN 280
290 LOCATE 1,22:PEN 1:PRINT TAB(11)"Hardco               [3571]
py Ctrl H"
300 LOCATE 5,24:PEN 1:PRINT"Weiter jeweils               [4810]
 KEY druecken"
310 LOCATE 1,13:INPUT"Jahr ";j1:IF j1<1582               [2758]
 THEN 310
320 '                                                    [117]
330 'Tagesanalyse                                        [980]
340 '                                                    [117]
350 WINDOW 1,40,3,25:CLS:z=40                            [1860]
360 PEN 3:LOCATE 5,1:PRINT"Analyse fuer de               [3528]
n heutigen Tag"
370 GOSUB 810:PRINT:PEN 1:PRINT"Sie sind s               [5006]
chon"f1-f"Tage alt"
380 tage=f1-f                                            [1022]
390 w=8+f+(INT(-f/7)*7):PRINT:PRINT"Ihr Ge               [3370]
burtstag ist ein ";
400 ON w GOSUB 840,850,860,870,880,890,900               [1789]
,840
410 ORIGIN 70,150                                        [516]
420 TAG:FOR x=-100 TO 100 STEP 20:MOVE -70               [5372]
,x+6:PRINT USING "####";x;:NEXT
430 MOVE -30,130:PRINT"%";:TAGOFF                        [2095]
440 MOVE 0,-100:DRAW 0,100,1:MOVE 0,0:DRAW               [2579]
 640,0
450 FOR x=-100 TO 100 STEP 10:MOVE -4,x:DR               [3531]
AW 4,x,1:NEXT
460 a=ROUND(SIN(tage/23*360)*100):GOSUB 94               [3018]
0
470 a=ROUND(SIN(tage/28*360)*100):GOSUB 94               [2366]
0
480 a=ROUND(SIN(tage/33*360)*100):GOSUB 94               [2693]
0
490 d=ROUND(d/3):'IF d<0 THEN s=-4 ELSE s=               [3310]
4
```

Listing "Rhythm and Soul"

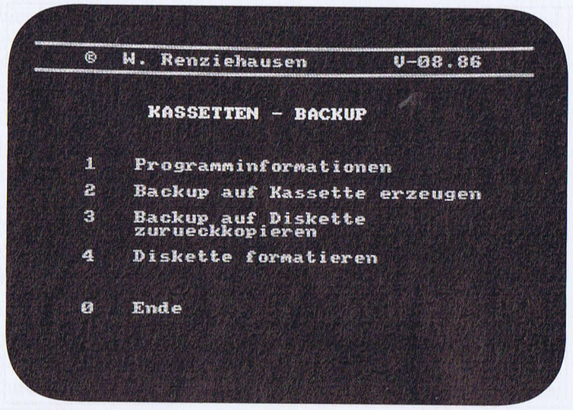
```
500 FOR x=z TO z+70 STEP 6:MOVE x,0:DRAW x               [6220]
,d,c:MOVE x+2,0:DRAW x+2,d:MOVE x+4,0:DRAW
 x+4,d:IF c=3 THEN c=0
510 c=c+1:NEXT:z=z+140                                   [1631]
520 TAG:IF d<0 THEN MOVE z-150,d-10 ELSE M               [3071]
OVE z-150,d+20
530 PRINT USING"####";d;:TAGOFF:IF d<0 THE               [5859]
N LOCATE 36,13:PRINT CHR$(186); ELSE LOCAT
E 36,15:PRINT CHR$(186);
540 LOCATE 1,23:PEN 1:PRINT"1 = Physisch                 [6575]
";:PEN 2:PRINT"2 = Psychisch  ";:PEN 3:PRI
NT"3 = Geistig";
550 a$=INKEY$:IF a$="" THEN 550 ELSE IF a$               [2613]
=CHR$(8) THEN GOSUB 1010
560 WINDOW #1,1,40,3,24:CLS#1                            [1920]
570 PEN 3:LOCATE 3,1:PRINT"Analyse fuer di               [3934]
e naechsten 31 Tage"
580 ORIGIN 70,182:DRAW 640,0,1:MOVE 0,-150               [1724]
:DRAW 0,150
590 FOR x=-150 TO 150 STEP 15:MOVE -4,x:DR               [4429]
AW 4,x:NEXT
600 FOR x=0 TO 570 STEP 18:MOVE x,-4:DRAW                [1890]
x,4:NEXT
610 TAG:y=-100:FOR x=-150 TO 150 STEP 30:M               [6360]
OVE -70,x+6:PRINT USING "####";y;:y=y+20:N
EXT:MOVER -64,0:PRINT"%";
620 MOVE 0,150*SIN(360*tage/23):c=tage:fla               [2824]
g=0
630 FOR x=0 TO 570 STEP 18:DRAW x,150*SIN(               [6635]
360*c/23),1:c=c+1:IF flag=0 AND INT(SIN(36
0*c/23)*150)=149 THEN MOVER -20,0:PEN 1:PR
INT"1";:flag=1
640 NEXT                                                 [350]
650 MOVE 0,150*SIN(360*tage/28):c=tage:fla               [2627]
g=0
660 FOR x=0 TO 570 STEP 18:DRAW x,150*SIN(               [8409]
360*c/28),2:c=c+1:IF flag=0 AND INT(150*SI
N(360*c/28))=149 THEN MOVER -20,0:PEN 2:PR
INT"2";:flag=1
670 NEXT                                                 [350]
680 MOVE 0,150*SIN(360*tage/33):c=tage:fla               [2332]
g=0
690 FOR x=0 TO 570 STEP 18:DRAW x,150*SIN(               [6586]
360*c/33),3:c=c+1:IF flag=0 AND INT(150*SI
N(360*c/33))=149 THEN MOVER -20,0:PEN 3:PR
INT"3";:flag=1
700 NEXT                                                 [350]
710 DEF FN d=(SIN(360*c/33)+SIN(360*c/23)+               [2783]
SIN(360*c/28))/3
720 c=tage:MOVE 0,150*FN d:s=1                           [1991]
730 FOR x=0 TO 570 STEP 18:DRAW x,150*FN d               [2786]
,s
740 IF s=3 THEN s=0                                      [1139]
750 s=s+1:c=c+1:NEXT:TAGOFF                              [2258]
760 a$=INKEY$:IF a$="" THEN 760 ELSE IF a$               [2975]
=CHR$(8) THEN GOSUB 1010
770 CLS:GOTO 170                                         [951]
780 '                                                    [117]
790 'Anzahl der Tage zwischen den Daten                  [2022]
800 '                                                    [117]
810 IF m<3 THEN f=365*j+t+31*(m-1)+INT((j-               [8863]
1)/4)-INT(0.75*INT((j-1)/100+1)) ELSE f=36
5*j+t+31*(m-1)-INT(0.4*m+2.3)+INT(j/4)-INT
(0.75*INT(j/100+1))
820 IF m1<3 THEN f1=365*j1+t1+31*(m1-1)+IN               [7559]
T((j1-1)/4)-INT(0.75*INT((j1-1)/100+1)) EL
SE f1=365*j1+t1+31*(m1-1)-INT(0.4*m1+2.3)+
INT(j1/4)-INT(0.75*INT(j1/100+1))
830 RETURN                                               [555]
840 PRINT"Samstag":RETURN                                [1948]
850 PRINT"Sonntag":RETURN                                [2406]
860 PRINT"Montag":RETURN                                 [1729]
870 PRINT"Dienstag":RETURN                               [2045]
880 PRINT"Mittwoch":RETURN                               [2171]
890 PRINT"Donnerstag":RETURN                             [3007]
900 PRINT"Freitag":RETURN                                [1683]
910 '                                                    [117]
920 'Zeichnen der Saeule                                 [2064]
930 '                                                    [117]
940 IF a<0 THEN s=-2 ELSE s=2                            [1678]
950 c=c+1                                                [701]
960 FOR x=0 TO a STEP s:MOVE z,x:DRAW z+70               [2852]
,x,c:NEXT:z=z+140
970 TAG:IF a<0 THEN MOVE z-150,a-10 ELSE M               [2795]
OVE z-150,a+20
980 PRINT USING"####";a;:TAGOFF:d=d+a:RETU               [2638]
RN
990 '                                                    [117]
1000 'Hardcopyroutine zum Einbau Aufruf: C               [2339]
trl H schon eingebaut
1010 '                                                   [117]
```

Listing "Rhythm and Soul"

Datensicherheit einmal anders

Schnell-Backup auf Kassette

Die Drei-Zoll-Disketten der CPC-Rechner sind für die Verwendung als Backup-Datenträger immer noch zu teuer. Wer ein 5,25" Laufwerk hat, wird dieses für die Datensicherung benutzen. Was machen jedoch die anderen CPC-Anwender?



Wer sich kostengünstig und schnell Sicherheitskopien anlegen möchte, ist mit unserem Backup-Programm bestens bedient.

Die Möglichkeit, Programme und Daten auf Kassette abzulegen, ist ein sehr preiswertes Verfahren. Nachteil: Die Ablage dauert lange. Es sind bereits einige Programme in verschiedenen Zeitschriften veröffentlicht worden. Die mir bekannten benötigen für die Sicherung einer Diskettenseite mindestens zwölf Minuten. Dies hier vorgestellte Programm speichert die 180 Kilobyte einer Diskettenseite auf acht Minuten Kassettenband in einem Durchgang ab.

Backup in acht Minuten

Auf einer C60-Kassette können sechs Diskettenseiten abgelegt werden. Selbst bei der höchsten Baudrate (3600 Baud – entspricht obigen acht Minuten) habe ich auf preiswerten Kassetten aus dem Kaufhaus (Stück circa 1,– DM) bisher keine Lesefehler gehabt.

Da mehrere Diskettenseiten auf eine Kassette passen und damit bei einer späteren Aktualisierung (überschreiben auf Kassette) nicht versehentlich der Anfang des folgenden Backups überschrieben wird, kann vom Programm, zum Abschluß des Backups, eine Bandlücke erzeugt werden.

Einzelne Programme aus einem Backup können von der Kassette nicht direkt eingelesen werden. Hierzu müssen die Daten komplett von Kassette auf Diskette zurückkopiert werden. Dies erfolgt durch das BACKUP-Programm auch in einem Durchgang. Zum Zurückkopieren muß eine formatierte Diskette eingelegt werden. Falls das Format der Diskette nicht dem Backup entspricht, wird die Diskette vom Programm entsprechend umformatiert.

Die Formatierroutine kann über das Menü auch einzeln verwendet werden. Mit ihr kann man Disketten im System- oder Datenformat formatieren. Zusätzlich kann man hier angeben, welche Spuren (Spuren: 0 bis 42) formatiert werden sollen. Es ist also möglich, daß neue Disketten nur teilweise formatiert werden (Stichwort: Kopierschutz).

(Werner Renziehausen/cd)

für 464-664-6128

```
10 'BACKUP -     WERNER RENZIEHAUSEN        [3325]
   8.86 (9363 BYTES)
20 '                                        [117]
30 DEFINT b-z:MEMORY &3000-1:MODE 1         [1683]
40 FOR adress=&3F00 TO &3F0F                [1505]
50   READ byte$:byte=VAL("&"+byte$):POKE a [3621]
dress,byte
60 NEXT                                     [350]
70 FOR adress=&3FE4 TO &402C                [1880]
80   READ byte$:byte=VAL("&"+byte$):POKE a [3621]
dress,byte
90 NEXT:baudvor$="3000":drive=0             [1944]
100 INK 1,18:INK 0,0:INK 2,26:PAPER#1,0:PE [2471]
N#1,1:BORDER 0
110 MODE 1:WINDOW#0,1,40,23,25:WINDOW#1,1, [3305]
40,1,25
120 CLS:CLS#1:PRINT#1,STRING$(40,154)       [2092]
130 LOCATE#1,5,2:PRINT#1,CHR$(164);"  W. R [5491]
enziehausen      V-08.86
140 PRINT#1,STRING$(40,154)                 [1037]
150 PEN#1,2:LOCATE#1,10,6:PRINT#1,"KASSETT [3083]
EN - BACKUP":PEN#1,1
160 LOCATE#1,5,10:PRINT#1,"1   Programminf [4079]
ormationen
170 LOCATE#1,5,12:PRINT#1,"2   Backup auf  [3049]
Kassette erzeugen
180 LOCATE#1,5,14:PRINT#1,"3   Backup auf  [2418]
Diskette
190 LOCATE#1,5,15:PRINT#1,"    zurueckkopi [2163]
eren
200 LOCATE#1,5,17:PRINT#1,"4   Diskette fo [3119]
rmatieren
210 LOCATE#1,5,21:PRINT#1,"0   Ende":CALL  [2428]
&BB03
220 a$=INKEY$:IF a$="" THEN 220             [1372]
230 IF ASC(a$)<48 OR ASC(a$)>52 THEN 220 E [3035]
LSE a=VAL(a$)
240 IF a=0 THEN 260                         [434]
250 ON a GOSUB 1740,560,710,2590:GOTO 100   [1362]
260 CLS#1:LOCATE#1,14,13:PRINT#1,"ENDE  J/ [1647]
N ?
270 a$=INKEY$:a$=UPPER$(a$):IF a$<>"J" AND [3386]
 a$<>"N" THEN 270
280 IF a$="N" THEN 100 ELSE MODE 2:INK 1,2 [3376]
6:MEMORY &A000:END
290 FOR trackstufe=0 TO 39 STEP 5           [1858]
300  IF a=2 THEN GOSUB 1000                 [1005]
310  FOR track=trackstufe TO trackstufe+4   [2973]
320   POKE &4031,track                      [640]
330   LOCATE#1,1,20:IF a=1 THEN PRINT#1,"E [5963]
inlesen von"; ELSE PRINT#1,"Schreiben auf"
;
340   PRINT#1," Diskette Spur ";track       [2951]
```

Listing "BACKUP"


```
350   FOR sektor=1 TO 9                     [1592]
360    adr=(track-trackstufe)*4608+(sektor [1911]
-1)*512+&4040
370    hexadr$=HEX$(adr,4)                  [1036]
380    hb=VAL("&"+LEFT$(hexadr$,2)):POKE & [3052]
4027,hb
390    lb=VAL("&"+RIGHT$(hexadr$,2)):POKE  [2809]
&4026,lb
400    POKE &4032,sektor+f:CALL &4010       [1673]
410   NEXT                                  [350]
420  NEXT:CLS:FOR zeit=1 TO 10000:NEXT:'La [4396]
ufwerkmotor abschalten
430  IF a=1 THEN GOSUB 850                 [1904]
440 NEXT                                   [350]
450 IF a=1 THEN GOSUB 1660                 [1755]
460 SOUND 1,100,50:RETURN                  [2374]
470 '                                      [117]
480 DATA 1e,00,16,00,0e,41,21,10,3f,df,0d, [2308]
3f,c9,52,c6,07
490 DATA cd,65,bc,21,6f,00                 [525]
500 DATA 3e,0a,cd,68,bc,c9,cd,6b,bc,21,40, [2860]
5a,06,10,11,00
510 DATA 30,cd,8c,bc,c9,cd,8f,bc,c9,21,3f, [2203]
40,11,11,5a,3e
520 DATA c5,cd,9e,bc,c9,00,21,2c,40,cd,d4, [2090]
bc,22,2d,40,79
530 DATA 32,2f,40,21,30,40,5e,23,56,23,4e, [2234]
21,40,40,df,2d
540 DATA 40,c0,84                          [202]
550 '                                      [117]
560 a=a-1:CLS#1:LOCATE#1,10,4:PRINT#1,CHR$ [4841]
(24)" BACKUP ERZEUGEN "CHR$(24)
570 GOSUB 1260:GOSUB 1380                  [1789]
580 z$="":GOSUB 810:LOCATE#1,1,13:PRINT#1, [5761]
"REC+PLAY, dann Eingabe
590 PRINT#1,:PRINT#1,"Diskettenseitenbezei [4750]
chng: ";STRING$(13,".");" "
600 LOCATE#1,27,15:INPUT#1,"",discseite$   [2333]
610 l=LEN(discseite$)                      [1104]
620 IF l>13 OR l=0 THEN LOCATE#1,27,15:PRI [4198]
NT#1, SPACE$(l):LOCATE#1,1,14:GOTO 590
630 PRINT#1,:PRINT#1,"Diskettenformat S/D: [3974]
";
640 IF drive=0 THEN ¦A ELSE ¦B             [1589]
650 f=PEEK(&A89F+64*drive)-1:CLS           [1895]
660 IF f=64 THEN f$="S":fz$="ystem" ELSE f [5278]
$="D":fz$="aten"
670 PRINT#1, "'";f$;"'";fz$                [1336]
680 POKE &403F,f:POKE &402C,&84            [1995]
690 GOTO 290                               [411]
700 '                                      [117]
710 a=a-1:CLS#1:LOCATE#1,7,4:PRINT#1,CHR$( [4954]
24)" BACKUP ZURUECKKOPIEREN "CHR$(24):GOSU
B 1260
720 z$="auf richtige Position gespult ":GO [3401]
SUB 810
730 LOCATE#1,1,13:PRINT#1,"¦   Diskette be [5757]
schreibbar schalten   ¦
740 PRINT#1,"¦ Format soll dem Backup ents [4136]
prechen ¦
750 LOCATE#1,1,16:PRINT#1,"PLAY, dann Tast [3855]
e druecken
760 CALL &BB06:POKE &402C,&85              [1303]
770 IF drive=0 THEN ¦A ELSE ¦B             [1589]
780 fd=PEEK(&A89F+64*drive)-1:FOR zeit=1 T [2647]
O 10000:NEXT:CLS
790 GOTO 290                               [411]
800 '                                      [117]
810 LOCATE#1,1,10:PRINT#1,"Diskette in Lau [5274]
fwerk ";CHR$(65+drive);" und Kassette
820 PRINT#1, z$;"einlegen                  [1884]
830 RETURN                                 [555]
840 '                                      [117]
850 'Einen Block - 22,5 kB - auf Band schr [2068]
eiben
860 '                                      [117]
870 LOCATE#1,1,20                          [588]
880 PRINT#1,"Schreiben auf Kassette  Block [4619]
";trackstufe/5+1
890 file$=LEFT$(discseite$,13)             [1140]
900 file$=file$+SPACE$(14-LEN(file$))+RIGH [4345]
T$(STR$(trackstufe/5+1),1)+f$
910 FOR adress=&9A40 TO &9A4F              [1729]
920  POKE adress,ASC(MID$(file$,adress-&9A [2396]
3F,1))
930 NEXT                                   [350]
940 IF trackstufe=0 THEN CALL &3FE4        [1705]
950 POKE &400C,&9E::CALL &4003             [1332]
960 IF trackstufe=35 THEN CALL &3FFF       [1731]
970 LOCATE#1,1,20:PRINT#1, SPACE$(40)      [2759]
980 RETURN                                 [555]
990 '                                      [117]
1000 'Einen Block - 22,5 kB - vom Band les [1366]
en
1010 '                                     [117]
1020 FOR adress=&9A40 TO &9A4F:POKE adress [2924]
,32:NEXT
1030 LOCATE#1,1,20                         [588]
1040 PRINT#1,"Einlesen von Kassette   Bloc [5838]
k";trackstufe/5+1
1050 POKE &400C,&A1:CALL &4003             [1330]
1060 file$=""                              [73]
1070 FOR adress=&9A40 TO &9A4F             [1729]
1080  file$=file$+CHR$(PEEK(adress))       [1179]
1090 NEXT                                  [350]
1100 fileanz$=LEFT$(file$,14)+"Block "+MID [3640]
$(file$,15,1)+" Format "+MID$(file$,16,1)
1110 IF " "+MID$(file$,15,1)<>STR$(trackst [4441]
ufe/5+1) THEN GOSUB 1170:GOTO 1050
1120 LOCATE#1,1,22:PRINT#1, fileanz$;"  ge [2964]
laden
1130 f=PEEK(&403F):LOCATE#1,1,20:PRINT#1,  [4811]
SPACE$(40)
1140 IF f<>fd THEN GOSUB 1530:GOTO 1140    [1858]
1150 RETURN                                [555]
1160 '                                     [117]
1170 'Fehlermeldung - falsches File gelade [1605]
n
1180 '                                     [117]
1190 SOUND 1,150                           [1031]
1200 LOCATE#1,1,22:PRINT#1, fileanz$;"  ge [3745]
funden";
1210 PRINT#1,"Kassette richtig spulen -  ' [4130]
W'  druecken
1220 a$=INKEY$:IF a$="" THEN 1220 ELSE a$= [3875]
UPPER$(a$):IF a$<>"W" THEN 1220
1230 LOCATE#1,1,22:PRINT#1, SPACE$(80)     [2417]
1240 RETURN                                [555]
1250 '                                     [117]
1260 'Laufwerk festlegen                   [2334]
1270 '                                     [117]
1280 LOCATE#1,1,7:PRINT#1,"Laufwerk  A/B : [4502]
";:PEN#1,2:PRINT#1, CHR$(65+drive):PEN#1,
1
1290 d$=INKEY$:IF d$="" THEN 1290          [1234]
1300 IF d$=CHR$(13) THEN d$=CHR$(65+drive) [2433]
1310 d$=UPPER$(d$):IF d$<>"A" AND d$<>"B"  [3017]
THEN 1290
1320 LOCATE#1,17,7:PRINT#1, d$:IF d$="A" T [3285]
HEN drive=0 ELSE drive=1
1330 POKE &4030,drive                      [1077]
1340 RETURN                                [555]
1350 '                                     [117]
1360 'Baudrate einstellen                  [1720]
1370 '                                     [117]
1380 LOCATE#1,1,8:PRINT#1,"Baudrate  1000. [4273]
..3600 : ";
1390 PEN#1,2:PRINT#1, baudvor$:PEN#1,1     [1597]
1400 baud$="":LOCATE#1,25,8                [1344]
1410 d$=INKEY$:IF d$="" THEN 1410          [1242]
1420 IF d$=CHR$(13) THEN baud$=baudvor$:LO [5425]
CATE#1,25,8:PRINT#1, baud$:GOTO 1450
1430 IF ASC(d$)<48 OR ASC(d$)>57 THEN 1410 [1541]
1440 baud$=baud$+d$:PRINT#1, d$;           [1505]
1450 IF LEN (baud$)<4 THEN 1410 ELSE baud= [3328]
VAL(baud$)
1460 IF baud<1000 OR baud>3600 THEN 1380   [1245]
1470 baud=333333/baud:baudvor$=baud$       [1870]
1480 POKE &3FE8,VAL("&"+RIGHT$(HEX$(baud,4 [1523]
),2))
1490 POKE &3FE9,VAL("&"+LEFT$(HEX$(baud,4) [3237]
,2))
1500 POKE &3FEB,INT(10/93*baud)            [1330]
1510 RETURN                                [555]
1520 '                                     [117]
1530 'Fehlermeldung - falsches Format      [1392]
1540 '                                     [117]
1550 SOUND 1,150                           [1031]
1560 LOCATE#1,1,23:PRINT#1, "Diskette ist  [3959]
falsch formatiert !
1570 PRINT#1,"Sie wird ins ";              [2577]
1580 IF f=64 THEN PRINT#1,"System"; ELSE P [3802]
RINT#1,"Daten";
1590 PRINT#1,"-Format umformatiert";:PRINT [4578]
#1,"        .....  bitte warten";
```

Listing "BACKUP"

```
1600 GOSUB 2440:' Umformatieren                [1573]
1610 LOCATE#1,1,23:PRINT#1, SPACE$(110)        [1762]
1620 IF drive=0 THEN |A ELSE |B                [1589]
1630 fd=PEEK(&A89F+64*drive)-1:CLS             [1296]
1640 RETURN                                    [555]
1650 '                                         [117]
1660 'Aufnahmeluecke erzeugen                  [944]
1670 '                                         [117]
1680 CLS:SOUND 1,150:INPUT"Aufnahmeluecke      [3738]
 0...30 Sekunden: ",sek
1690 IF sek<0 OR sek>30 THEN 1680              [1045]
1700 IF sek=0 THEN 1730                        [459]
1710 sek=sek-2                                 [498]
1720 CALL &BC6E:FOR ai=1 TO 900*sek:NEXT:C     [3398]
ALL &BC71
1730 CLS:RETURN                                [832]
1740 CLS#1:PRINT#1,"          PROGRAMMINFOR    [3663]
MATIONEN
1750 PRINT#1                                   [318]
1760 PRINT#1:PRINT#1,"Backup erzeugen:         [3062]
1770 PRINT#1                                   [318]
1780 '                                         [203]
     '
1790 PRINT#1,"Es wird eine  Kopie  des  In     [5357]
halts  einer";
1800 PRINT#1,"Diskettenseite  auf  Kassett     [5183]
e   erzeugt.";
1810 PRINT#1,"Diese Kopie enthaelt keine d     [5024]
irekt  lauf-";
1820 PRINT#1,"faehigen Programme.              [2567]
1830 PRINT#1                                   [318]
1840 PRINT#1,"Das Diskettenlaufwerk und di     [4257]
e  Aufnahme-";
1850 PRINT#1,"geschwindigkeit koennen gewa     [4515]
ehlt werden.";
1860 PRINT#1,"Bei  Uebernahme  der  Vorgab     [4391]
ewerte   nur";
1870 PRINT#1,"ENTER  druecken.  Die  Bezei     [3320]
chnung   der";
1880 PRINT#1,"Diskettenseite darf maximal      [4422]
 13  Zeichen";
1890 PRINT#1,"lang sein.                       [1725]
1900 PRINT#1                                   [318]
1910 PRINT#1,"Bei 3600 Baud werden  die  1     [2747]
80 kB  einer";
1920 PRINT#1,"Diskettenseite  auf  Kassett     [4811]
enband   von";
1930 PRINT#1,"acht  Minuten  Laenge  gesch     [4132]
rieben.  Die";
1940 PRINT#1,"Aufzeichnung erfolgt in acht     [3997]
 Bloecken zu";
1950 PRINT#1,"jeweils 22,5 kB; dies ist gl     [3763]
eich dem In-";
1960 PRINT#1,"halt von 5 Spuren der Disket     [3110]
te.
1970 PRINT#1                                   [318]
1980 PRINT#1,"       > Weiter - Taste druec    [2142]
ken <";
1990 CALL &BB03:CALL &BB06                     [812]
2000 CLS#1:PRINT#1,"          PROGRAMMINFOR    [3663]
MATIONEN
2010 PRINT#1                                   [318]
2020 PRINT#1,"Am Ende des Kassettenbackup'     [4510]
s kann  eine";
2030 PRINT#1,"Aufnahmeluecke erzeugt  werd     [3365]
en,  um  bei";
2040 PRINT#1,"einer spaeteren Aktualisieru     [4127]
ng  zu  ver-";
2050 PRINT#1,"hindern, dass der Beginn des     [4421]
 nachfolgen-";
2060 PRINT#1,"den Backup's versehentlich       [4436]
ueberschrie-";
2070 PRINT#1,"ben wird.                        [1724]
2080 PRINT#1                                   [318]
2090 PRINT#1:PRINT#1,"Backup zurueckkopier     [3320]
en:
2100 PRINT#1                                   [318]
2110 PRINT#1,"Zum Zurueckkopieren sollte d     [4438]
ie  Diskette";
2120 PRINT#1,"entsprechend der Originaldis     [4296]
kette forma-";
2130 PRINT#1,"tiert sein. Ein eventueller      [4424]
Inhalt  wird";
2140 PRINT#1,"ueberschrieben. Bei falschem     [5810]
 Format wird";
2150 PRINT#1,"die Diskette umformatiert.       [3021]
2170 PRINT#1                                   [318]
```

Listing "BACKUP"

```
2180 PRINT#1,"Das Backup kann unabhaengig      [4959]
 vom  Quell-";
2190 PRINT#1,"laufwerk auf  beide  Laufwer     [4272]
ke  zurueck-";
2200 PRINT#1,"kopiert werden.                  [2634]
2210 PRINT#1:PRINT#1                           [789]
2220 PRINT#1,"       > Weiter - Taste druec    [2142]
ken <";
2230 CALL &BB03:CALL &BB06                     [812]
2240 CLS#1:PRINT#1,"          PROGRAMMINFOR    [3663]
MATIONEN
2250 PRINT#1:PRINT#1                           [789]
2260 PRINT#1,"Praktische Tips:                 [2810]
2270 PRINT#1                                   [318]
2280 PRINT#1,"Auf eine C60 Kassette  passe     [2806]
n  je  Seite";
2290 PRINT#1,"drei Backupfiles mit 3000 Ba     [4503]
ud und einer";
2300 PRINT#1,"Bandluecke von 30 Sekunden.      [3876]
2310 PRINT#1                                   [318]
2320 PRINT#1,"Auf eine C90 Kassette  passe     [3573]
n  je  Seite";
2330 PRINT#1,"fuenf Backupfiles mit 3300 B     [5766]
aud und  ei-";
2340 PRINT#1,"ner Bandluecke von 10 Sekund     [2943]
en.
2350 PRINT#1                                   [318]
2360 PRINT#1,"Die Blocknamen koennen nicht     [3809]
 mit dem CAT";
2370 PRINT#1,"-Befehl ermittelt werden,  d     [3665]
a  sie  ohne";
2380 PRINT#1,"System-Kennzeichnungsteil  a     [3391]
bgespeichert";
2390 PRINT#1,"werden. Der Name ist in den      [4181]
 Daten  ent-";
2400 PRINT#1,"halten.                          [1215]
2410 LOCATE#1,1,25:PRINT#1,"       > Weiter    [3577]
 - Taste druecken <";
2420 CALL &BB03:CALL &BB06:RETURN              [1141]
2430 '                                         [117]
2440 ' Up - Diskette formatieren (f=Sektor     [4199]
summand [Format])
2450 '                                         [117]
2460 spur=39:i0=0                              [777]
2470 POKE &3F01,drive                          [1408]
2480 FOR i=i0 TO spur                          [631]
2490  adr=&3F10:POKE &3F03,i:POKE &3F05,f+     [2491]
1
2500  FOR j=1 TO 9                             [604]
2505   IF j=6 THEN adr=&3F14                   [853]
2510   POKE adr,i:POKE adr+1,0:POKE adr+2,     [3001]
f+j:POKE adr+3,2:adr=adr+8
2520  NEXT                                     [350]
2560  CALL &3F00:' Track formatieren           [1882]
2570 NEXT:RETURN                               [940]
2580 '                                         [117]
2590 ' Menueteil Diskette formatieren          [3584]
2600 '                                         [117]
2610 MODE 1:PRINT:PRINT SPC(9)CHR$(24)" Di     [5750]
skette formatieren "CHR$(24)
2620 PRINT:PRINT:PRINT"  Laufwerk - A oder     [2615]
 B: ";
2630 a$=INKEY$:IF a$="" THEN 2630 ELSE a$=     [2361]
UPPER$(a$)
2640 IF a$="A" THEN drive=0:GOTO 2670          [2503]
2650 IF a$="B" THEN drive=1:GOTO 2670          [1319]
2660 GOTO 2630                                 [357]
2670 PRINT a$                                  [399]
2680 PRINT:PRINT:PRINT"  Format - 'S'ystem     [5393]
 oder 'D'aten: ";
2690 a$=INKEY$:IF a$="" THEN 2690 ELSE a$=     [2612]
UPPER$(a$)
2700 IF a$="S" THEN f=64:GOTO 2730             [1098]
2710 IF a$="D" THEN f=&C0:GOTO 2730            [1410]
2720 GOTO 2690                                 [393]
2730 PRINT a$                                  [399]
2740 LOCATE 3,11:INPUT"Formatieren ab Spur     [5431]
 (0 ... 42): ",i0
2750 IF i0>42 OR i0<0 THEN LOCATE 1,11:PRI     [2716]
NT SPACE$(100):GOTO 2740
2760 LOCATE 3,14:PRINT"Formatieren bis Spu     [7396]
r (";:PRINT USING"##";i0;:PRINT" ... 42):
";:INPUT"",spur
2770 IF spur>42 OR spur<i0 THEN LOCATE 1,1     [4174]
4:PRINT SPACE$(100):GOTO 2760
2780 PRINT:PRINT:PRINT"Diskette wird           [6435]
 formatiert -- bitte warten";
2790 GOSUB 2470:SOUND 1,100,100:RETURN         [1751]
```

Listing "BACKUP"

Ein Tipp – ein Wort

Funktionstastenbelegung für den CPC mit Turbo Pascal Schlüsselwörtern

Beide CP/M Betriebssysteme bieten ausgezeichnete Möglichkeiten, die Arbeit des Programmierers zu erleichtern. So lassen sich beispielsweise die Funktionstasten des CPC mit beliebigen Zeichenketten belegen, die auch nach dem Start des Turbo Pascal Editors abrufbereit zur Verfügung stehen.

Die Vorgehensweise ist bei den beiden Betriebssystemen unterschiedlich, das Ergebnis beinahe identisch. Die Schlüsselwörter können je nach Bedarf ausgetauscht oder durch andere ersetzt werden. In unseren Beispielen stehen unter CP/M 2.2 neunzehn und unter CP/M Plus zweiundzwanzig Schlüsselwörter zur Verfügung, davon jeweils neun bzw. elf in Verbindung mit der <SHIFT>-Taste.

Ablauf unter CP/M 2.2

1. Nehmen Sie eine leere Diskette im CP/M 2.2 Systemformat mit Betriebssystem. Kopieren Sie die Dateien SETUP.COM, DIR.COM und die Turbo Pascal Programm- und Overlay-Dateien auf diese Diskette. Danach rufen Sie CP/M 2.2 auf und starten es.
2. Jetzt laden und starten Sie SETUP.COM
2.1 ** Initial command buffer **
   - Frage mit "N" beantworten
   - Eingabe: dir↑M turbo↑M<RETURN>
   - Frage mit "Y" beantworten

2.2 ** Sign-on string **
   - Frage mit "N" beantworten
   - Eingabe: ↑\@ww↑\a@@↑]ww Turbo Pascal mit Funktionstasten-Belegung↑J↑M <RETURN>
   - Frage mit "Y" beantworten

2.3 ** Power-up string **
   - Frage mit "Y" beantworten

2.4 ** Keyboard translation set **
   - Frage mit "N" beantworten
   - Command: a10,128,131,131 <RET>
     a11,129,132,132 "
     a 3,130,133,133 "
     a20,134,137,137 "
     a12,135,138,138 "
     a 4,136,139,139 "
     a13,140,143,143 "
     a14,141,144,144 "
     a 5,142,145,145 "
     a15,146,147,147 "
     f "
   - Frage mit "Y" beantworten

2.5 ** Keyboard expansion set **
   - Frage mit "N" beantworten

   (* Die nachfolgenden Kommandos jeweils mit <SPACE> und <RETURN> abschließen *)
   - Command: a0,IF / a1,THEN / a3,FOR / a4,TO / a5,DOWNTO / a6,DO/ a7,REPEAT / a8,UNTIL/ a9,AND / a10,OR/ a11,XOR / a12,WHILE/ a13,WITH / a14,CASE/ a15,PROGRAM / a16,PROCEDURE/ a17,FUNCTION / a18,BEGIN / a19,END / f
   - Frage mit "Y" beantworten

2.6 Alle übrigen Fragen mit "Y" beantworten.

Nun kommt die Stunde der Wahrheit. Haben Sie alles richtig eingegeben, passiert folgendes:
- das durch Sie geänderte CP/M2.2 wird gestartet,
- auf dem Monitor erscheint die Meldung "TurboPascal mit Funktionstastenbelegung",
- das Directory der Diskette in Laufwerk A wird gelistet,
- Turbo Pascal wird gestartet,
- die eingegebenen Schlüsselwörter stehen nun zur Verfügung.

WARNUNG: Arbeiten Sie immer mit einer Kopie Ihres Betriebssystems, nie mit dem Original.

Da SETUP.COM Änderungen am Betriebssystem vornimmt, führen fehlerhafte Eingaben zu katastrophalen Folgen, bis hin zum Absturz, und Ihr Original ist vernichtet.

Ablauf unter CP/M Plus

Unter CP/M Plus brauchen Sie Ihr Betriebssystem nicht verändern. Mit Hilfe eines beliebigen Editors wird eine ASCII-Datei erstellt, die die Tastencodes und die jeweilen Zeichenketten enthält.

Nach dem Aufruf SETKEYS TURBO.KEY stehen die Schlüsselwörter zur Verfügung. Der Aufruf muß vor dem Start von Turbo Pascal erfolgen. Hier nun ein Beispiel mit dem Turbo Pascal Editor:

```
- Turbo Pascal aufrufen
- W (* Workfile *)
- Eingabe: TURBO.KEY
- E (* Edit *)
- (* Das nachfolgende Listing eingeben *)
- (* Den Editor mit <CTRL>KD verlassen *)
- S (* Datei abspeichern *)
- Q (* Turbo Pascal verlassen *)
```

Inhalt der Datei TURBO.KEY:

```
10 N "↑'128'"          15 N"↑'146'"
10 S "↑'129'"          15 S"↑'147'"
10 C "↑'129'"          15 C"↑'147'"
11 N "↑'130'"           7 N"↑'148'"
11 S "↑'131'"           7 S"↑'149'"
11 C "↑'131'"           7 C"↑'149'"
 3 N "↑'132'"         E 128"IF "
 3 S "↑'133'"         E 129"END "
 3 C "↑'133'"         E 130"THEN "
20 N "↑'134'"         E 131"CASE "
20 S "↑'135'"         E 132"ELSE "
20 C "↑'135'"         E 133"OF "
12 N "↑'136'"         E 134"FOR "
12 S "↑'137'"         E 135"WITH "
12 C "↑'137'"         E 136"TO "
 4 N "↑'138'"         E 137"IN "
 4 S "↑'139'"         E 138"DOWNTO "
 4 C "↑'139'"         E 139"AND "
13 N "↑'140'"         E 140"WHILE "
13 S "↑'141'"         E 141"OR "
13 C "↑'141'"         E 142"REPEAT "
14 N "↑'142'"         E 143"XOR "
14 S "↑'143'"         E 144"UNTIL "
14 C "↑'143'"         E 145"NOT "
 5 N "↑'144'"         E 146"BEGIN "
 5 S "↑'145'"         E 147"PROGRAM "
 5 C "↑'145'"         E 148"PROCEDURE "
                      E 149"FUNCTION "
```

Viel Erfolg und Spaß beim weiteren Experimentieren.

(H.-W. Fromme/JB)

Je kürzer, desto besser

BPC komprimiert Ihre Programme

BASIC-Programme können mehrere Pluspunkte für sich einnehmen: Sie sind leicht zu verstehen, kennen viele Möglichkeiten der Programmierung und lassen sich leicht auf andere Rechner anpassen. Leider gibt es hier auch ein Problem. Je übersichtlicher sie sind, desto länger werden sie. Und dann ist plötzlich kein Platz mehr da für diese oder jene Funktion, die das Programm noch verbessern soll. Dagegen gibt es aber ein Hilfsmittel, unseren BPC.

BPC bedeutet "BASIC Program-Code-Compressor". Es handelt sich um ein Maschinenprogramm, das BASIC-Programme im Speicher des CPC auf ein Minimum an Speicherplatzbedarf komprimiert (verkürzt), was gleichzeitig zur Beschleunigung des Programmablaufs beiträgt. Die Komprimierung hängt sehr stark von der Beschaffenheit des jeweiligen Programmes ab: Kommentarloser "Spaghetti-Code" und sogar Data-Lader werden meist noch um 10%, sparsam mit Kommentarzeilen versehene, geringfügig gegliederte Programme um 15-25%, weitergehend optisch und logisch gegliederter und kommentierter Programmtext um 30-40% verkürzt, und wer beim Programmieren gleich an BPC denkt und seinem Programm mit Leerzeichen, Kommentarzeilen und selbsterklärenden Variablennamen die Lesbarkeit von Klartext verleiht, darf mit 50% Verkürzung und mehr rechnen. Eine nach der letztgenannten Methode in 39K programmierte Adreßverwaltung läßt nach der Kompression also auch noch genug Platz frei für gut und gerne 150 Datensätze plus Diskettenpuffer.

Der Zeitbedarf der Kompression selbst fällt übrigens in keinem Fall nennenswert ins Gewicht, er bleibt bei kurzen Programmen unter 10K im Bereich weniger Sekunden und wird auch bei vollem Speicher 30 Sekunden nicht übersteigen.

So arbeitet das Programm

Zunächst tippen Sie den BASIC-Lader ab, der auf Ihrem CPC lauffähig ist, und zwar Listing 2 für CPC 664/6128 (im Programm ist nur der 6128 genannt) oder Listing 1 für den CPC 464. Das Ladeprogramm sollten Sie in jedem Fall nach dem Abtippen vor dem ersten Start speichern, weil es sich nach dem Durchlauf selbst löscht. Nun können Sie den Lader mit RUN starten, worauf ein Maschinenprogramm von knapp 1.5K direkt unterhalb des aktuellen HIMEM angelegt wird und der Lader sich mit NEW löscht. Jetzt laden Sie das zu verkürzende BASIC-Programm in den Speicher und starten die Kompression mit dem RSX-Befehl '!BPC'. Die Kompression erfolgt in drei Abschnitten:

In Pass 1 werden sämtliche Sprungziele in Zeilennummern gewandelt und die angesprungenen Zeilen im Programmtext markiert.

In Pass 2 findet nun die eigentliche Verkürzung statt. Kommentare, überflüssige Semikolons und Leerzeichen werden entfernt, Variablennamen auf ein bis drei Zeichen verkürzt (es werden natürlich alle bis zu 40 Zeichen des Namens berücksichtigt), Variablenlisten hinter NEXT durch eine entsprechende Anzahl einzelner NEXT-Befehle ersetzt und Zeilen, wo immer das geht, aneinandergefügt. Weil dabei der Bildschirmspeicher als Variablentabelle verwendet wird, wird der Bildschirm zuvor gelöscht und zeigt während der Kompression einzelne Querstreifen. Keine Angst, das ist normal, Ihr CPC ist nicht abgestürzt! Alles in allem nutzt BPC die Tatsache aus, daß Einschränkungen wie z. B. das Leerzeichen nach Befehlsworten, maximal 255 Zeichen in einer Zeile und Semikolons innerhalb von PRINT-Befehlen nur vom BASIC-Editor herrühren, der Interpreter diese platz- und zeitverschwendenden Zutaten bei der Programmausführung jedoch keineswegs benötigt.

In Pass 3 schließlich werden Sprünge in nicht mehr vorhandene Kommentarzeilen auf die jeweils folgende Zeile mit "echten" Befehlen korrigiert. Abschließend wird der Bildschirm nochmals gelöscht und der BASIC-Befehl CLEAR zur Anpassung der Programmzeiger durchgeführt. Wenn alles in Ordnung war, gibt BPC die Länge des Ausgangsprogramms (Source-Code), die des komprimierten Programms (Object-Code) sowie die Anzahl der eingesparten Bytes an. Das neue, verkürzte Programm kann jetzt ganz normal gestartet oder (sinnvoller) abgespeichert werden (nicht als ASCII-Datei!), es ist aber nur eingeschränkt listbar und läßt sich per Editor nicht mehr bearbeiten. Daher muß das Ausgangsprogramm unbedingt, für den Fall einer späteren Programmänderung, aufbewahrt werden! Wenn hingegen ein Fehler aufgetreten ist, erfolgt die Anzeige der Zeilennummer der fehlerhaften Zeile und einer Fehlermeldung; jeder Fehler muß im Ausgangsprogramm korrigiert werden und erfordert damit eine erneute Kompression. Folgende Fehler sind möglich:

1. Table space full: Die Variablentabelle übersteigt 16K. Das ist theoretisch möglich, wird aber kaum jemals eintreten.

2. Variable overflow: Es dürfen höchstens 4161 verschiedene Variablennamen denselben Anfangsbuchstaben haben. Auch dieser Fehler ist nur rein rechnerisch vorstellbar.

3. Invalid Variable: Beim CPC dürfen Variablennamen nicht länger sein als 40 Zeichen.

4. Invalid line number: Ein Sprung geht auf eine nicht vorhandene Zeile oder auf eine Kommentarzeile, die hinter der letzten Zeile mit "echten" Befehlen liegt (der Sprung ist nach der Kompression also hinter das Programmende gerichtet).

5. Object overflow: Der Object Code wird länger als der Source Code (am Programmbeginn fehlt ein schließendes Anführungszeichen, oder eine einstellige Variable, die am Programmbeginn häufig auftritt, kommt erst nach einer mehrstelligen Variable mit gleichem Anfangsbuchstaben erstmalig vor). Es ist außerdem zu beachten, daß ein Programm durch die Kompression auf dem CPC 664/6128 seine Lauffähigkeit auf dem CPC 464 einbüßen kann.

Und nun viel Spaß bei der fleißigen "Programmverkleinerung!"

(Marek Musial/jb)

für 464-664-6128

```
1000 SYMBOL AFTER 256                                    [1408]
1010 PRINT                                               [361]
1020 PRINT "***   B  P  C   ***"                         [1327]
1030 PRINT " - v 1.2  (464)  - "                         [1435]
1040 PRINT          " 1988 by M. Musial"                 [2194]
1050 PRINT                                               [361]
1060 MEMORY HIMEM-1442                                   [472]
1070 RESTORE 1180:z1!=1379                               [751]
1080 FOR z1!=HIMEM+1 TO HIMEM+z1!:READ ch$               [2187]
1090 IF LEN(ch$)=2 THEN POKE z1!,VAL("&"+c               [2257]
h$):GOTO 1120
1100 ad!=HIMEM+1+VAL("&"+RIGHT$(ch$,2)+LEF               [3826]
T$(ch$,2))
1110 POKE z1!,UNT(ad!) AND 255:z1!=z1!+1:P               [2815]
OKE z1!,INT(ad!/256)
1120 NEXT                                                [350]
1130 CALL HIMEM+1                                        [329]
1140 PRINT "Extension installed"                         [1435]
1150 PRINT "Type '|BPC' to call"                         [2091]
1160 PRINT                                               [361]
1170 NEW                                                 [318]
1180 DATA 01,57*05,21,63*05,C3,D1,BC,21,00               [3116]
,C0,11,01,C0,01,FF,3F,36,00
1190 DATA ED,B0,21,6C*05,11,6D*05,01,33,00               [3202]
,36,1F,ED,B0,AF,32,6B*05,2A
1200 DATA 83,AE,22,A0*05,CD,80*01,B7,CC,C2               [2726]
*01,B7,CC,F4*02,B7,CD,54*03
1210 DATA 0C,00,CA,F5*00,CD,54*03,45,52,52               [3091]
,4F,52,20,69,6E,20,6C,69,6E
1220 DATA 65,20,00,2A,69*05,CD,1F*03,CD,54               [3458]
*03,3A,0D,0A,00,3D,20,14,CD
1230 DATA 54*03,54,61,62,6C,65,20,73,70,61               [3076]
,63,65,20,66,75,6C,6C,00,3D
1240 DATA 20,15,CD,54*03,56,61,72,69,61,62               [3870]
,6C,65,20,6F,76,65,72,66,6C
1250 DATA 6F,77,00,3D,20,14,CD,54*03,49,6E               [4087]
,76,61,6C,69,64,20,76,61,72
1260 DATA 69,61,62,6C,65,00,3D,20,17,CD,54               [2854]
*03,49,6E,76,61,6C,69,64,20
1270 DATA 6C,69,6E,65,20,6E,75,6D,62,65,72               [1889]
,00,3D,20,13,CD,54*03,4F,62
1280 DATA 6A,65,63,74,20,6F,76,65,72,66,6C               [3228]
,6F,77,00,CD,54*03,0D,0A,0A
1290 DATA 43,6F,6D,70,72,65,73,73,69,6F,6E               [2160]
,20,61,62,6F,72,74,65,64,2E
1300 DATA 0D,0A,0A,00,C9,DF,60*05,CD,54*03               [3722]
,53,6F,75,72,63,65,20,63,6F
1310 DATA 64,65,3A,20,00,2A,A0*05,ED,5B,81               [3473]
,AE,37,ED,52,CD,1F*03,CD,54*03
1320 DATA 20,62,79,74,65,73,0D,0A,4F,62,6A               [3674]
,65,63,74,20,63,6F,64,65,3A
1330 DATA 20,00,2A,83,AE,37,ED,52,CD,1F*03               [2625]
,CD,54*03,20,62,79,74,65,73
1340 DATA 0D,0A,3D,3E,20,00,2A,A0*05,ED,5B               [4024]
,83,AE,B7,ED,52,CD,1F*03,CD
1350 DATA 54*03,20,62,79,74,65,73,20,73,61               [2884]
,76,65,64,0D,0A,0A,43,6F,6D
1360 DATA 70,72,65,73,73,69,6F,6E,20,66,69               [2903]
,6E,69,73,68,65,64,2E,0D,0A
1370 DATA 0A,00,C9,2A,81,AE,AF,CD,D6*04,D8               [4029]
,CD,01*04,B7,28,F6,FE,1D,28
1380 DATA 09,FE,1E,28,1A,CD,CD*04,18,ED,3C               [2627]
,77,23,4E,23,46,2B,03,03,03
1390 DATA 0A,77,23,03,0A,77,23,0B,0B,18,0C               [3072]
,23,4E,23,46,23,CD,5D*04,3E
1400 DATA 04,C8,03,0A,CB,FF,02,18,C6,2A,81               [2912]
,AE,5D,54,13,CD,D6*04,30,0D
1410 DATA CD,39*05,AF,12,13,12,13,ED,53,83               [2931]
,AE,C9,CC,39*05,CD,01*04,B7
1420 DATA 20,1A,3A,6B*05,CB,77,CC,2E*05,C4               [3623]
,39*05,18,D8,23,18,EA,CD,CD*04
1430 DATA 18,E5,CD,C2*04,18,E0,FE,01,20,05               [4157]
,CD,2E*05,18,EA,FE,0E,30,06
1440 DATA CD,61*03,C0,18,CD,FE,20,28,DC,FE               [3539]
,22,20,06,CD,8C*04,C0,18,BF
1450 DATA FE,3B,20,14,CD,C2*04,3A,6B*05,CB               [4466]
,4F,28,B1,CD,B1*04,FE,02,38
1460 DATA AA,1B,18,A7,FE,8C,20,47,3A,6B*05               [2498]
,CB,67,28,09,CB,A7,CB,C7,32
1470 DATA 6B*05,36,2C,06,01,CD,C2*04,CD,B1               [3302]
*04,FE,22,20,07,CD,01*04,CD
1480 DATA 8C*04,C0,7E,FE,02,30,0E,CD,B8*04               [2392]
,3A,6B*05,CB,E7,32,6B*05,C3
1490 DATA DD*01,FE,2C,20,08,CD,B8*04,12,23               [3471]
,13,18,D3,12,23,13,18,DC,FE
1500 DATA A0,20,10,1B,1A,13,FE,97,CA,F0*01               [4212]
,FE,EB,CA,F0*01,C3,F8*01,FE
1510 DATA A1,20,0B,3A,6B*05,CB,F7,32,6B*05               [3721]
,C3,F8*01,FE,B0,20,21,CD,C2*04
1520 DATA 0E,FF,CD,01*04,FE,02,DA,DD*01,CD               [2429]
,CD*04,FE,0E,30,F1,0C,28,EE
1530 DATA 3E,01,12,13,3E,B0,12,13,18,E4,FE               [2880]
,BF,20,0B,3A,6B*05,CB,CF,32
1540 DATA 6B*05,C3,F8*01,FE,ED,20,0B,3A,6B               [3203]
*05,CB,8F,32,6B*05,C3,F8*01
1550 DATA FE,C0,CA,F3*01,FE,C5,CA,F3*01,C3               [2336]
,F8*01,2A,81,AE,AF,CD,D6*04
1560 DATA D8,CD,01*04,B7,28,F6,FE,1E,28,05               [3582]
,CD,CD*04,18,F1,3D,77,23,4E
1570 DATA 23,46,2B,CD,5D*04,3E,04,D8,0B,71               [4397]
,23,70,23,18,DD,F5,D5,C5,01
1580 DATA 00,05,11,01,00,D5,11,0A,00,D5,11               [3053]
,64,00,D5,11,E8,03,D5,11,10
1590 DATA 27,D5,D1,AF,3C,ED,52,30,FB,19,3D               [3131]
,20,01,0D,C6,30,0C,C4,5A,BB
1600 DATA 10,EC,CC,5A,BB,C1,D1,F1,C9,E3,F5               [2968]
,7E,23,CD,5A,BB,B7,20,F8,F1
1610 DATA E3,C9,C5,CD,F9*04,12,13,D5,78,FE               [3735]
,2C,3E,03,D2,FD*03,05,05,23
1620 DATA 23,70,E5,05,23,CB,76,28,02,CB,EE               [3516]
,10,F7,11,00,C0,1A,B7,28,15
1630 DATA 47,E1,E5,1A,BE,13,23,28,08,68,26               [2731]
,00,2C,19,EB,18,EB,10,F0,18
1640 DATA 37,E1,E5,6E,26,00,2C,2C,19,3E,01               [3624]
,38,54,E1,E5,4E,06,00,ED,B0
1650 DATA E1,E5,23,7E,E6,1F,3D,07,4F,21,6C               [3375]
*05,09,0E,02,ED,B0,2B,2B,1B
1660 DATA 1B,3E,60,34,BE,20,09,36,20,23,34               [3539]
,BE,3E,02,38,29,E1,4E,0D,23
1670 DATA 7E,CB,BF,09,E3,70,23,70,06,03,FE               [3807]
,1F,28,06,23,77,1A,13,10,F6
1680 DATA CB,FE,23,EB,E1,E5,B7,ED,52,E1,3E               [4282]
,05,38,05,C1,AF,C9,E1,D1,C1
1690 DATA B7,C9,7E,F5,E5,CD,0A*04,E1,F1,C9               [4128]
,06,01,FE,02,D8,FE,0E,30,0B
1700 DATA 23,04,23,04,23,04,CB,7E,28,FA,C9               [3392]
,FE,19,D8,06,02,C8,06,03,FE
1710 DATA 1F,D8,06,06,C8,06,01,FE,22,20,0A               [2877]
,23,7E,B7,C8,04,FE,22,C8,18
1720 DATA F6,FE,7C,28,D5,FE,8C,20,09,3E,01               [4221]
,05,23,04,BE,38,FB,C9,FE,C0
1730 DATA 20,03,AF,18,F2,FE,C5,28,F9,FE,FF               [3830]
,D8,04,C9,D5,F5,E5,2A,81,AE
1740 DATA 23,E5,5E,23,56,23,CB,BA,7B,B2,20               [3739]
,04,1E,01,18,12,7E,23,66,6F
1750 DATA ED,42,30,04,E1,19,18,E5,1E,02,28               [3570]
,02,1E,00,C1,E1,F1,CB,3B,D1
1760 DATA C9,CD,C2*04,C5,1B,05,1A,FE,20,28               [3570]
,F9,13,FE,22,20,04,78,B7,20
1770 DATA 0D,3E,22,12,13,E5,B7,ED,52,E1,3E               [3382]
,05,38,01,AF,B7,C1,C9,7E,FE
1780 DATA 20,C0,23,18,F9,F5,1B,1A,FE,20,28               [3757]
,FA,13,F1,C9,CD,F9*04,C5,48
1790 DATA 06,00,ED,B0,C1,C9,F5,C5,48,06,00               [2242]
,09,C1,F1,C9,C5,F5,D5,23,7E
1800 DATA 23,46,23,0E,03,B0,28,08,0E,00,CB               [2407]
,78,20,02,0E,02,5E,23,56,23
1810 DATA ED,53,69*05,D1,F1,CB,39,C1,C9,F5               [4370]
,3A,6B*05,CB,7F,28,11,CB,67
1820 DATA C4,39*05,20,0A,CB,47,20,1B,3E,01               [3355]
,12,13,18,10,ED,53,67*05,13
1830 DATA 13,3A,69*05,12,13,3A,6A*05,12,13               [3533]
,3A,6B*05,F6,81,CB,A7,32,6B*05
1840 DATA F1,C9,F5,3A,6B*05,E6,FC,32,6B*05               [2784]
,F1,C9,F5,3A,6B*05,CB,7F,28
1850 DATA 14,AF,32,6B*05,12,13,E5,D5,2A,67               [2497]
*05,EB,ED,52,EB,73,23,72,D1
1860 DATA E1,F1,C9,5C*05,C3,09*00,42,50,C3               [2269]
,00,32,C1,FD
```

Listing "BPC 464"

```
1000 SYMBOL AFTER 256                                   [1408]
1010 PRINT                                              [361]
1020 PRINT "***   B  P  C   ***"                        [1327]
1030 PRINT " - v 1.2  (6128) - "                        [1077]
1040 PRINT          " 1988 by M. Musial"                [2194]
1050 PRINT                                              [361]
1060 MEMORY HIMEM-1435                                  [507]
1070 RESTORE 1180:zl!=1372                              [722]
1080 FOR zl!=HIMEM+1 TO HIMEM+zl!:READ ch$              [2187]
1090 IF LEN(ch$)=2 THEN POKE zl!,VAL("&"+c               [2257]
h$):GOTO 1120
1100 ad!=HIMEM+1+VAL("&"+RIGHT$(ch$,2)+LEF              [3826]
T$(ch$,2))
1110 POKE zl!,UNT(ad!) AND 255:zl!=zl!+1:P              [2815]
OKE zl!,INT(ad!/256)
1120 NEXT                                               [350]
1130 CALL HIMEM+1                                       [329]
1140 PRINT "Extension installed"                        [1435]
1150 PRINT "Type '|BPC' to call"                        [2091]
1160 PRINT                                              [361]
1170 NEW                                                [318]
1180 DATA 01,50*05,21,5C*05,C3,D1,BC,21,00              [2651]
,C0,11,01,C0,01,FF,3F,36,00
1190 DATA ED,B0,21,65*05,11,66*05,01,33,00              [3356]
,36,1F,ED,B0,AF,32,64*05,2A
1200 DATA 66,AE,22,99*05,CD,80*01,B7,CC,C2              [2374]
*01,B7,CC,F4*02,B7,CD,54*03
1210 DATA 0C,00,CA,F5*00,CD,54*03,45,52,52              [3091]
,4F,52,20,69,6E,20,6C,69,6E
1220 DATA 65,20,00,2A,62*05,CD,1F*03,CD,54              [1853]
*03,3A,0D,0A,00,3D,20,14,CD
1230 DATA 54*03,54,61,62,6C,65,20,73,70,61              [3076]
,63,65,20,66,75,6C,6C,00,3D
1240 DATA 20,15,CD,54*03,56,61,72,69,61,62              [3870]
,6C,65,20,6F,76,65,72,66,6C
1250 DATA 6F,77,00,3D,20,14,CD,54*03,49,6E              [4087]
,76,61,6C,69,64,20,76,61,72
1260 DATA 69,61,62,6C,65,00,3D,20,17,CD,54              [2854]
*03,49,6E,76,61,6C,69,64,20
1270 DATA 6C,69,6E,65,20,6E,75,6D,62,65,72              [1889]
,00,3D,20,13,CD,54*03,4F,62
1280 DATA 6A,65,63,74,20,6F,76,65,72,66,6C              [3228]
,6F,77,00,CD,54*03,0D,0A,0A
1290 DATA 43,6F,6D,70,72,65,73,73,69,6F,6E              [2160]
,20,61,62,6F,72,74,65,64,2E
1300 DATA 0D,0A,0A,00,C9,DF,59*05,CD,54*03              [2136]
,53,6F,75,72,63,65,20,63,6F
1310 DATA 64,65,3A,20,00,2A,99*05,ED,5B,64              [2768]
,AE,37,ED,52,CD,1F*03,CD,54*03
1320 DATA 20,62,79,74,65,73,0D,0A,4F,62,6A              [3674]
,65,63,74,20,63,6F,64,65,3A
1330 DATA 20,00,2A,66,AE,37,ED,52,CD,1F*03              [3332]
,CD,54*03,20,62,79,74,65,73
1340 DATA 0D,0A,3D,3E,20,00,2A,99*05,ED,5B              [2858]
,66,AE,B7,ED,52,CD,1F*03,CD
1350 DATA 54*03,20,62,79,74,65,73,20,73,61              [2884]
,76,65,64,0D,0A,0A,43,6F,6D
1360 DATA 70,72,65,73,73,69,6F,6E,20,66,69              [2903]
,6E,69,73,68,65,64,2E,0D,0A
1370 DATA 0A,00,C9,2A,64,AE,AF,CD,D6*04,D8              [3481]
,CD,01*04,B7,28,F6,FE,1D,28
1380 DATA 09,FE,1E,28,1A,CD,CD*04,18,ED,3C              [2627]
,77,23,4E,23,46,2B,03,03,03
1390 DATA 0A,77,23,03,0A,77,23,0B,0B,18,0C              [3072]
,23,4E,23,46,23,CD,5D*04,3E
1400 DATA 04,C8,03,0A,CB,FF,02,18,C6,2A,64              [3614]
,AE,5D,54,13,CD,D6*04,30,0D
1410 DATA CD,32*05,AF,12,13,12,13,ED,53,66              [2235]
,AE,C9,CC,32*05,CD,01*04,B7
1420 DATA 20,1A,3A,64*05,CB,77,CC,27*05,C4              [4189]
,32*05,18,D8,23,18,EA,CD,CD*04
1430 DATA 18,E5,CD,C2*04,18,E0,FE,01,20,05              [4300]
,CD,27*05,18,EA,FE,0E,30,06
1440 DATA CD,61*03,C0,18,CD,FE,20,28,DC,FE              [3539]
,22,20,06,CD,8C*04,C0,18,BF
1450 DATA FE,3B,20,14,CD,C2*04,3A,64*05,CB              [3886]
,4F,28,B1,CD,B1*04,FE,02,38
1460 DATA AA,1B,18,A7,FE,8C,20,47,3A,64*05              [3629]
,CB,67,28,09,CB,A7,CB,C7,32
```

Listing "BPC 664/6128"

```
1470 DATA 64*05,36,2C,06,01,CD,C2*04,CD,B1              [2972]
*04,FE,22,20,07,CD,01*04,CD
1480 DATA 8C*04,C0,7E,FE,02,30,0E,CD,B8*04              [2221]
,3A,64*05,CB,E7,32,64*05,C3
1490 DATA DD*01,FE,2C,20,08,CD,B8*04,12,23              [3471]
,13,18,D3,12,23,13,18,DC,FE
1500 DATA A0,20,10,1B,1A,13,FE,97,CA,F0*01              [4212]
,FE,EB,CA,F0*01,C3,F8*01,FE
1510 DATA A1,20,0B,3A,64*05,CB,F7,32,64*05              [3357]
,C3,F8*01,FE,B0,20,21,CD,C2*04
1520 DATA 0E,FF,CD,01*04,FE,02,DA,DD*01,CD              [2429]
,CD*04,FE,0E,30,F1,0C,28,EE
1530 DATA 3E,01,12,13,3E,B0,12,13,18,E4,FE              [2827]
,BF,20,0B,3A,64*05,CB,CF,32
1540 DATA 64*05,C3,F8*01,FE,ED,20,0B,3A,64              [3108]
*05,CB,8F,32,64*05,C3,F8*01
1550 DATA FE,C0,CA,F3*01,FE,C5,CA,F3*01,C3              [3413]
,F8*01,2A,64,AE,AF,CD,D6*04
1560 DATA D8,CD,01*04,B7,28,F6,FE,1E,28,05              [3582]
,CD,CD*04,18,F1,3D,77,23,4E
1570 DATA 23,46,2B,CD,5D*04,3E,04,D8,0B,71              [4397]
,23,70,23,18,DD,F5,D5,C5,01
1580 DATA 00,05,11,01,00,D5,11,0A,00,D5,11              [3053]
,64,00,D5,11,E8,03,D5,11,10
1590 DATA 27,D5,D1,AF,3C,ED,52,30,FB,19,3D              [3131]
,20,01,0D,C6,30,0C,C4,5A,BB
1600 DATA 10,EC,CC,5A,BB,C1,D1,F1,C9,E3,F5              [2968]
,7E,23,CD,5A,BB,B7,20,F8,F1
1610 DATA E3,C9,C5,CD,F9*04,12,13,D5,78,FE              [3735]
,2C,3E,03,D2,FD*03,05,05,23
1620 DATA 23,70,E5,05,23,CB,76,28,02,CB,EE              [3516]
,10,F7,11,00,C0,1A,B7,28,15
1630 DATA 47,E1,E5,1A,BE,13,23,28,08,68,26              [2731]
,00,2C,19,EB,18,EB,10,F0,18
1640 DATA 37,E1,E5,6E,26,00,2C,2C,19,3E,01              [3624]
,38,54,E1,E5,4E,06,00,ED,B0
1650 DATA E1,E5,23,7E,E6,1F,3D,07,4F,21,65              [3752]
*05,09,0E,02,ED,B0,2B,2B,1B
1660 DATA 1B,3E,60,34,BE,20,09,36,20,23,34              [3539]
,BE,3E,02,38,29,E1,4E,0D,23
1670 DATA 7E,CB,BF,09,E3,70,23,70,06,03,FE              [3807]
,1F,28,06,23,77,1A,13,10,F6
1680 DATA CB,FE,23,EB,E1,E5,B7,ED,52,E1,3E              [4282]
,05,38,05,C1,AF,C9,E1,D1,C1
1690 DATA B7,C9,7E,F5,E5,CD,0A*04,E1,F1,C9              [4128]
,06,01,FE,02,D8,FE,0E,30,0B
1700 DATA 23,04,23,04,23,04,CB,7E,28,FA,C9              [3392]
,FE,19,D8,06,02,C8,06,03,FE
1710 DATA 1F,D8,06,06,C8,06,01,FE,22,20,0A              [2877]
,23,7E,B7,C8,04,FE,22,C8,18
1720 DATA F6,FE,7C,28,D5,FE,8C,20,09,3E,01              [4221]
,05,23,04,BE,38,FB,C9,FE,C0
1730 DATA 20,03,AF,18,F2,FE,C5,28,F9,FE,FF              [3729]
,D8,04,C9,D5,F5,E5,2A,64,AE
1740 DATA 23,E5,5E,23,56,23,CB,BA,7B,B2,20              [3739]
,04,1E,01,18,12,7E,23,66,6F
1750 DATA ED,42,30,04,E1,19,18,E5,1E,02,28              [3570]
,02,1E,00,C1,E1,F1,CB,3B,D1
1760 DATA C9,CD,C2*04,C5,1B,05,1A,FE,20,28              [3570]
,F9,13,FE,22,20,04,78,B7,20
1770 DATA 0D,3E,22,12,13,E5,B7,ED,52,E1,3E              [3382]
,05,38,01,AF,B7,C1,C9,7E,FE
1780 DATA 20,C0,23,18,F9,F5,1B,1A,FE,20,28              [3757]
,FA,13,F1,C9,CD,F9*04,C5,48
1790 DATA 06,00,ED,B0,C1,C9,F5,C5,48,06,00              [2242]
,09,C1,F1,C9,C5,F5,D5,23,7E
1800 DATA 23,46,23,0E,03,B0,28,08,0E,00,CB              [2407]
,78,20,02,0E,02,5E,23,56,23
1810 DATA ED,53,62*05,D1,F1,CB,39,C1,C9,F5              [2723]
,3A,64*05,CB,7F,28,0A,CB,47
1820 DATA 20,1B,3E,01,12,13,18,10,ED,53,60              [2554]
*05,13,13,3A,62*05,12,13,3A
1830 DATA 63*05,12,13,3A,64*05,F6,81,CB,A7              [2484]
,32,64*05,F1,C9,F5,3A,64*05
1840 DATA E6,FC,32,64*05,F1,C9,F5,3A,64*05              [2893]
,CB,7F,28,14,AF,32,64*05,12
1850 DATA 13,E5,D5,2A,60*05,EB,ED,52,EB,73              [3045]
,23,72,D1,E1,F1,C9,55*05,C3
1860 DATA 09*00,42,50,C3,00,33,C1,FD                    [1420]
```

Listing "BPC 664/6128"

Mit KOMPOS KByte sparen!

Bilder, die den meisten Programmen mit auf die Diskette gegeben werden, sind auf dem Monitor zwar schön anzusehen, haben aber einen großen Nachteil. 17 KByte pro Bild sind eine ganze Menge an Speicherplatz, die pro Diskette verbraucht werden. Es gibt aber Möglichkeiten, diese Dateien so zu verkleinern, daß von den 17 KByte eine nicht unerhebliche Menge an Speicherplatz wieder zum freien Gebrauch freigegeben wird. KOMPOS ist so eine Möglichkeit.

Dieses kleine Programm komprimiert Bildschirmdateien, die normal einen Platzbedarf von 17 KByte auf Diskette haben, auf bis zu 313 Bytes (einfarbiger Bildschirm). Im Durchschnitt werden die neu abgespeicherten Screens eine Länge von 7 bis 8 KByte haben.
Die Idee für dieses Programm stammt noch aus der Zeit, als die Drei-Zoll-Disketten für den CPC 10,– bis 12,– DM kosteten. Also kostete, bei einem theoretisch nutzbaren Speicherplatz von 356KByte, eine Bildschirmdatei ca. 60 Pf. Wenn man nun annimmt, daß man im Schnitt vier bis fünf solcher Dateien auf einer Diskette abgespeichert hat, ist das schon ein Viertel der Anschaffungskosten. Um diese Kosten zu senken, wurde also KOMPOS erstellt. Es komprimiert eine beliebige Datei und integriert dabei den Positionierer. Damit benötigt man kein zusätzliches Programm mehr bis auf eine kleine Anweisungsfolge, welche den alten Bildschirm wieder positioniert. Auch die farbenfrohesten Screens (z.B. Ghost'N'Goblins: 14kB) werden im Gegensatz zu anderen Screenkompressoren von KOMPOS noch komprimiert.

Zum Ablauf des Programmes

Nachdem man das Listing fehlerfrei abgetippt hat, sollten Sie es abspeichern und danach mit RUN starten. Dann warten Sie, bis die beiden MC-Blöcke einge'poke't worden sind. Danach verlangt das Programm nach dem Namen der Bildschirmdatei, die bearbeitet werden soll. Es muß sich natürlich auch der entsprechende Datenträger im Laufwerk befinden. Nach dem Komprimieren des Bildschirms wird nach dem neuen Namen der Datei gefragt. Falls die Datei unter dem alten Namen wieder abgespeichert werden soll, wird das Kürzel ".KOM" angehängt. Um nun eine komprimierte Datei wieder auf dem Bildschirm sichtbar zu machen, wird folgende Anweisungsfolge notwendig:

```
MODE m:MEMORY &3FFF:'m=Modus-Parameter
LOAD"XXXXXXXX.KOM":'hier den Namen des komprimierten Files ein-
                    setzen
CALL &4000
```

Den richtigen Bildschirmmodus m findet man, indem man m von 0-2 ausprobiert. Falls sich dabei senkrechte Streifen bilden, muß man m um eins erhöhen.

(Ulrich Köster/jb)

für 464-664-6128

```
10 '************************                        [577]
20 '***** K O M P O S *****                        [1419]
30 '************************                        [577]
40 'Dieses Programm laed eine Bildschirmda          [3792]
tei vom
50 'Datentraeger und komprimiert das Bild           [2303]
60 'Die komprimierte Datei wird mit folgen          [2961]
der
70 'Ladeanweisung wieder auf den Bildschir          [2329]
m aus-
80 'gegeben:     MODE m                             [1930]
90 '             MEMORY &3FFF                       [897]
100 '            LOAD "XXXXXXXX.KOM"                [2257]
110 '            CALL &4000                         [686]
120 'Wobei "XXXXXXXX.KOM" fuer einen Datei          [3168]
namen einer
130 'komprimierten Bildschirmdatei steht.           [2395]
140 'Der Bildschirmmodus muss natuerlich            [2277]
150 'auch erst richtig eingestellt werden,          [2836]
160 'sonst kommt es zu einem ungewollten            [1126]
170 'Streifenmuster.                                [343]
180 '*** Expander in Speicher einlesen ***          [1609]
190 '************************************           [1476]
200 MEMORY &3FFF                                    [758]
210 FOR adr=&4000 TO &4036:READ a$                  [1142]
220 wert=VAL("&"+a$):sum=sum+wert                   [2543]
230 POKE adr,wert:NEXT                              [851]
240 DATA 21,36,40,11,00,C0,23,46                    [1219]
250 DATA CB,78,20,0A,23,7E,12,CD                    [998]
260 DATA 22,40,10,F9,18,F0,CB,B8                    [1011]
270 DATA 23,7E,12,CD,22,40,10,F8                    [1271]
280 DATA 18,E4,3E,08,82,57,FE,08                    [395]
290 DATA 38,01,C9,C6,C0,57,13,7A                    [1441]
300 DATA FE,C8,28,01,C9,D1,C9                       [839]
310 IF sum<>&167D THEN PRINT"DATA-Fehler i          [4427]
m Expanderteil":END
320 '*** Komprimierroutine einpoken ***             [2662]
330 '************************************           [1391]
340 sum=0                                           [431]
350 FOR adr=&A000 TO &A07D:READ a$                  [1461]
360 wert=VAL("&"+a$):sum=sum+wert                   [2543]
370 POKE adr,wert:NEXT                              [851]
380 DATA 21,38,40,DD,21,00,C0,11                    [531]
390 DATA 00,C8,FD,21,37,40,97,FD                    [1455]
400 DATA 77,00,1A,DD,BE,00,28,22                    [1785]
410 DATA DD,7E,00,77,CD,5F,A0,FD                    [1394]
420 DATA 34,00,23,3E,7E,FD,BE,00                    [1408]
430 DATA 28,06,1A,DD,BE,00,20,E8                    [960]
440 DATA FD,CB,00,FE,E5,FD,E1,23                    [1727]
450 DATA 18,D4,FD,34,00,CD,5F,A0                    [983]
460 DATA 3E,7E,FD,BE,00,28,06,1A                    [1459]
470 DATA DD,BE,00,28,ED,FD,34,00                    [1421]
480 DATA DD,7E,00,77,CD,5F,A0,23                    [1659]
490 DATA 23,FD,23,FD,23,18,AF,D5                    [1639]
500 DATA DD,E1,3E,08,82,57,FE,08                    [1083]
510 DATA 38,01,C9,28,0A,C6,C0,57                    [1607]
520 DATA 13,7A,FE,C8,28,01,C9,D1                    [934]
530 DATA 22,7C,A0,C9,00,00                          [598]
540 IF sum<>&3829 THEN PRINT"DATA-Fehler i          [6591]
m Komprimierteil !":END
550 MODE 2                                          [513]
560 a$=STRING$(110,207)+"  K o m p o s V2.          [4045]
0  "+STRING$(110,207)
570 PRINT a$                                        [399]
580 PRINT,"Welche Bildschirmdatei moechten          [8022]
 Sie komprimieren :";:INPUT"",n$
590 IF n$=""THEN 580 ELSE LOAD n$,&C000             [882]
600 CALL &A000:MODE 2:PRINT a$                      [1664]
610 laenge=PEEK(&A07C)+PEEK(&A07D)*256-&3F          [2320]
FF
620 PRINT,n$" wurde von 17 kByte auf ";ROU          [6255]
ND((laenge+128)/1024+0.5);"kByte komprimie
rt"
630 PRINT:PRINT,"Die Datei wieder unter "n          [3682]
$" absaven ? ";
640 CALL &BB06:IF NOT INKEY(45) THEN PRINT          [2617]
"Ja":GOTO 670
650 PRINT" Nein":PRINT:PRINT,"Neuer Datein          [6040]
ame (max. 8 Ziffern) :";:INPUT"",n$
660 IF LEN(n$)>8 THEN 650                           [1242]
670 PRINT:PRINT,"Datentraeger bereit (J)            [3100]
?"
680 CALL &BB06:IF INKEY(45) THEN 680                [2084]
690 SAVE LEFT$(n$,8)+".kom",b,&4000,laenge          [2853]
,&4000
700 PRINT:PRINT,,"Noch eine Datei komprimi          [3814]
eren (J)"
710 CALL &BB06:IF NOT INKEY(45) THEN 550            [1810]
```

Listing "KOMPOS"

Aus Eins mach Zwei

Stellen Sie sich mal folgende Situation vor: Sie sitzen an einem warmen Sommerabend vor Ihrem Computer und brauchen dringend noch einige Daten aus Ihrem Dateiprogramm. Nebenbei fällt Ihnen auf, daß auf der Datendiskette kein Platz mehr ist, eine neue muß her, und die möchte bitteschön auch noch vorher formatiert werden. Gesagt, getan, die Diskette mit dem Formatierprogramm wird ins Laufwerk geschoben, und die Formatierroutine gestartet. Wie schon gesagt, der Abend ist schön und warm, die Gedanken sind beim nächsten Grillfest und die Formatierdiskette wird formatiert. Das Ergebnis ist klar: Diskette und gute Laune sind kaputt. Was also muß her? Ein gutes Kopierprogramm.

Sicherheitskopien von den wichtigsten Disketten sind so nötig, wie eine Tastatur beim Computer. Sehr schnell ist es passiert, daß ein Programm, das für die Arbeit mit dem Computer fast unentbehrlich geworden ist, durch irgendeinen widrigen Umstand nicht mehr zur Verfügung steht, sei es durch einen Fehler des Benutzers, oder durch mechanische Fehler. Erst wenn so etwas passiert, weiß man ein gutes Kopierprogramm, sowie die angefertigten Sicherheitskopien zu schätzen. Bei der Entwicklung dieses Kopierprogrammes wurde besonders auf Vielseitigkeit und Bedienungsfreundlichkeit Wert gelegt. Nachdem man das Programm mit RUN"COPYALL" gestartet hat, wird das Programm und der Maschinenspracheteil geladen. Gleich darauf erscheint

Das Hauptmenü

Hier kann man die Tracknummer ändern, wo das Kopieren begonnen bzw. beendet werden soll. So ist es beispielsweise möglich, auch die Tracks 40-43 zu kopieren. Des weiteren ist es möglich, die Anzahl der vorhandenen Laufwerke einzugeben. Es kann mit einem oder zwei Laufwerken gearbeitet werden. Arbeitet man mit zwei Laufwerken, muß sich die Source-Diskette (also die Diskette, die kopiert werden soll) im Drive A und die Destination-Disk (die Zieldiskette) in Drive B befinden. Das Kopieren mit zwei Laufwerken ist bei diesem Programm eine enorme Arbeitserleichterung.
Mit dem Menüpunkt 'P' kann man das Programm verlassen und ENTER läßt der Kopiervorgang beginnen. Es gibt aber noch einen weiteren Menüpunkt, den des Copy-Formats.

Unser Kopierprogramm unterscheidet prinzipiell zwischen zwei Kopierarten: Normal-Copy und CAD-Copy. Bei Normal-Copy wird das Format des ersten Tracks bestimmt, und danach wird davon ausgegangen, daß die anderen Tracks das gleiche Format besitzen. Bei CAD-Copy wird für jeden Track das Format bestimmt, so daß diese Art des Kopierens zwar langsamer, aber sicherer ist.

Der Kopiervorgang

Startet man nun den Kopiervorgang mit ENTER, wird man aufgefordert, bei zwei Laufwerken die Disketten einzulegen oder, bei einem Laufwerk, die Diskette entsprechend zu wechseln. Es empfiehlt sich, die Original-Diskette mit einem Schreibschutz zu versehen, so daß während des Kopierens bei einem Verwechseln der Disketten kein Schaden entsteht.

Im Folgenden wird ein grober Ablaufvorgang des Kopierens geschildert:

Erst wird das Format der ersten sechs Tracks erkannt (es werden auch die Disketten kopiert, die man mit DISCCOPY der CP/M-Diskette nicht kopieren kann, weil das Programm illegale Tracks erkennt!) und diese werden dann, falls formatiert, eingelesen. Dann erfolgt ein Disketten- bzw. Laufwerkswechsel, die zweite Diskette wird formatiert und, falls nötig, beschrieben. Wird man aufgefordert, die Diskette(n) einzulegen, sollte man nach dem Einlegen für einen kurzen Moment warten, da sonst Amsdos die Disketten noch nicht registriert hat. Der Programmteil, der für das Erkennen des Formates verantwortlich ist, ist sehr empfindlich und wird dann den ersten Track für nicht formatiert erklären, so daß die Kopie unvollständig ist.

Zum Programm

COPY ALL DISKS besteht aus zwei Teilen, einem BASIC-Hauptteil (Listing 1) und einem BASIC-Lader (Listing 2), der nach dem Start das benötigte Binär-File COPYALL.BIN herstellt. Der BASIC-Lader sollte vor dem ersten Start auf Diskette gesichert werden, andernfalls bekommen Ihre Finger Überstunden auferlegt.

Ach ja, eines noch. COPY ALL DISK heißt nicht unbedingt, daß jede Diskette kopiert wird. Der Kopierschutz kommerzieller Programme wird von COPY ALL DISK nicht bewältigt. Für Ihre Arbeitsdisketten und sonstige Kopien reicht das Programm jedoch allemal aus.

(Stefan Riege/jb)

für 464-664-6128

```
100 REM ** COPY ALL DISCS **                    [1714]
110 REM * by  Stefan Riege *                    [575]
120 REM BASIC CTRL  PROGRAMM                    [1027]
130 :                                           [174]
140 REM init + prg loop                         [1867]
150 :                                           [174]
160 MODE 2:PEN 1:PAPER 0:INK 0,0:BORDER 0:      [5883]
INK 1,26:CLEAR:DEFINT a-z:¦DISC
170 mc=&A400:tdat!=&3700:DIM form(6,9):tr=      [6228]
0:erfl=&A4F7:offset=erfl+1
180 CALL &BB00:GOSUB 920:IOBuffer=&BE80         [2700]
```

Listing "Copy all Disk"

```
190 tr.1=0:tr.2=39:copy.art=1:lw=1:laufw1=      [4071]
&A48D:laufw2=&A4C0
200 PRINT TAB(10)CHR$(24)" COPY ALL DISCS       [5327]
"CHR$(24)" written by Stefan Riege";
210 PRINT" (C) 1986,87 - v2.05":PRINT STRI      [4012]
NG$(80,"_"):WINDOW 9,80,4,25
220 PRINT "Weil das Format der Quelldisket      [6421]
te immer wieder neu erkannt und"
230 PRINT "kopiert werden muss, ist die Ue      [6054]
bertragung relativ langsam !!!!"
240 WINDOW 1,80,6,25:PRINT STRING$(80,"_")      [3429]
:WINDOW 1,80,10,25
250 menu=1                                      [421]
260 WHILE menu                                  [1068]
270  CLS:PRINT"HAUPTMENUE":PRINT                [1429]
```

Listing "Copy all Disk"


```
280  PRINT"S      - Start-Track    :"tr.1     [2311]
290  PRINT"E      - End-Track      :"tr.2:PR  [2698]
INT
300  PRINT"A      - Anzahl Drives :"lw:PRIN  [2207]
T
310  PRINT"C      - Copy-Format    : ";       [2281]
320  IF copy.art=1 THEN PRINT"Normal-Copy"   [3416]
 ELSE PRINT"CAD-Copy"
330  PRINT:PRINT"ENTER - Start Copy"          [1841]
340  PRINT"P      - Programmende"             [1919]
350  a$="":WHILE a$="":a$=UPPER$(INKEY$):W   [2173]
END
360  IF a$="P" THEN MODE 2:END               [2045]
370  IF a$=CHR$(13) THEN menu=0              [1059]
380  IF a$="S" THEN LOCATE 25,3:INPUT"    " [2971]
,tr.1
390  IF a$="E" THEN LOCATE 25,4:INPUT"    " [2154]
,tr.2
400  IF a$="A" THEN LOCATE 25,6:INPUT"    " [3006]
,lw
410  IF a$="C" THEN copy.art=3-copy.art     [1888]
420 WEND                                     [390]
430 :                                        [174]
440 REM eigentliches Kopierprogramm          [872]
450 :                                        [174]
460 CLS:IF lw=1 THEN 490                     [1544]
470 PRINT"Please insert Source Disk into D   [4813]
rive A,"
480 PRINT"Destination Disk into Drive B _"   [3939]
CHR$(143):CALL &BB18
490 FOR l1=tr.1 TO tr.2 STEP 6               [1219]
500  tr=l1:etr=l1+5:IF etr>tr.2 THEN etr=t   [1942]
r.2
510  GOSUB 1000:CLS:a$="source":b$="read":   [4400]
GOSUB 780:GOSUB 580:GOSUB 720
520  PRINT:a$="destination":b$="writ":GOSU   [4335]
B 780:GOSUB 870:GOSUB 960
530 NEXT l1                                  [435]
540 GOTO 250                                 [419]
550 :                                        [174]
560 :                                        [174]
570 REM get format                           [995]
580 IF copy.art=1 THEN IF tr<>tr.1 THEN RE   [2057]
TURN
590 FOR i=1 TO 6:track=tr+i-1:IF track<=et   [1360]
r THEN GOSUB 610
600 GOSUB 1000:NEXT i:RETURN                 [2062]
610 l2=0                                     [404]
620 ¦GETFORM,track:IF PEEK(erfl)<>0 THEN l   [3482]
2=l2+1:IF l2<>5 THEN 620 ELSE n=0:GOTO 680
630 m=256:buff=offset:xbuff=buff             [1727]
640 p=PEEK(buff):IF p=m THEN 670             [1805]
650 IF p<m THEN m=p:xbuff=buff               [1213]
```

Listing "Copy all Disk"

```
660 buff=buff+1:GOTO 640                     [1304]
670 n=buff-xbuff:IF n>9 OR (n<8 AND n<>0)    [2088]
THEN n=9
680 IF n=0 THEN FOR w=1 TO 9:form(i,w)=-1:   [2460]
NEXT:RETURN
690 IF n<>0 THEN FOR s=0 TO n-1:form(i,s+l   [4507]
)=PEEK(s+xbuff):NEXT:RETURN
700 :                                        [174]
710 REM read tracks                          [1232]
720 dat!=tdat!:FOR i=l TO 6:track=tr+i-1     [1922]
730 IF track<=etr AND form(i,1)<>-1 THEN G   [2567]
OSUB 750
740 NEXT i:RETURN                            [719]
750 FOR k=1 TO 9:¦READ,track,form(i,k),dat   [5208]
!:dat!=dat!+512:GOSUB 1000:NEXT:RETURN
760 :                                        [174]
770 REM insert disk..                        [1222]
780 IF lw=2 THEN GOTO 810                    [1541]
790 PRINT"Please insert "a$" disc into dri   [6362]
ve A then press any key:_"CHR$(143);
800 CALL &BB18:PRINT CHR$(8)" ":GOTO 830     [1854]
810 IF a$="source" THEN laufw=0 ELSE laufw   [3280]
=1
820 POKE laufw1,laufw:POKE laufw2,laufw      [2309]
830 PRINT:PRINT b$"ing track"tr"-"etr        [1745]
840 RETURN                                   [555]
850 :                                        [174]
860 REM format tracks                        [1047]
870 FOR i=1 TO 6:track=tr+i-1:IF track<=et   [2448]
r AND form(i,1)<>-1 THEN GOSUB 890
880 NEXT i:RETURN                            [719]
890 FOR f=1 TO 9:POKE IOBuffer+f-1,form(i,   [5255]
f):NEXT:¦FORMAT,track:GOSUB 1010:RETURN
900 :                                        [174]
910 REM load mc                              [782]
920 IF PEEK(mc)=&21 THEN RETURN ELSE SYMBO   [4951]
L AFTER 256:MEMORY mc-1
930 LOAD"copyall.bin",mc:CALL mc:RETURN      [2303]
940 :                                        [174]
950 REM write tracks                         [752]
960 dat!=tdat!:FOR i=1 TO 6:track=tr+i-1     [1922]
970 IF track<=etr AND form(i,1)<>-1 THEN G   [2519]
OSUB 990
980 NEXT i:RETURN                            [719]
990 FOR k=1 TO 9:¦WRITE,track,form(i,k),da   [4825]
t!:dat!=dat!+512:GOSUB 1000:NEXT:RETURN
1000 REM auf Abbruch pruefen                 [1531]
1010 IF INKEY$<>CHR$(3) THEN RETURN          [2451]
1020 PRINT:PRINT"*** ^C abgebrochen !!! **   [2431]
*"
1030 PRINT:PRINT"Druecke ENTER":WHILE INKE   [4886]
Y$<>CHR$(13):WEND
1040 GOTO 250                                [419]
```

Listing "Copy all Disk"

```
100 REM BASIC LOADER fuer COPYALL.BIN        [3043]
110 REM --     1986 by Stefan Riege --       [2274]
120 REM                                      [272]
130 :                                        [174]
140 MODE 1:RESTORE:DEFINT a-z:mc=&A400       [2185]
150 SYMBOL AFTER 256:MEMORY mc-1             [1610]
160 zeile=1000                               [659]
170 :                                        [174]
180 FOR a=1 TO 26                            [926]
190  pr=0                                    [117]
200  FOR b=1 TO 10                          [1234]
210   READ by$                               [744]
220   POKE mc,VAL("&"+by$)                   [571]
230   pr=pr+PEEK(mc)                         [693]
240   mc=mc+1                                [686]
250  NEXT b                                  [378]
260  READ pruef                              [378]
270  IF pr<>pruef THEN 500                   [920]
280  zeile=zeile+10                          [568]
290 NEXT a                                   [383]
300 :                                        [174]
310 PRINT"Alles OK - Bitte Diskette einlegen,"  [3273]
320 PRINT"um MC zu speichern !"              [2752]
330 CALL &BB18                               [389]
340 SAVE"copyall.bin",b,&A400,mc-&A400       [1219]
350 END                                      [110]
360 :                                        [174]
500 PRINT"Data-Fehler in Zeile"zeile:STOP    [3383]
```

Listing "Copy all Disk"

```
510 :                                        [174]
1000 DATA 21,09,A4,01,0D,A4,C3,D1,BC,00,976  [2249]
1010 DATA 00,00,00,1B,A4,C3,32,A4,C3,73,910  [1618]
1020 DATA A4,C3,87,A4,C3,9B,A4,47,45,54,1396 [1126]
1030 DATA 46,4F,52,CD,52,45,41,C4,57,52,1017 [1955]
1040 DATA 49,54,C5,46,4F,52,4D,41,D4,00,939  [1538]
1050 DATA FE,01,C0,11,FF,01,CD,5E,A4,1E,1213 [2617]
1060 DATA 00,DD,56,00,DF,67,A4,21,F8,A4,1242 [1537]
1070 DATA E5,DF,6A,A4,E1,30,0B,3A,51,BE,1335 [1883]
1080 DATA 77,23,7D,FE,1F,20,EF,AF,32,F7,1307 [1688]
1090 DATA A4,11,00,0A,7B,DF,6D,A4,7A,DF,1155 [2385]
1100 DATA 70,A4,C9,63,C7,07,6C,C5,07,72,1208 [1849]
1110 DATA CA,07,03,C6,07,FE,03,C0,D5,E1,1304 [1739]
1120 DATA 1E,00,DD,56,04,DD,4E,02,DF,84,997  [2080]
1130 DATA A4,C9,66,C6,07,FE,03,C0,D5,E1,1559 [1783]
1140 DATA 1E,00,DD,56,04,DD,4E,02,DF,98,1017 [2277]
1150 DATA A4,C9,4E,C6,07,FE,01,C0,DD,56,1402 [2185]
1160 DATA 00,21,D3,A4,06,09,7A,77,23,23,734  [1448]
1170 DATA 23,23,10,F9,21,D5,A4,11,80,BE,1080 [1788]
1180 DATA 06,09,1A,77,13,23,23,23,23,10,335  [2443]
1190 DATA F7,1E,00,3A,D3,A4,57,3A,D5,A4,1232 [2269]
1200 DATA 4F,21,D3,A4,DF,D0,A4,C9,52,C6,1563 [2355]
1210 DATA 07,00,00,00,02,00,00,00,02,00,11   [1226]
1220 DATA 00,00,02,00,00,00,02,00,00,00,4    [1551]
1230 DATA 02,00,00,00,02,00,00,00,02,00,6    [1311]
1240 DATA 00,00,02,00,00,00,02,00,00,00,4    [1551]
1250 DATA 00,00,00,00,00,00,00,00,00,00,0    [976]
```

Listing "Copy all Disk"

664/6128-Emulator

für den CPC 464

Jeder CPC 464 Besitzer, der schon mal versucht hat, CPC 6128 BASIC-Software zu laden, wird spätestens beim Programmablauf mit einem SYNTAX ERROR belohnt. Der Grund liegt in den verschiedenen Betriebssystemen der Rechner. Diese Inkompatibilität kann aber mit unserem Emulator abgeschafft werden.

Das Programm besteht aus zwei Teilen, dem Lader für das erzeugte Binärfile und einem Datalader, der das Binärfile (EMU.BIN) erzeugt.

Speichern Sie zuerst das Ladeprogramm ab. Danach sollten Sie den Datalader abspeichern und anschließend starten. Das erzeugte Binärfile ist sehr kurz (1KB), und damit auch sehr schnell geladen, was besonders den Kassettenbesitzern zugute kommt.

Echte Befehle

Die neuen Kommandos werden nicht als RSX-Befehle definiert, sondern als "echte" BASIC-Befehle eingebunden! Das heißt, es können Programme vom CPC 664/6128 eingeladen und gestartet werden. Sollten Probleme beim Einladen entstehen, können die Programme, wenn sie im ASCII-Format (SAVE "name",a) abgespeichert wurden, problemlos eingeladen werden.

Befehlsliste

CLEAR INPUT
– löscht den Tastaturpuffer
Gibt eine Fehlermeldung auf dem CPC 464 aus, wenn das Programm nicht im ASCII-Format vorliegt. Drücken Sie in diesem Falle einfach die ENTER-Taste und starten Sie das Programm neu; der Emulator sorgt automatisch für eine Korrektur.

COPYCHR$(#WINDOW)
– liest das Zeichen an der aktuellen Cursorposition im angegebenen Textfenster.

CURSOR Schalter1,Schalter2
– damit kann der Cursor während eines Programmablaufs ein- oder ausgeschaltet werden.

DEC$(numerischer Ausdruck,Formatstring)
– diese Funktion ist eigentlich schon auf dem CPC 464 vorhanden, aber aufgrund eines Fehlers im Interpreter müssen nach DEG$ immer zwei offene Klammern gesetzt werden. Das wird jetzt umgangen.

Fill Farbstift
– schnelles Fill. Der Füllvorgang beginnt immer an der aktuellen Position des Grafikcursors, er muß also vorher mit MOVE x,y in die Fläche gesetzt werden.

FRAME
– synchronisiert die Bildschirmausgabe mit dem Strahlrücklauf des Monitors.

GRAPHICS PAPER Farbstift
– bestimmt die Farbe des Grafikhintergrundes bei Textausgaben auf Grafikcursorposition (nach TAG) und bei Anwendung des MASK-Befehls.

GRAFICS PEN Farbstift,Hintergrund
– Wählt die Schreibfarbe und den Hintergrundmodus für Grafik. Der zweite Parameter hat folgende Bedeutung:
0 = normal überschreibend
1 = transparent

MASK Bitmaske
– eine Bitmaske ist ein Wert von 0 - 255, der den Status von jeweils acht benachbarten Bildpunkten beim Zeichnen einer Linie bestimmt. Damit können beliebig unterbrochene beziehungsweise gestrichelte Linien erzeugt werden.

PEN #WINDOW,Farbstift,Hintergrund
– Wie beim CPC 464, nur kann jetzt mit dem letzten Parameter der Hintergrundmodus für Textausgaben gewählt werden (0= normal, 1= transparent). Von den letzten beiden Werten kann wahlweise einer entfallen.

MERGE und CHAIN MERGE "Dateiname"
– Diese Kommandos laufen jetzt auch bei Diskettenbetrieb einwandfrei.

Parameterübergabe bei CALL und RSX
– bei CALL und RSX-Kommandos können Strings direkt ohne Umwege übergeben werden. Zum Beispiel:
|ERA,"TESTPROG.BAK" (Matthias Uphoff/cd)

für 464

```
10 'Lader fuer 664/6128-Emulator          [1585]
20 SYMBOL AFTER 256                       [1408]
30 MEMORY &A1FF                           [450]
40 SYMBOL AFTER 240                       [1392]
50 LOAD"emu.bin",&A200                    [2257]
60 CALL &A200                             [597]
70 PRINT"BASIC 1.1 installiert"           [1931]

------------------------------------------

10 MEMORY &A1FF                           [450]
20 FOR adr=&A200 TO &A639 STEP 8          [1122]
30   FOR i=adr TO adr+ 7                  [1011]
40     READ b$                            [315]
50     byte=VAL("&"+b$)                   [465]
60     POKE i,byte                        [294]
70   NEXT i                               [375]
80 NEXT adr                               [547]
```

Listing "664/6128-Emulator"

```
90 SAVE"emu.bin",b,&A200,&439:END         [1466]
100 DATA 3E,C3,32,16,AC,21,83,A2          [1312]
110 DATA 22,17,AC,32,19,AC,21,8E          [1277]
120 DATA A2,22,1A,AC,32,07,AC,21          [1313]
130 DATA 9A,A2,22,08,AC,32,04,AC          [1745]
140 DATA 21,B4,A4,22,05,AC,32,0D          [1184]
150 DATA AC,21,2A,A3,22,0E,AC,32          [1627]
160 DATA 01,AC,21,AB,A2,22,02,AC          [1019]
170 DATA 32,90,BB,21,D8,A2,22,91          [1575]
180 DATA BB,32,DE,BB,21,11,A3,22          [1810]
190 DATA DF,BB,32,C0,BB,21,F5,A2          [1232]
200 DATA 22,C1,BB,32,C3,BB,21,FA          [1543]
210 DATA A2,22,C4,BB,21,53,A3,22          [1537]
220 DATA E3,BD,21,62,A3,22,E9,BD          [2104]
230 DATA 3A,80,BC,FE,C3,C8,32,36          [1770]
240 DATA A6,2A,81,BC,22,37,A6,3E          [944]
250 DATA C3,32,80,BC,21,B2,A2,22          [1562]
260 DATA 81,BC,C9,11,E1,A5,CD,27          [1497]
270 DATA E3,D0,F1,C3,61,DF,21,E1          [1285]
280 DATA A5,CD,13,E3,D0,7E,23,C1          [1633]
290 DATA C1,C9,FE,C5,D0,D6,BA,D1          [751]
300 DATA EB,4F,06,00,21,1C,A6,09          [1413]
310 DATA C3,BB,DD,21,91,B2,75,C3          [1126]
320 DATA 68,A2,E5,3A,36,A6,32,80          [1430]
330 DATA BC,2A,37,A6,22,81,BC,CD          [1480]
```

Listing "664/6128-Emulator"


```
340 DATA 80,BC,21,80,BC,36,C3,21          [1421]
350 DATA B2,A2,22,81,BC,E1,D8,C8          [1269]
360 DATA FE,1A,37,3F,C0,B7,37,C9          [1051]
370 DATA CD,D5,A5,F3,D9,CB,59,D9          [1082]
380 DATA FB,C0,E1,E3,CD,55,DD,30          [1455]
390 DATA 0A,3E,02,CD,FB,C1,E5,CD          [2199]
400 DATA 9F,BB,E1,E3,E9,CD,DB,A5          [1355]
410 DATA 18,03,CD,DE,A5,F3,D9,CB          [1527]
420 DATA 59,D9,FB,C0,E1,E3,CD,55          [1531]
430 DATA DD,DC,0E,A3,E3,E9,CD,4B          [2240]
440 DATA C2,CD,D8,A5,F3,D9,CB,59          [1024]
450 DATA D9,FB,C0,CD,55,DD,D0,3E          [1468]
460 DATA 04,CD,FB,C1,E5,CD,59,BC          [1153]
470 DATA E1,C9,2B,7E,FE,20,28,FA          [1225]
480 DATA FE,2C,C0,2B,E5,2A,34,AE          [1763]
490 DATA CD,3F,DD,21,28,A6,CD,AA          [1804]
500 DATA FF,E1,38,08,FE,BB,C0,CD          [1770]
510 DATA 93,BB,18,03,CD,E1,BB,D1          [754]
520 DATA C3,4F,D0,3A,2F,A6,2F,32          [1266]
530 DATA 33,A6,CD,3C,18,AF,32,33          [714]
540 DATA A6,C9,3A,33,A6,B7,20,0F          [921]
550 DATA 3A,39,B3,B8,C2,CC,B1,3A          [605]
560 DATA 30,A6,B7,CA,CC,B1,C9,3A          [1876]
570 DATA 46,B3,B7,3A,2F,A6,20,11          [1515]
580 DATA 07,32,2F,A6,3A,38,B3,47          [1267]
590 DATA DA,CC,B1,3A,39,B3,47,18          [1282]
600 DATA DE,E5,D5,C5,21,CF,B1,5F          [1766]
610 DATA 3A,07,B2,ED,44,47,AF,CB          [1472]
620 DATA 01,30,05,CB,03,30,01,B6          [2015]
630 DATA 23,10,F4,4F,7B,32,2F,A6          [960]
640 DATA 3A,30,A6,B7,20,0A,2A,38          [1225]
650 DATA B3,7D,AC,A1,AC,C1,18,01          [1524]
660 DATA F1,47,D1,E1,C3,CC,B1,CA          [1180]
670 DATA A3,FE,CD,86,0C,32,3A,B3          [1271]
680 DATA CD,1A,16,CD,FF,16,D2,68          [1264]
690 DATA A4,E5,FD,E1,D5,DD,E1,CD          [1183]
700 DATA A9,0B,3A,3A,B3,57,AE,A1          [1195]
710 DATA 28,7E,3A,38,B3,5F,AE,A1          [1188]
720 DATA 28,76,06,00,FD,23,CD,2D          [458]
730 DATA 0C,3A,34,B3,FD,95,38,52          [1326]
740 DATA 7A,AE,A1,28,4D,7B,AE,A1          [2160]
750 DATA 20,EA,18,46,AE,77,CB,20          [1201]
760 DATA E5,CB,01,DC,05,0C,7A,AE          [1551]
770 DATA A1,28,12,7B,AE,A1,28,0D          [1077]
780 DATA CB,C0,CB,48,20,07,DD,2B          [1427]
790 DATA DF,8A,A4,DD,23,CB,A0,CB          [1782]
800 DATA 09,E1,E5,CB,09,DC,F9,0B          [984]
810 DATA 7A,AE,A1,28,12,7B,AE,A1          [1767]
820 DATA 28,0D,CB,E0,CB,68,20,07          [733]
830 DATA DD,23,DF,8A,A4,DD,2B,CB          [1367]
840 DATA 01,E1,3A,36,B3,FD,95,30          [1917]
850 DATA 0F,FD,2B,CD,13,0C,7B,AE          [1894]
860 DATA A1,28,05,7A,AE,A1,20,A4          [1286]
870 DATA 2A,34,A6,7D,B4,C8,2B,22          [1518]
880 DATA 34,A6,DF,78,A4,C3,D3,A3          [1234]
890 DATA 7B,A4,FD,2A,89,AE,2B,56          [2087]
```

Listing "664/6128-Emulator"

```
900 DATA 2B,5E,2B,22,89,AE,6E,26          [1822]
910 DATA 00,C9,8D,A4,FD,C5,D5,2A          [466]
920 DATA 89,AE,01,03,00,CD,18,F6          [1345]
930 DATA 38,17,EB,22,89,AE,DD,E5          [847]
940 DATA D1,FD,E5,C1,2B,72,2B,73          [1147]
950 DATA 2B,71,2A,34,A6,23,22,34          [1282]
960 DATA A6,D1,C1,C9,FE,03,28,41          [1373]
970 DATA FE,AB,28,57,FE,9F,CA,4A          [1386]
980 DATA A5,FE,2C,28,03,FE,28,C0          [1562]
990 DATA 3E,02,B9,20,64,F1,C1,F5          [2185]
1000 DATA 3E,F8,B8,C0,79,FE,EE,28         [1224]
1010 DATA 15,FE,F5,C0,7E,2B,FE,72         [1151]
1020 DATA 20,FA,7E,3C,20,F6,23,CD         [1430]
1030 DATA 3F,DD,CD,37,DD,28,F1,CD         [999]
1040 DATA FB,CE,CD,37,DD,2C,C3,F5         [1746]
1050 DATA F8,3E,0D,BB,C0,2A,34,AE         [1472]
1060 DATA CD,3F,DD,FE,7C,28,03,FE         [1396]
1070 DATA 83,C0,E1,2A,C2,B0,CD,21         [556]
1080 DATA FB,37,C9,3E,05,BB,C0,3E         [1681]
1090 DATA 1A,91,C0,B8,C0,F1,CD,F5         [1012]
1100 DATA C1,CD,D3,C1,CD,37,DD,29         [526]
1110 DATA CD,60,BB,E5,CD,19,FA,E1         [1119]
1120 DATA C9,3E,49,B9,C0,F1,C1,F5         [872]
1130 DATA 3E,D0,B8,C0,3E,93,B9,C0         [1553]
1140 DATA F1,E5,00,00,AF,CD,0A,FF         [1716]
1150 DATA E1,C9,F1,C1,F5,3E,C8,B8         [1410]
1160 DATA C0,3E,D6,B9,C0,3E,8B,BE         [998]
1170 DATA C0,F1,11,61,A5,1B,C3,DA         [846]
1180 DATA C8,06,00,FF,FF,C9,00,CD         [2264]
1190 DATA 55,DD,38,10,3E,02,CD,FB         [1856]
1200 DATA C1,B7,CC,84,BB,C4,81,BB         [393]
1210 DATA CD,55,DD,D0,3E,02,CD,FB         [1575]
1220 DATA C1,B7,CA,7E,BB,C3,7B,BB         [668]
1230 DATA 23,01,E4,BB,FE,BA,CA,18         [2158]
1240 DATA C2,FE,BB,C2,C6,DD,CD,55         [917]
1250 DATA DD,38,0A,CD,4B,C2,CD,D8         [599]
1260 DATA A5,CD,55,DD,D0,3E,02,CD         [2270]
1270 DATA FB,C1,32,30,A6,C9,CD,4B         [889]
1280 DATA C2,E5,DF,C7,A3,E1,C9,CD         [950]
1290 DATA 55,DD,38,0A,CD,67,CE,32         [1889]
1300 DATA 2F,A6,CD,55,DD,D0,3E,02         [1892]
1310 DATA CD,FB,C1,32,31,A6,C9,CD         [2316]
1320 DATA 09,BB,38,FB,C9,CF,A9,92         [1262]
1330 DATA CF,F6,97,CF,F4,95,CF,F1         [723]
1340 DATA 95,46,49,4C,CC,DD,47,52         [1078]
1350 DATA 41,50,48,49,43,D3,DE,4D         [895]
1360 DATA 41,53,CB,DF,46,52,41,4D         [1794]
1370 DATA C5,E0,43,55,52,53,4F,D2         [1101]
1380 DATA E1,43,4C,45,41,52,20,49         [1288]
1390 DATA 4E,50,55,D4,E2,43,4F,50         [1622]
1400 DATA 59,43,48,52,A4,7E,44,45         [1573]
1410 DATA 52,D2,49,00,AE,A5,88,A5         [952]
1420 DATA B7,A5,19,BD,67,A5,CF,A5         [1232]
1430 DATA 94,95,AE,AF,BC,BD,00,FF         [1309]
1440 DATA 00,FF,FF,00,00,00,00,00         [1953]
1450 DATA 00,00,00,00,00,00,00,00         [1005]
```

Listing "664/6128-Emulator"

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 unsererseits.
5) Behalten Sie auf jeden Fall eine Kopie Ihres Programms, 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 uns die 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 I und i oder O 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 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 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 Rechner usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre »PC International« auf Ihre Beiträge.

Einsendung bitte an: DMV – Daten & Medien Verlagsges.mbH, Fuldaer Straße 6, 3440 Eschwege

PAGE

Eine Text-Grafik-Schnittstelle

DTP (Desktop Publishing) ist zu einem Schlagwort geworden und inzwischen in aller Munde. Doch nicht für alle Zwecke ist es erforderlich, die meist sehr komplizierten und teuren DTP-Programme zu benutzen. PAGE kann für einige Aufgaben schon vollauf genügen.

Text und Grafik

Bei einfacheren Ausdrucken sind die wichtigsten Gestaltungselemente meist Abbildungen wie Zeichnungen oder Fotos. Wollten Sie nicht auch schon einmal Text und Grafik kombinieren?

Dann haben wir hier DAS ideale Programm für Sie: Sie genießen die Vorzüge einer Textverarbeitung, wie zum Beispiel CONTEXT (Blocksatz und Textbearbeitung/Editieren) und die Vorzüge einer guten Grafikverarbeitung, wie zum Beipiel das OCP Art Studio (Grafikbearbeitung, Fensterbearbeitung, mehrere Zeichensätze, Window-Library etc.).

Das hier vorliegende Programm kombiniert beides und ist somit bestens geeignet für Illustrationen von Texten, Dokumentationen und (mit neuem Farbband!) für ein komplettes Layout von DIN-A4-Seiten für Schülerzeitungen.

Minimal-DTP

Sie fragen nach dem WIE? Das ist so verblüffend einfach, beinahe schon primitiv: Wir haben hier ein Programm, das eine Schnittstelle ("Software-Interface") zwischen Textverarbeitung und Grafikverarbeitung, die Textfiles (von CONTEXT) in Screen-Files (für OCP) umwandelt. Im Gegensatz zu teuren DTP-Programmen geschieht die Bearbeitung aber absolut komfortabel und wie man es gewohnt ist: Sie schreiben Ihren Text mit CONTEXT und lassen am Bildschirm überall dort, wo nachher Grafik erscheinen soll, einfach den Platz frei. Danach adaptieren Sie die Textdatei in Screens für OCP. Ein Screen hat ja bekanntlich 25 Zeilen und eine CONTEXT-Seite hat 64 Zeilen. Also sind drei Screens zur Darstellung einer Seite nötig.

Danach können Sie diese drei Screens mit OCP grafisch mit allen Ihnen zur Verfügung stehenden Tricks bearbeiten. Zum Laden müssen Sie vorher den Modus auf MODE 2 einstellen. Nach der Bearbeitung kann die Seite ausgedruckt werden: Diese drei bearbeiteten Screens werden dazu hintereinander per "Print"-Menü ausgedruckt. Den Drucker dürfen Sie während und zwischen der einzelnen Hardcopy-Phasen allerdings nicht verändern, damit ein nahtloser Ausdruck gewährleistet ist.

What to do?

Wenn Sie das Programm gestartet haben, müssen Sie zuerst die Diskette mit der Textdatei einlegen. Danach sollen Sie folgende Angaben machen:

Quelldatei (Text-File) Wenn Sie keine Extension angeben, nimmt das Programm automatisch ".TXT" als Anhang. Die Datei muß nach Eingabe des Namens bereits im Laufwerk bereit sein.

Zieldatei (Screen-File) Hier darf nur ein siebenstelliger Name eingegeben werden; es werden noch die Nummer des Screens (1-3) und die Extension ".SCR" angefügt.

Zeichensatz (Character Set) Bei diesem Punkt ist die Standardeinstellung "I"(International). Sie kann mit "I" oder ENTER/RETURN übernommen werden. Den deutschen Zeichensatz stellt man mit der Taste "G" (German) ein.

Man kann aber auch als Option OCP-Font-Files ("????????.FNT") mit der Taste "O" zur Darstellung laden; Sie werden dann um den Namen der Datei gebeten: Wenn Sie keine Extension angeben, dann nimmt das Programm automatisch ".FNT" als Anhang. Die Datei muß nach der Eingabe im Diskettenlaufwerk bereit sein.

Jetzt werden Sie aufgefordert, die Diskette für die drei Zieldateien (Screen-Files) einzulegen. Den Rest macht der Computer alleine: Er stellt jetzt jeweils 25 Zeilen am Bildschirm dar und speichert dann die jeweilige Datei ab.

CONTEXT-Steuerzeichen

Die normalen Context-Steuerzeichen können im Text enthalten sein: "Fett", "unterstrichen" und "vergrößert" werden richtig dargestellt. Statt der kleinen Schrift (CTRL+'<') ist hier aus technischen Gründen aber die Schriftart light (dünn) implementiert.

Statt der hohen Indices erhält man doppelte Höhe (das bedeutet, daß die Zeichen in der Zeile darüber auch benutzt werden, also Platz einsparen!), anstelle der tiefen Indices Kursivschrift. Diese Vereinbarung war einfacher zu realisieren und bietet bei weitem bessere Möglichkeiten zur normalen Textgestaltung.

Geheimtip

Am effektivsten ist das Programm in Verbindung mit CONTEXT, OCP Art Studio und mit "Ultraschrift" aus Heft 2/88 für die Überschriften.
Da eine DIN-A4-Seite beim OCP Art Studio aber keinen formartfüllenden Ausdruck ergibt, kann man die einzelnen Ausdrucke für Fotokopien auch von Hand auf einem DIN-A4-Blatt layouten, so wie es bei den meisten Schülerzeitungen noch praktiziert wird.

Die Ergebnisse sind für die meisten Ansprüche vollauf befriedigend...

(Eckehart Röscheisen/cd)

für 464-664-6128

```
1000 ' PAGE 3.2 - CPC 464/664/6128          [1674]
1010 '            PC International          [846]
1020 '            (P) January 1988          [1437]
1030 '            H.E. Roescheisen          [1485]
1040 :                                      [174]
1050 REM "PAGE    .BIN" - CPC6128 DATA      [2429]
1060 :                                      [174]
1070 CLEAR:DEFINT a-z                       [767]
1080 SYMBOL AFTER 256                       [1408]
1090 MEMORY &9FFF                           [134]
1100 RESTORE 1250                           [749]
1110 dl=1250                                [492]
1120 SYMBOL AFTER 32:x!=HIMEM+1             [2242]
1130 FOR adr=&A000 TO &A147 STEP 8          [1081]
1140   c=0                                  [348]
1150   FOR i=adr TO adr+7                   [1011]
1160     READ byte                          [703]
1170     c=c XOR byte                       [923]
1180     POKE i,byte                        [294]
1190   NEXT i                               [375]
1200   READ cs                              [534]
1210   IF cs<>c THEN PRINT"DATA error in l [1227]
ine"dl:END
1220   dl=dl+10                             [631]
1230 NEXT adr                               [547]
1240 :                                      [174]
1250 DATA 254,  1,192,221,126,  0, 50, 70, [1400]
 232
1260 DATA 161, 33, 10, 20,183, 40,  3, 33, [968]
  35
1270 DATA  22,160, 34,218,189,201,254, 32, [1941]
 228
1280 DATA 218, 10, 20,205,212, 18, 17, 38, [1487]
 248
1290 DATA 161,  1,  8,  0, 58, 70,161,203, [1238]
 190
1300 DATA  71, 32, 16,203, 79, 40, 24, 65, [830]
 130
1310 DATA 126,203, 63,166, 18, 35, 19, 16, [1756]
  30
1320 DATA 247, 24, 14, 65,126,203, 63,182, [695]
 156
1330 DATA  18, 35, 19, 16,247, 24,  2,237, [1908]
  50
1340 DATA 176,235, 58, 70,161,203,111, 40, [1569]
  10
1350 DATA  24,229, 33, 38,161,  6,  2,203, [1479]
 148
1360 DATA  30, 35, 16,251,  6,  3, 35, 16, [1450]
 224
1370 DATA 253,  6,  3,203, 38, 35, 16,251, [1526]
 221
1380 DATA 225, 58, 70,161,203,103, 40,  4, [1464]
 188
1390 DATA  43, 54,255, 35,203,119, 40, 14, [1220]
  91
1400 DATA  33, 38,161,  6,  8,126, 47,119, [1429]
 142
1410 DATA  35, 16,250, 58, 70,161,  6,  1, [1900]
  19
1420 DATA 203, 87, 40, 39,221, 33, 38,161, [1356]
 232
1430 DATA   6,  8,197,221, 78,  0,  6,  8, [1089]
  86
1440 DATA 203,  9,221,203,  0, 30,221,203, [1097]
 220
1450 DATA  16, 30,221,203,  0, 46,221,203, [1659]
  32
1460 DATA  16, 30, 16,236,221, 35,193, 16, [2059]
 221
1470 DATA 225,  6,  2,203, 95, 40, 24,197, [1011]
 132
1480 DATA   1,  8,  0,229,  9,235,225, 65, [1442]
 174
1490 DATA  43,126, 27, 18, 27, 18, 16,248, [1703]
 189
1500 DATA  17, 24,  0, 25,193, 16,232, 42, [1544]
   3
```

Listing "PAGE"

```
1510 DATA  54,183,229, 33, 38,161, 34, 54, [1501]
 214
1520 DATA 183, 58, 52,183,245,175, 50, 52, [1035]
  82
1530 DATA 183, 79,205,  2,161,121,205, 53, [1316]
  23
1540 DATA  19, 58, 70,161,203, 87, 40,  9, [588]
 115
1550 DATA  14,  2,205,  2,161,121,205, 53, [1701]
 227
```



```
1560 DATA  19,241, 50, 52,183,225, 34, 54, [1770]
 166
1570 DATA 183,201, 58, 70,161,203, 95,200, [1551]
 255
1580 DATA  33, 39,183,229, 58, 42,183,190, [1476]
  77
1590 DATA  40,  8, 58, 44,183,190, 56,  2, [885]
   5
1600 DATA  43, 53,121,197,205, 53, 19,193, [1179]
 136
1610 DATA 225, 53, 43, 52, 12,201,  0,  0, [1077]
  14
1620 DATA   0,  0,  0,  0,  0,  0,  0,  0, [1151]
   0
1630 DATA   0,  0,  0,  0,  0,  0,  0,  0, [1151]
   0
1640 DATA   0,  0,  0,  0,  0,  0,  0,  0, [1151]
   0
1650 DATA   0,  0,  0,  0,  0,  0,  0,  0, [1151]
   0
1660 :                                      [174]
1670 REM headline and copyright             [1520]
1680 :                                      [174]
1690 cpc=PEEK(&BB4F)                        [897]
1700 IF cpc=&78 THEN GOSUB 2800 '(adaption [3341]
 cpc464)
1710 IF cpc=&70 THEN GOSUB 3000 '(adaption [2411]
 cpc664)
1720 SPEED KEY 15,1                         [1264]
1730 INK 0,0:INK 1,19:INK 2,9:INK 3,26:BOR [2979]
DER 0:MODE 1
1740 CALL &A000,42:PEN 2                    [1184]
1750 PRINT CHR$(24)" PAGE "CHR$(24);        [1873]
1760 CALL &A000,8:PEN 1                     [1447]
1770 PRINT" CPC 464 ";                      [1015]
1780 CALL &A000,10:PEN 3                    [963]
1790 PRINT" - A CONTEXT utility V3.2";      [3253]
1800 CALL &A000,0:PEN 2                     [931]
1810 SYMBOL 164,60,66,185,165,185,161,66,6 [2447]
0
1820 PRINT           " 1988 by ER & COMPUTE [4663]
R project Software
1830 CALL &A000,8:LOCATE 11,2:PEN 1         [1479]
1840 IF cpc=&70 THEN PRINT" 664":POKE &B63 [3273]
2,255:GOTO 1890
1850 IF cpc=&74 THEN PRINT"6128":POKE &B63 [2746]
2,255:GOTO 1890
1855 POKE &B4E8,255                         [809]
1860 :                                      [174]
1870 REM init and system inputs             [2949]
1880 :                                      [174]
1890 CALL &A000,0                           [813]
```

Listing "PAGE"

```
1900 LOCATE 1,5                                          [623]
1910 WHILE INKEY$<>"":WEND                               [1786]
1920 PEN 3:PRINT"Name of source text file:               [2706]
 ";
1930 PEN 1:INPUT"",f1$                                   [1336]
1940 IF f1$=""OR LEN(f1$)>12 THEN 1900                   [1546]
1950 IF INSTR(f1$,".")=0 THEN f1$=f1$+".TX               [2446]
T"
1960 PEN 2:LOCATE 1,7                                    [834]
1970 PRINT"Please insert disc with "CHR$(3               [4709]
4)UPPER$(f1$)CHR$(34)"!";
1980 PRINT"Then press any key:_                          [2070]
1990 WHILE INKEY$<>"":WEND                               [1786]
2000 LOCATE 1,8:CALL &BB06:PRINT STRING$(2               [2421]
0,32)
2010 DIM a$[500]                                         [435]
2020 OPENIN"!"+f1$                                       [720]
2030 i=0                                                 [420]
2040 WHILE NOT EOF                                       [1840]
2050    LINE INPUT#9,a$[i]                               [1759]
2060    i=i+1                                            [444]
2070 WEND                                                [390]
2080 CLOSEIN                                             [752]
2090 LOCATE 1,7:PRINT STRING$(40,32):LOCAT               [3289]
E 1,7
2100 PEN 3:PRINT"Name of destination scree               [2154]
n file: ";
2110 WHILE INKEY$<>"":WEND                               [1786]
2120 PEN 1:INPUT"",f2$                                   [1334]
2130 IF f2$=""OR LEN(f2$)>7 OR INSTR(f2$,"               [2937]
.")THEN 2090
2140 PEN 3:LOCATE 1,11                                   [965]
2150 PRINT"Font (German/International/OCP                [4071]
font): ";
2160 PEN 1:PRINT"I                                       [1109]
2170 WHILE INKEY$<>"":WEND                               [1786]
2180 i$=UPPER$(INKEY$)                                   [1006]
2190 IF i$="I"OR i$=CHR$(13)THEN 2470                    [1511]
2200 IF i$<>"O"THEN 2350                                 [569]
2210 LOCATE 39,11:PRINT"O"                               [1385]
2220 PEN 3:LOCATE 1,13                                   [963]
2230 PRINT"Name of OCP font file: ";                     [2685]
2240 WHILE INKEY$<>"":WEND                               [1786]
2250 PEN 1:INPUT"",f3$                                   [1332]
2260 IF f3$=""OR LEN(f3$)>12 THEN 2220                   [1881]
2270 IF INSTR(f3$,".")=0 THEN f3$=f3$+".FN               [1161]
T"
2280 PEN 2:LOCATE 1,15                                   [847]
2290 PRINT"Please insert disc with "CHR$(3               [5210]
4)UPPER$(f3$)CHR$(34)"!";
2300 PRINT"Then press any key:_                          [2070]
2310 WHILE INKEY$<>"":WEND                               [1786]
2320 LOCATE 1,16:CALL &BB06:PRINT STRING$(               [2617]
20,32)
2330 LOAD"!"+f3$,x!                                      [1000]
2340 GOTO 2470                                           [309]
2350 IF i$<>"G"THEN 2180                                 [1240]
2360 LOCATE 39,11:PRINT"G"                               [1464]
2370 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66               [1536]
2380 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C               [2213]
2390 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C               [2189]
2400 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76                 [2270]
2410 SYMBOL 124,&66,0,&3C,&66,&66,&66,&3C                [2226]
2420 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E                [2030]
2430 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6               [2404]
C,&60
2440 :                                                   [174]
2450 REM generate screen from text file                  [1813]
2460 :                                                   [174]
2470 MODE 2:PEN 1:CALL &A000,0                           [2315]
2480 PRINT CHR$(24)" Please insert";                     [2463]
2490 CALL &A000,1:PRINT" disc";                          [1533]
2500 CALL &A000,0:PRINT" to save. "CHR$(24               [1917]
);
2510 CALL &A000,4:PRINT" Then press any ke               [4726]
y ..."
2520 CALL &BB06                                          [393]
2530 CLS                                                 [91]
2540 sa=0                                                [264]
2550 j=0:f=0:l=0:v=0:d=0:i=0                             [1506]
2560 IF a$[j]=""THEN 2700                                [1328]
2570 FOR k=1 TO LEN(a$[j])                               [1828]
2580    e$=MID$(a$[j],k,1)                               [891]
2590    IF v OR k+1>LEN(a$[j])THEN v=0:GOTO              [2885]
 2610
2600    IF MID$(a$[j],k+1,1)=CHR$(255)THEN               [4393]
sa=sa OR 4:v=1 ELSE sa=sa AND 251
```

Listing "PAGE"

```
2610    IF e$=CHR$(30)THEN IF f=0 THEN sa=s              [4222]
a OR 1:f=1 ELSE sa=sa AND 254:f=0
2620    IF e$=CHR$(31)THEN IF l=0 THEN sa=s              [3303]
a OR 2:l=1 ELSE sa=sa AND 253:l=0
2630    IF e$=CHR$(29)THEN IF i=0 THEN sa1=              [3971]
sa:sa=sa OR 34:i=1 ELSE sa=sa1:i=0
2640    IF e$=CHR$(28)THEN IF d=0 THEN sa=s              [3974]
a OR 8:d=1 ELSE sa=sa AND 247:d=0
2650    IF ASC(e$)<>255 AND (ASC(e$)AND 128              [3539]
) THEN sa=sa OR 16 ELSE sa=sa AND 239
2660    CALL &A000,sa                                    [736]
2670    IF e$=CHR$(255) THEN 2690                        [1188]
2680    IF ASC(e$)>31 THEN PRINT CHR$(ASC(e              [2917]
$)AND 127); ELSE PRINT" ";
2690 NEXT k                                              [373]
2700 PRINT:j=j+1                                         [683]
2710 IF j=25 THEN SAVE"!"+f2$+"1.SCR",b,&C               [2592]
000,&4000:CLS
2720 IF j=50 THEN SAVE"!"+f2$+"2.SCR",b,&C               [2729]
000,&4000:CLS
2730 IF j<64 THEN 2560                                   [324]
2740 SAVE"!"+f2$+"3.SCR",b,&C000,&4000                   [2716]
2750 CALL &A000,12:PRINT"Ready."                         [2182]
2760 CALL &A000,0:END                                    [943]
2770 :                                                   [174]
2780 REM adaption cpc version 464                        [2784]
2790 :                                                   [174]
2800 RESTORE 2910                                        [872]
2810 READ a1$,a2$                                        [509]
2820 IF a1$="%"THEN RETURN                               [2338]
2830 a1=VAL("&"+a1$)                                     [541]
2840 a2=VAL("&"+a2$)                                     [812]
2850 b1=UNT(a2)AND &FF                                   [769]
2860 b2=INT(a2/256)AND &FF                               [1443]
2870 POKE a1,b1                                          [189]
2880 POKE a1+1,b2                                        [570]
2890 GOTO 2810                                           [337]
2900 :                                                   [174]
2910 DATA A00A,140C,A019,140C,A01C,12D3                  [1673]
2920 DATA A0D0,B296,A0D7,B296,A0DA,B294                  [1944]
2930 DATA A0DF,B294,A0E7,1334,A0F7,1334                  [1795]
2940 DATA A0FB,B294,A0FF,B296,A109,B286                  [1947]
2950 DATA A10D,B289,A113,B28B,A11D,1334                  [1798]
2960 DATA %,%                                            [126]
2970 :                                                   [174]
2980 REM adaption cpc version 664                        [2713]
2990 :                                                   [174]
3000 RESTORE 3110                                        [689]
3010 READ a1$,a2$                                        [509]
3020 IF a1$="%"THEN RETURN                               [2338]
3030 a1=VAL("&"+a1$)                                     [541]
3040 a2=VAL("&"+a2$)                                     [812]
3050 b1=UNT(a2)AND &FF                                   [769]
3060 b2=INT(a2/256)AND &FF                               [1443]
3070 POKE a1,b1                                          [189]
3080 POKE a1+1,b2                                        [570]
3090 GOTO 3010                                           [455]
3100 :                                                   [174]
3110 DATA A00A,1406,A019,1406,A01C,12D0                  [2192]
3120 DATA A0D0,B736,A0D7,B736,A0DA,B734                  [1688]
3130 DATA A0DF,B734,A0E7,1331,A0F7,1331                  [1342]
3140 DATA A0FB,B734,A0FF,B736,A109,B727                  [1915]
3150 DATA A10D,B72A,A113,B72C,A11D,1331                  [1696]
3160 DATA %,%                                            [126]
```



Listing "PAGE"

Programmschreiber

Ein Programm erzeugt sich (fast) selbst

Übergeben Sie Programmzeilen in einem String, aus denen ein Programm entsteht.

Der RSX-Befehl ‚GEN' ermöglicht das Generieren von BASIC-Zeilen von einem BASIC-Programm aus. Dabei muß eine STRING-Variable die zu generierende Zeile enthalten und dann bei Verwendung des Befehls mittels des STRING-Descriptors,@' als Parameter angegeben werden. Ein Beispiel für die Syntax:

```
10 var$="40 REM Beispiel"
20 |gen,@var$
30 end
```

Startet man das Programm nun einmal und listet es dann, ergibt sich folgendes Bild:

```
10 var$="40 REM Beispiel"
20 |gen,@a$
30 end
40 REM Beispiel
```

Die Zeile 40 wurde während des Programmablaufs eingefügt.

Das MC-Programm beginnt bei Adresse &AF00 und hat eine Länge von 70 Byte. Es kopiert per Blocktransfer den Inhalt der STRING-Variablen in den BASIC-Eingabe-Puffer ab &ACA4 und ruft danach die ROM-Routine ' Statement → Interpretercode ' bei Adresse &E6BC auf. Für Interessierte folgt unten ein kommentiertes Assembler-Listing.

Das Programm ist nur auf dem CPC464 lauffähig, da direkt ins BASIC-ROM gesprungen wird.

Es steht dem Benutzer offen, welche Zeilen er generiert, nur sollten sich Neulinge zunächst darauf beschränken, diese hinter das laufende Programm zu hängen. Will man nämlich eine Zeile mitten ins Programm einfügen, so muß an dieser Stelle ein Platzhalter, ein sogenannter ‚DUMMY', vorhanden sein, beispielsweise ein REMark mit der Länge der später einzufügenden Zeile; diese Technik ist in GENDEMO1.BAS demonstriert.

(Jürgen Schob/cd)

für 464

```
100 '   " |GEN,@var$ "   DATALADER GEN                   [2656]
110 '                                                    [117]
120 '     by J.Schob                                     [591]
130 '                                                    [117]
140 FOR a=&AF00 TO &AF45:READ b$:b=VAL("&"               [4080]
+b$):POKE a,b:c=c+b:NEXT a
150 IF c<>&2010 THEN PRINT"Data-Fehler !!!               [1916]
":END
160 CALL &AF00:PRINT"Alles O.K.!":END                    [2433]
170 DATA 3e,c9,32,00,af,01,0e,af,21,17,af,               [3276]
c3,d1,bc,13,af
180 DATA c3,1b,af,47,45,ce,00,00,00,00,00,               [3731]
fe,01,c0,dd,66
190 DATA 01,dd,6e,00,06,00,4e,23,e5,dd,e1,               [2666]
dd,66,01,dd,6e
200 DATA 00,11,a4,ac,ed,b0,af,12,cd,00,b9,               [3131]
f5,21,a4,ac,cd
210 DATA bc,e6,f1,c3,0c,b9                               [845]
```

```
100 ' GEN-DEMO (1) Listing 2.                            [1594]
110 ' ------------                                       [797]
120 ' (Funktionsplotter)                                 [897]
130 ON ERROR GOTO 230                                    [1527]
140 MODE 2:INK 1,20:INPUT "Geben Sie bitte               [7389]
 die Gleichung des Graphen ein : ",gleich$
150 CLS:ORIGIN 0,200:PLOT 0,200,1:DRAW 0,-               [3231]
200:PLOT 0,0:DRAW 640,0:PLOT 0,0
160 gleich$="250 "+gleich$:|GEN,@gleich$                 [1890]
170 x=0:GOSUB 260:ax=x:ay=y                              [1356]
180 FOR x=0 TO 26 STEP 0.1                               [1444]
190 GOSUB 250 'Zuweisung                                 [1496]
200 IF y<1000 AND ay<1000 THEN IF y>-1000                [3013]
AND ay>-1000 THEN PLOT ax*25,ay*25:DRAW x*
25,y*25
210 ax=x:ay=y:NEXT x                                     [942]
220 CALL &BB06:RUN                                       [1044]
230 'error                                               [561]
240 CLS:PRINT"FUNKTION NICHT DARSTELLBAR !               [2998]
!!":RESUME 220
250 'Platz fuer Funktion                                 [847]
260 '                                                    [203]
270 RETURN                                               [555]
```

```
100 ' GEN-DEMO (2) Listing 3                             [1222]
110 ' ------------                                       [797]
120 ' (data-generator)                                   [938]
130 CLS:INPUT "von : ",a:INPUT "bis : ",b:               [2309]
INPUT "zeile : ",z
140 PRINT:PRINT"Ich arbeite...bitte nicht                [4419]
stoeren.":PRINT
150 b=(32-((b-a)MOD 32))+b:i%=a                          [1770]
160 WHILE i%<b:                                          [927]
```

Listing "Programmschreiber"

```
170             a$=STR$(z)+" data":gc=FRE(" [2476]
")
180             FOR i1%=0 TO 31:a$=a$+STR$( [2841]
PEEK(i%+i1%))+",":NEXT i1%
190             a$=MID$(a$,1,(LEN(a$)-1)):z [3560]
=z+10:|GEN,@a$:i%=i%+32
200 WEND        :                                        [435]
210 PRINT"Schon..."                                      [1403]
220 a$=STR$(z)+" memory"+STR$(a-1)+":for i [8538]
="+STR$(a)+" to "+STR$(b-1)+":read p:poke
i,p:next i":|GEN,@A$:DELETE -220
```

```
1000 '          :  org  &af00      ;          [907]
1010 '  &af00   :  ld   a,C9       ;          [521]
1020 '  &af02   :  ld   (&af00),a  ;     Schu [2956]
tz gegen doppelte Initialisierung
1030 '  &af05   :  ld   bc,&af0e   ;          [1377]
1040 '  &af08   :  ld   hl,&af17   ;          [685]
1050 '  &af0b   :  jp   &bcd1      ;     RSX  [1719]
einbinden
1060 '  &af0e   :  defw &af13      ;          [599]
1070 '  &af10   :  jp   &af1b      ;          [581]
1080 '  &af13   :  defm "GE"       ;     RSX- [1733]
Befehlstabelle
1090 '  &af15   :  defb "N"+&80    ;          [782]
1100 '  &af16   :  defb 0          ;          [376]
1110 '  &af17   :  defs 4          ;     4 By [2431]
te fuer Kernel
1120 '  &af1b   :  cp   1          ;     Begi [1856]
nn von Routine
1130 '  &af1d   :  ret  nz         ;     1 Pa [3396]
rameter ?  Wenn nein --> zurueck
1140 '  &af1e   :  ld   h,(ix+1)   ;          [1193]
1150 '  &af21   :  ld   l,(ix)     ;          [970]
1160 '  &af24   :  ld   b,0        ;          [769]
1170 '  &af26   :  ld   c,(hl)     ;     Laen [1270]
ge von String in bc
1180 '  &af27   :  inc  hl         ;          [536]
1190 '  &af28   :  push hl         ;          [925]
1200 '  &af29   :  pop  ix         ;          [350]
1210 '  &af2b   :  ld   h,(ix+1)   ;          [840]
1220 '  &af2e   :  ld   l,(ix)     ;     Begi [3580]
nn von String in hl
1230 '  &af31   :  ld   de,&ACA4   ;     Ziel [3295]
adresse in de
1240 '  &af34   :  ldir            ;     Stri [2683]
ng in BASIC-Eingabepuffer kopieren
1250 '  &af36   :  inc  hl         ;          [464]
1260 '  &af37   :  ld   (hl),0     ;          [454]
1270 '  &af39   :  call &b900      ;     Uppe [1523]
r-ROM on
1280 '  &af3c   :  push af         ;          [1238]
1290 '  &af3d   :  ld   hl,&ACA4   ;          [1442]
1300 '  &af40   :  call &e6bc      ;     BASI [3790]
C-Statement --> Interpretercode
1310 '  &af42   :  pop  af         ;          [526]
1320 '  &af43   :  jp   &b90c      ;     Uppe [645]
r-ROM off
1330 '                             ;     ...u [1249]
nd zurueck
```

Listing "Programmschreiber"

Hüllenlos

Kassetteninlays selbst gestaltet

Dieses BASIC-Programm ist sehr hilfreich beim Erstellen von Kassettenhüllen. Denn man braucht hierzu nur die Daten, also z.B. Interpret, Titel des Liedes, usw., einzutippen, die Hülle ausdrucken zu lassen und am Rand auszuschneiden. Dann legt man sie in die Plastikschachtel, fertig!

Natürlich kann man auch gleich mehrere dieser Hüllen anfertigen und auf Diskette abspeichern und wieder laden. Die einzelnen Menüpunkte werden durch die Cursortasten angewählt und durch Drücken der kleinen ENTER-Taste ausgeführt. Wofür die einzelnen Menüpunkte gut sind, wird im nachstehenden Teil erklärt.

EINGEBEN:

Wenn man eine neue Hülle erstellen will, gibt man einfach seine Daten ZEILENWEISE ein. Das heißt, nach jeder Zeile muß man die RETURN-Taste drücken, denn das Einlesen geschieht mit dem Befehl INPUT und darf nicht wie eine Textverarbeitung bedient werden. Zum Verlassen dieses Menüpunktes drückt man am Anfang einer neuen Hülle einfach RETURN.

ÄNDERN:

Hat man sich bei der Eingabe einmal vertippt, so kann man in diesem Menüpunkt die entsprechende(n) Zeile(n) wieder verbessern. Alle übrigen Zeilen bleiben durch Drücken von RETURN im Speicher, sonst muß die ganze Zeile neu geschrieben werden.

BLÄTTERN:

Hier kann man sich alle bereits erstellten Hüllen ansehen. Geblättert wird mit den Cursortasten. DIRECTORY zeigt das Directory der Diskette (=CAT).

SPEICHERN:

Alle bereits erstellten Hüllen werden auf der Diskette abgespeichert. Sollten hierbei Diskettenfehler auftreten, werden diese abgefangen und dem Benutzer gemeldet.

LADEN:

Eine bereits erstellte Datei wird wieder von der Diskette geladen.

DRUCKEN:

Alle Hüllen werden ausgedruckt. Bei mehr als drei Hüllen empfiehlt sich Endlospapier. Der Druckertreiber ist für einen EPSON-kompatiblen Drucker vorgesehen.

1 DRUCKEN:

Es wird nur eine bestimmte Hülle ausgedruckt.

MEMORY:

Diese Meldung zeigt Ihnen, wie groß der noch verfügbare Speicher ist, damit Sie nicht plötzlich durch eine Fehlermeldung wie MEMORY FULL überrascht werden und all Ihre Daten verloren wären.

Hinweis:

Das Programm wurde auf einem CPC 6128 entwickelt, läuft aber auch auf dem CPC 464 mit Emulator (befindet sich im Heft).

(Markus Holzhäuser/cd)

```
-------------------------------------------------------------------

  Dark side of the moon:          ! Breakfast in America:
                                  !
  -Speak to me                    ! Gone to Hollywood
  Breathe in the air              ! The logical song
  On the run                      ! Goodby stranger
  Time                            ! Breakfast in Amerika
  The great gig in the sky        ! On darling
  Money                           ! Take the long way home
  Us and them                     ! Is it mine
  Any colour you like             ! Into ...
  Brain demage                    ! I really believe I'm glad
  Eclipse                         ! One of the vision
                                  !
                                  !
-------------------------------------------------------------------

            PINK FLOYD: )) Dark Side of the Moon ((
            SUPERTRAMP: )) Breakfast in America ((

-------------------------------------------------------------------


  TDK  SA-X 90
```

Eine Kassette ohne Beschriftung ist wie eine Suppe ohne Salz und Pfeffer. Geben Sie Ihrer Kassettenhülle das richtige Aussehen.

für 464-664-6128

```
10 '*********************************          [1416]
20 '**       Kassettenhuellen        **         [503]
30 '**          (c) 1987 by          **         [560]
40 '**                               **         [104]
50 '**       Markus Holzhaeuser      **        [1822]
60 '**                               **         [104]
70 '*********************************          [1416]
80 '                                             [117]
90 CLEAR                                         [217]
100 DIM titela$(980),titelb$(980),notizen$       [4156]
(210),namea$(71),nameb$(70)
```

Listing "Hüllenlos"

```
110 free=FRE(0):z=1                              [404]
120 wahl$(1)="Eingeben "                         [1779]
130 wahl$(2)="Aendern  "                         [1908]
140 wahl$(3)="Blaettern"                         [1771]
150 wahl$(4)="Directory"                         [1745]
160 wahl$(5)="Laden    "                         [1360]
170 wahl$(6)="Speichern"                         [919]
180 wahl$(7)="Drucken  "                         [1492]
190 wahl$(8)="1 drucken"                         [1697]
200 MODE 2:MOVE 1,377:DRAWR 552,0:DRAWR 0,       [4366]
-350:DRAWR -552,0:DRAWR 0,350
210 MOVE 1,88:DRAWR 552,0:MOVE 1,135:DRAWR       [3864]
 552,0:MOVE 280,377:DRAWR 0,-243
220 WINDOW #1,2,34,3,16:WINDOW #2,37,68,3,       [1749]
16
230 WINDOW #3,2,68,18,19:WINDOW #4,2,68,21       [3123]
```

Listing "Hüllenlos"


```
,23
240 WINDOW #5,1,79,25,25                         [1428]
250 PRINT CHR$(22);CHR$(1):PEN 3: 'Transpa       [3307]
rente Schrift an
260 FOR i=1 TO 80:LOCATE i,1:PRINT CHR$(20       [2501]
7);:NEXT
270 LOCATE 1,1:PEN 13                            [748]
280 PRINT " Kassettenhuelle einfach gemach       [3709]
t!                 ";
290 PRINT " by  Markus Holzhaeuser "             [1915]
300 LOCATE 72,3:PRINT "MENUE:"                   [1363]
310 LOCATE 72,3:PRINT "_____"                    [1504]
320 PRINT CHR$(22);CHR$(0);: 'Transparente       [3219]
 Schrift aus
330 FOR i=1 TO 8                                 [462]
340 u=4+i*2:LOCATE 72,u:PRINT wahl$(i)           [1398]
350 NEXT                                         [350]
360 CLS #5:LOCATE 1,25:PRINT"Waehlen mit         [3905]
     ";CHR$(24);
370 PRINT "[";CHR$(240);"]";CHR$(24);"           [2845]
,    ";CHR$(24);
380 PRINT "[";CHR$(241);"]";CHR$(24);"           [1992]
und      ";CHR$(24);
390 PRINT "[kl. Enter]";CHR$(24);" -Taste        [1778]
"
400 LOCATE 72,23:PRINT "SPEICHER:"               [2001]
410 LOCATE 71,24:PRINT INT(FRE(0)/free*100       [2577]
)+1;" %  "
420 CLS #1:CLS #2:CLS #3:CLS #4                  [1769]
430 LOCATE 69,18:PRINT " "                       [1820]
440 LOCATE 72,4+z*2:PRINT CHR$(24);wahl$(z       [1416]
)
450 IF INKEY(0)=0 THEN z=z-1:LOCATE 72,4+(       [6350]
z+1)*2):PRINT CHR$(24);wahl$(z+1):LOCATE
72,4+(z*2):PRINT CHR$(24);wahl$(z)
460 IF INKEY(2)=0 THEN z=z+1:LOCATE 72,4+(       [7696]
(z-1)*2):PRINT CHR$(24);wahl$(z-1):LOCATE
72,4+(z*2):PRINT CHR$(24);wahl$(z)
470 IF INKEY(6)=0 THEN CLEAR INPUT:GOTO 52       [2085]
0
480 IF z<1 THEN z=8:LOCATE 72,20:PRINT wah       [1990]
l$(z)
490 IF z>8 THEN z=1:LOCATE 72,6:PRINT wahl       [3735]
$(z)
500 CALL &BB06: 'Zusaetzliche Tastaturabfr       [4763]
age, weil sonst die Tasten zu empfindlich
reagieren.
510 GOTO 450                                     [442]
520 LOCATE 72,4+(z*2):PRINT CHR$(24);wahl$       [2918]
(z)
530 ON z GOTO 550,1490,1810,1440,1060,1270       [1861]
,720,2030
540 '                                            [117]
550 '*****  Eingeben  *****                      [740]
560 '                                            [117]
570 anz=anz+1                                    [1018]
580 CLS #1:CLS #2:CLS #3:CLS #4:CLS #5           [1646]
590 LOCATE 1,25:PRINT "RETURN am Anfang ei       [7475]
ner neuen Huelle fuehrt zurueck ins Menue.
"
600 LOCATE 71,24:PRINT INT(FRE(0)/free*100       [2577]
)+1;" %  "
610 LOCATE 66,18:PRINT CHR$(24);anz;CHR$(2       [1519]
4)
620 INPUT#3,"",namea$(anz)                       [996]
630 IF LEFT$(namea$(anz),1)="" THEN pmax=a       [3865]
nz-1:anz=pmax:GOTO 360
640 INPUT#3,"",nameb$(anz)                       [1257]
650 FOR i=1 TO 14:LINE INPUT #1,"",titela$       [2546]
(i+14*(anz-1)):NEXT
660 FOR i=1 TO 14:LINE INPUT #2,"",titelb$       [2444]
(i+14*(anz-1)):NEXT
670 FOR n=1 TO 3:LINE INPUT #4,"",notizen$       [3076]
(n+3*(anz-1)):NEXT
680 GOTO 570                                     [379]
690 '                                            [117]
700 '*****  Drucken  *****                       [898]
710 '                                            [117]
720 MODE 1:LOCATE 1,12:PRINT"Wenn fertig,        [5045]
bitte eine Taste druecken."
730 LOCATE 8,14:PRINT"(ausser [M] fuer Men       [3413]
ue)"
740 a$=INKEY$:IF a$="" THEN 740                  [1428]
750 IF a$="m" OR a$="M" THEN 200                 [1785]
760 CLS:LOCATE 14,13:PRINT "Ich drucke !"        [2428]
770 '                                            [117]
780 '++++++++++++++++++++++++++++++++++++        [2494]
++++++++++++++++++++++
```

Listing "Hüllenlos"

```
790 'Hier wird Ihr Drucker auf SCHMALSCHRI        [4256]
FT eingestellt.
800 'Passen Sie, wenn noetig bitte diese Z       [4872]
eile Ihrem Drucker an.
810 '                                            [117]
820  PRINT #8,CHR$(27);"x";CHR$(0);CHR$(15       [2438]
)
830 '                                            [117]
840 '++++++++++++++++++++++++++++++++++++        [2494]
++++++++++++++++++++++
850 '                                            [117]
860 FOR x=1 TO pmax                              [927]
870 PRINT #8,"----------------------------       [4326]
----------------------------------------
"
880 FOR i=1 TO 14                                [311]
890 h=i+14*(x-1):la=LEN(titela$(h)):lb=LEN       [3756]
(titelb$(h))
900 PRINT #8,"!";SPACE$(1);titela$(h);SPAC       [3620]
E$(33-la);"!";SPACE$(1);titelb$(h);SPACE$(
32-lb);"!"
910 NEXT                                         [350]
920 PRINT #8,"----------------------------       [4326]
----------------------------------------
"
930 la=LEN(namea$(x)):lb=LEN(nameb$(x))          [1335]
940 IF la MOD 2=0 THEN f=1 ELSE f=0              [954]
950 PRINT #8,"!";SPACE$(34-la/2);namea$(x)       [2655]
;SPACE$(33-la/2+f);"!"
960 IF lb MOD 2=0 THEN f=1 ELSE f=0              [738]
970 PRINT #8, "!";SPACE$(34-lb/2);nameb$(x       [2978]
);SPACE$(33-lb/2+f);"!"
980 PRINT #8,"----------------------------       [4326]
----------------------------------------
"
990 FOR n=1 TO 3                                 [801]
1000 h=n+3*(x-1):l=LEN(notizen$(h))              [2091]
1010 PRINT #8,"!";SPACE$(1);notizen$(h);SP       [2884]
ACE$(67-l);"!"
1020 NEXT                                        [350]
1030 PRINT #8,"----------------------------       [4326]
----------------------------------------
-"
1040 NEXT:GOTO 200                               [967]
1050 '                                           [117]
1060 '*****  Laden  *****                        [623]
1070 '                                           [117]
1080 CLS #5:LOCATE 1,25                          [1133]
1090 PRINT"Bitte geben Sie den Namen der z       [6189]
u ladenden Datei ein. => ";CHR$(24);
1100 INPUT "",dat$                               [478]
1110 ON ERROR GOTO 2350                          [1684]
1120 OPENIN dat$                                 [883]
1130 anz=0: 'Wenn Sie eine Datei laden wol       [6274]
len, ohne dass die Alte verloren geht, so
machen sie bitte ein "REM" davor.
1140 anz=anz+1                                   [1018]
1150 IF EOF THEN 1210                            [386]
1160 INPUT #9,namea$(anz),nameb$(anz):           [2606]
1170 FOR n=1 TO 3 :INPUT #9,notizen$(n+3*(       [3946]
anz-1)) :NEXT
1180 FOR i=1 TO 14 :INPUT #9,titela$(i+14*       [2457]
(anz-1)) :NEXT
1190 FOR i=1 TO 14 :INPUT #9,titelb$(i+14*       [1263]
(anz-1)) :NEXT
1200 GOTO 1140                                   [311]
1210 pmax=anz-1                                  [418]
1220 anz=anz-1                                   [1014]
1230 CLOSEIN                                     [752]
1240 PRINT CHR$(24);:CLS #5                      [1817]
1250 GOTO 360                                    [524]
1260 '                                           [117]
1270 '*****  Speichern  *****                    [1001]
1280 '                                           [117]
1290 CLS #5:LOCATE 1,25                          [1133]
1300 PRINT"Bitte geben Sie den Namen der z       [6818]
u speichernden Datei ein. => ";CHR$(24);
1310 INPUT "",dat$                               [478]
1320 ON ERROR GOTO 2350                          [1684]
1330 OPENOUT dat$                                [749]
1340 FOR x=1 TO pmax                             [927]
1350 WRITE #9,namea$(x),nameb$(x)                [2117]
1360 FOR n=1 TO 3 :WRITE #9,notizen$(n+3*(       [1539]
x-1)) :NEXT
1370 FOR i=1 TO 14 :WRITE #9,titela$(i+14*       [2226]
(x-1)) :NEXT
1380 FOR i=1 TO 14 :WRITE #9,titelb$(i+14*       [2383]
(x-1)) :NEXT
```

Listing "Hüllenlos"

```
1390 NEXT                                          [350]
1400 CLOSEOUT                                      [902]
1410 PRINT CHR$(24);:CLS #5                        [1817]
1420 GOTO 360                                      [524]
1430 '                                             [117]
1440 '*****  CAT  *****                            [475]
1450 '                                             [117]
1460 MODE 1:CAT                                    [1106]
1470 b$=INKEY$:IF b$="" THEN 1470 ELSE GOT         [3564]
O 200
1480 '                                             [117]
1490 '*****  Aendern  *****                        [923]
1500 '                                             [117]
1510 CLS #5:LOCATE 1,25                            [1133]
1520 PRINT"Welche Nummer Ihrer Kassette so         [6084]
ll verbessert werden? => ";CHR$(24);
1530 INPUT "",num                                  [526]
1540 PRINT CHR$(24);:CLS #5                        [1817]
1550 IF num<1 OR num>anz THEN 360                  [1848]
1560 CLS #1:CLS #2:CLS #3:CLS #4:CLS #5            [1646]
1570 LOCATE 1,25:PRINT "RETURN: Keine Aend         [6166]
erung an einer Zeile. Sonst ganze Zeile ve
rbessern."
1580 LOCATE 66,18:PRINT CHR$(24);num;CHR$(         [1718]
24)
1590 PRINT #3,namea$(num):PRINT #3,nameb$(         [1988]
num)
1600 FOR i=1 TO 14:PRINT #1,titela$(i+14*(         [1815]
num-1)):NEXT
1610 FOR i=1 TO 14:PRINT #2,titelb$(i+14*(         [4872]
num-1)):NEXT
1620 FOR n=1 TO 3:PRINT #4,notizen$(n+3*(n         [2677]
um-1)):NEXT
1630 h$=namea$(num):LOCATE #3,1,1:LINE INP         [2997]
UT #3,"",namea$(num)
1640 IF namea$(num)="" THEN namea$(num)=h$         [2705]

1650 h$=nameb$(num):LINE INPUT #3,"",nameb         [4393]
$(num)
1660 IF nameb$(num)="" THEN nameb$(num)=h$         [1421]

1670 LOCATE #1,1,1:FOR i=1 TO 14                   [1771]
1680 h$=titela$(i+14*(num-1)):LINE INPUT #         [3940]
1,"",titela$(i+14*(num-1))
1690 IF titela$(i+14*(num-1))="" THEN tite         [3309]
la$(i+14*(num-1))=h$
1700 NEXT                                          [350]
1710 LOCATE #2,1,1:FOR i=1 TO 14                   [1478]
1720 h$=titelb$(i+14*(num-1)):LINE INPUT #         [4314]
2,"",titelb$(i+14*(num-1))
1730 IF titelb$(i+14*(num-1))="" THEN tite         [2866]
lb$(i+14*(num-1))=h$
1740 NEXT                                          [350]
1750 LOCATE #4,1,1:FOR n=1 TO 3                    [1399]
1760 h$=notizen$(n+3*(num-1)):LINE INPUT #         [4982]
4,"",notizen$(n+3*(num-1))
1770 IF notizen$(n+3*(num-1))="" THEN noti         [2623]
zen$(n+3*(num-1))=h$
1780 NEXT                                          [350]
1790 GOTO 360                                      [524]
1800 '                                             [117]
1810 '*****  Blaettern  *****                       [1353]
1820 '                                             [117]
1830 x=1                                           [350]
1840 CLS #1:CLS #2:CLS #3:CLS #4:CLS #5            [1646]
1850 LOCATE 1,25:PRINT"Blaettern mit      "        [4107]
;CHR$(24);
1860 PRINT "[";CHR$(240);"]";CHR$(24);"            [2659]
und    ";CHR$(24);
1870 PRINT "[";CHR$(241);"]";CHR$(24);"            [2750]
        Exit mit    ";CHR$(24);
1880 PRINT"[RETURN]";CHR$(24);""                   [1741]
1890 CLS #1:CLS #2:CLS #3:CLS #4                   [1769]
1900 LOCATE 66,18:PRINT CHR$(24);x;CHR$(24         [1575]
)
1910 PRINT #3,namea$(x):PRINT #3,nameb$(x)         [1936]
1920 FOR i=1 TO 14:PRINT #1,titela$(i+14*(         [1467]
x-1)):NEXT
1930 FOR i=1 TO 14:PRINT #2,titelb$(i+14*(         [3470]
x-1)):NEXT
1940 FOR n=1 TO 3:PRINT #4,notizen$(n+3*(x         [2215]
-1)):NEXT
1950 a$=UPPER$(INKEY$):IF a$="" THEN 1950          [1890]
```

Listing "Hüllenlos"

```
1960 IF a$=CHR$(240) THEN x=x+1:IF x>pmax          [3469]
THEN x=pmax:GOTO 1950 ELSE GOTO 1890
1970 IF a$=CHR$(241) THEN x=x-1:IF x<1 THE         [3525]
N x=1:GOTO 1950 ELSE GOTO 1890
1980 IF a$=CHR$(13) THEN x=0:CLS #5:GOTO 3         [437]
60
1990 IF a$<>CHR$(240) OR a$<>CHR$(242) OR          [3522]
 a$<>"" THEN GOTO 1950
2000 '                                             [117]
2010 '*****  Einzeln Drucken *****                  [1178]
2020 '                                             [117]
2030 CLS #5:LOCATE 1,25                            [1133]
2040 PRINT"Welche Nummer darf ich Ihnen dr         [4670]
ucken? => ";CHR$(24);
2050 INPUT "",x:LOCATE 1,1:PRINT CHR$(24)          [1751]
2060 IF x<1 OR x>anz THEN CLS #5:GOTO 360          [1396]
2070 '                                             [117]
2080 '++++++++++++++++++++++++++++++++++           [2494]
++++++++++++++++++++++
2090 'Hier wird Ihr Drucker auf SCHMALSCHR         [4256]
IFT eingestellt.
2100 'Passen Sie, wenn noetig bitte diese          [4872]
Zeile Ihrem Drucker an.
2110 '                                             [117]
2120 PRINT #8,CHR$(27);"x";CHR$(0);CHR$(15         [2483]
):
2130 '                                             [117]
2140 '++++++++++++++++++++++++++++++++++           [2494]
++++++++++++++++++++++
2150 '                                             [117]
2160  PRINT #8,"--------------------------          [4326]
----------------------------------------
--"
2170 FOR i=1 TO 14                                 [311]
2180 h=i+14*(x-1):la=LEN(titela$(h)):lb=LE         [3756]
N(titelb$(h))
2190 PRINT #8,"!";SPACE$(1);titela$(h);SPA         [3620]
CE$(33-la);"!";SPACE$(1);titelb$(h);SPACE$
(32-lb);"!"
2200 NEXT                                          [350]
2210 PRINT #8,"--------------------------          [4326]
----------------------------------------
-"
2220 la=LEN(namea$(x)):lb=LEN(nameb$(x))           [1335]
2230 IF la MOD 2=0 THEN f=1 ELSE f=0               [954]
2240 PRINT #8,"!";SPACE$(34-la/2);namea$(x         [2655]
);SPACE$(33-la/2+f);"!"
2250 IF lb MOD 2=0 THEN f=1 ELSE f=0               [738]
2260 PRINT #8,"!";SPACE$(34-lb/2);nameb$(x         [2978]
);SPACE$(33-lb/2+f);"!"
2270 PRINT #8,"--------------------------          [4326]
----------------------------------------
-"
2280 FOR n=1 TO 3                                  [801]
2290 h=n+3*(x-1):l=LEN(notizen$(h))                [2091]
2300 PRINT #8,"!";SPACE$(1);notizen$(h);SP         [2884]
ACE$(67-l);"!"
2310 NEXT                                          [350]
2320 PRINT #8,"--------------------------          [4326]
----------------------------------------
-"
2330 CLS #5:GOTO 420                               [1193]
2340 '                                             [117]
2350 '*****  Fehlermeldungen: Diskette  **          [1251]
***
2360 '                                             [117]
2370 IF DERR=144 THEN fehler$="    Sie hab         [8251]
en einen unzulaessigen
                                   Dateina
men angegeben.          "
2380 IF DERR=147 THEN fehler$="    Ihr Inh         [3574]
altsverzeichnis ist voll.    "
2390 IF DERR=148 THEN fehler$="     Ihre D         [5151]
iskette ist schon voll.      "
2400 IF DERR=146 THEN fehler$="        Dies        [6966]
e Datei gibt es nicht
                                     auf d
ieser Diskette.          "
2410 MODE 1:LOCATE 1,13:PRINT fehler$:PRIN         [3477]
T CHR$(24)
2420 CALL &BB18                                    [389]
2430 RESUME 200                                    [603]
```

Listing "Hüllenlos"

ConText CPC

Autor: Matthias Uphoff

damit das Schreiben wieder Spaß macht

Der Klassiker

Context CPC – bis heute ungeschlagen in der Gruppe der Textverarbeitungen. Dieses Programm besticht vor allem durch seine leichte Anwendungsart, die selbst unerfahrenen Computerbesitzern den Umgang mit einer Textverarbeitung möglich macht.

Context CPC – das heißt:

- Einfachste Bedienung durch logische Tastaturbelegung; alle Funktionen sind über die CONTROL- und eine definierte Taste zu erreichen.
- Funktionen wie EINFÜGEN, FLIESSTEXT, BLOCKFORMATIERUNG und ZEILEN KOPIEREN sind über Tastendruck aufrufbar und werden in einer Statuszeile angezeigt.
- Mehrspaltiges Schreiben und Textkopieren erleichtert Ihnen das Erstellen Ihrer Korrespondenz.
- 25 KByte Textspeicher mit insgesamt 5 DIN-A4-Seiten, damit Sie auch lange Briefe problemlos erstellen können.
- Voreingestellt für die meisten CENTRONICS- Drucker, durch übersichtliche Tabellenprogrammierung anpaßbar an viele EPSON- kompatible Drucker.
- Weiterschreiben während des Druckens, denn 'Time is Money'.
- Darstellung von Sonderschriften wie VERGRÖSSERN und UNTERSTREICHEN, Anzeigen von anderen Schriftarten durch Steuerzeichen, denn Sie wollen ja sehen, was Sie drucken.
- Auf Diskette / Kassette gespeicherte Textbausteine sind überall im Text plazierbar, das erspart doppelte Schreibarbeit.
- Eingebauter Taschenrechner und Kalender, damit Sie den Überblick behalten.

Dies sind nur einige der vielen Möglichkeiten, die Context Ihnen als Textverarbeitung bietet.

Der benötigte Hardware-Aufwand ist gering.

Sie brauchen nur einen CPC 464 / 664 oder 6128 und einen Drucker. Alles andere erklärt Ihnen die ausführliche deutsche Bedienungsanleitung, welche dem Programm beiliegt.

Und wo gibt's Context CPC ?

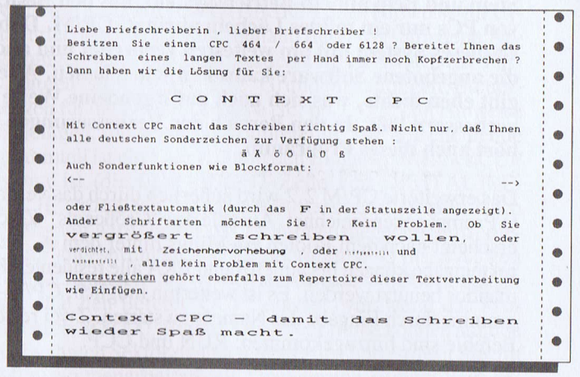
Bei DMV zu bestellen als

3"-Diskette
(Best.-Nr. 207)

Inland:		Ausland:	
Einzelpreis	59,– DM	Einzelpreis	59,– DM
zzgl. Versandkosten	3,– DM	zzgl. Versandkosten	5,– DM
Endpreis	**62,– DM**	**Endpreis**	**64,– DM**

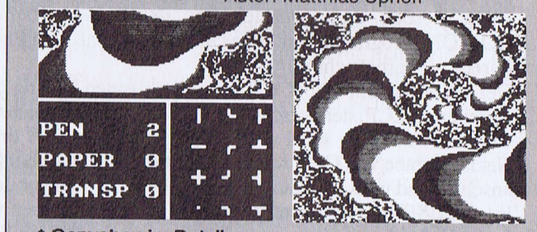
Kassette
(Best.-Nr. 206)

Inland:		Ausland:	
Einzelpreis	49,– DM	Einzelpreis	49,– DM
zzgl. Versandkosten	3,– DM	zzgl. Versandkosten	5,– DM
Endpreis	**52,– DM**	**Endpreis**	**54,– DM**



Copyshop

Das universelle Hardcopy-Programm
für CPC 464/664/6128
Autor: Matthias Uphoff



*** Copyshop im Detail:**

– Hardcopy in 4 (!) Formaten: DIN A4, DIN A5, 13,5x8,5 cm und 21,5x13,5 cm – superschnelle Hardcopy-Routine: DIN A4 in ca. 4 Minuten (DMP 4000) – arbeitet in allen 3 Modes – Anpaßmenue für JEDEN Epson-kompatiblen Drucker – läuft ebenfalls mit den Seikosha-Druckern GP-500 CPC, GP-550 CPC und GP-1000 CPC – Anpassung an Drucker möglich, die mit 1280 Punkten pro Zeile arbeiten, z.B. CPA-80 GS – Okimate ML 182 – Anpassung kann beim Verlag angefordert werden – Anpassung auch für Drucker möglich, die die Bitbild-Bytes verkehrt herum drucken (das MSB unten statt oben), z.B. NEC P2-Pinwriter. – 32 Farbraster über Menue wählbar – Grafikeditor – komfortable Pull-Down-Menues – Schnelle Fill-Routine – Beliebige Ausschnittvergrößerungen – Bildschirm invertieren – selbstrelozierende Hardcopy-Routinen für eigene Programme – neue Save- und Load-Routinen erkennen automatisch Mode und Farbwerte – Freezer – saved auf Tastendruck Screenshots aus laufenden BASIC-Programmen, die anschließend ausgedruckt werden können

* Das auf dem Datenträger mitgelieferte Programm »Screen Save«, welches beliebige Screens auf Disk abspeichert, arbeitet nur nach Entfernen des Vortex-Controllers.

Und die Weltneuheit: **Hardcopy-Simulation auf dem Bildschirm!**
Sie könne sich Ihre Hardcopy vor dem endgültigen Ausdruck auf dem Bildschirm ansehen!

COPYSHOP ist das ultimate Hardcopy-Programm für alle CPC Computer.
Erhältlich auf Kassette (Best.-Nr. 201), 3"-Diskette (Best.-Nr. 202) und 5 1/4"-Vortex (Best.-Nr. 203) inkl. ausführlicher Bedienungsanleitung.

Kassette 59,– DM (unverbindl. Preisempf.) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	59,– DM	Einzelpreis	59,– DM
zzgl. Vers.-Kosten	3,– DM	zzgl. Vers.-Kosten	5,– DM
Endpreis	**62,– DM**	**Endpreis**	**64,– DM**

Diskette 3" oder 5 1/4" 69,– DM (unverbindl. Preisempf.) Wenn Sie über den DMV-Bestellservice bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	69,– DM	Einzelpreis	69,– DM
zzgl. Vers.-Kosten	3,– DM	zzgl. Vers.-Kosten	5,– DM
Endpreis	**72,– DM**	**Endpreis**	**74,– DM**

DMV Verlag, Fuldaer Str. 6, 3440 Eschwege

DMV GmbH · Abt. SOFTWARE · POSTFACH 250 · 3440 ESCHWEGE

Bitte benutzen Sie den Coupon auf Seite 129.

Auch Gutes kann noch besser werden!

Bedienungshinweise zum erweiterten CP/M 2.2. für die CPCs 464/664/6128.

Sehr viele Computerbenutzer im Heimbereich schwören Stein und Bein auf ein Betriebssystem, das den Besitzern von PCs nur ein müdes Lächeln abringt: CP/M. Dabei ist dieses System mit am weitesten verbreitet und auch die angebotene Software kann sich sehen lassen. Aber es gibt eben nichts, was sich noch auf irgendeine Weise verbessern läßt. In den Bereich der Verbesserungen gehört auch dieses Programm.

Das erweiterte CP/M 2.2 wird äußerlich durch das veränderte Prompt gekennzeichnet. Anstelle des "Größer als"-Zeichens erscheint nach dem Booten ein Minus hinter dem A. Bei Direkteingabe können bis auf DIR und ERA alle residenten Kommandos benutzt werden. Es ist weiterhin möglich, CP/M-Programme durch Eingabe des Namens zu starten. Zwei residente Befehle sind hinzugekommen: RUN und CCP.

Zwei neue Befehle!

Wenn man mit einem CP/M- Programm gearbeitet hat, es anschließend verläßt und in die Kommandoebene zurückgekehrt ist (mit dem A-), kann man, sofern das Programm den Speicher nicht zu sehr durcheinandergewirbelt hat, durch Eingabe von RUN erneut starten, ohne daß das Programm nachgeladen werden muß (vergleichbar dem allseits beliebten GO.COM-Programm; mit dem Unterschied, daß kein Platz im Directory verbraucht wird).
Der zweite Befehl heißt CCP. Er bewirkt einen Sprung in den erweiterten CCP, der verschiedene neue Befehle beinhaltet. Die CCP- Erweiterung wird verlassen, indem man ESC drückt. Nach Eingabe von CCP wird zunächst ein CLS durchgeführt, anschließend wird ein Catalog ausgegeben, bei dem vor den Files die USER-Nr. in hexadezimaler Form ausgegeben wird. In der Fußzeile wird das angewählte Laufwerk, die USER-Nr., diesmal dezimal, die Anzahl der Files hexadezimal und die eingestellte Diskettenart (CP/M oder DATA) ausgegeben. Auf dem 1. Buchstaben des 1. Files erscheint ein Cursor. Dieser Cursor kann mit den Pfeiltasten gesteuert werden, mit der COPY-Taste werden ein oder mehrere Files zur späteren Bearbeitung angekreuzt. Verschiedene Funktionen können durch direkte Eingabe angewählt werden. Das Programm arbeitet mit Data- und CP/M-formatierten Disketten. Die Einstellung erfolgt automatisch, der jeweilige Typ wird in der Fußzeile angezeigt.

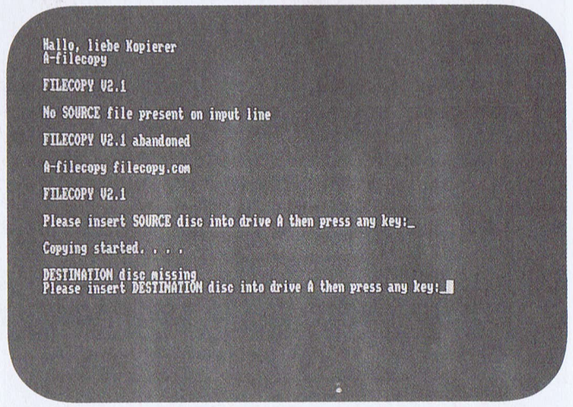


Abb. 1: Kennzeichnend für das überarbeitete CP/M: das neue PROMPT

Funktionen, die angekreuzte Files benötigen

'E' löscht alle angekreuzten Files und verabschiedet sich von Ihnen nach Bestätigung mit einer Taste. 'S' ist ein Flip-Flop. Angekreuzte, beim normalen Catalog unter AMSDOS sichtbare Files werden zu "unsichtbaren" Files gemacht. Gibt man dieser Funktion ein "unsichtbares" File vor, wird es sichtbar abgespeichert. Der eingebaute Catalog gibt auch unter AMSDOS nicht sichtbare Files aus. Sie sind an dem etwas merkwürdig aussehenden ersten Buchstaben der Extension erkennbar. 'R' funktioniert ähnlich wie S, setzt entweder Read/ Write- oder Read/ Only- Attribute. Erkennbar am zweiten Buchstaben der Extension. 'X' startet den Single-File- Copierer. Die Files werden in der angekreuzten Reihenfolge einzeln geladen und auf das vorher gewählte Ziellaufwerk überspielt. Files, die länger sind als 51 kB werden gekürzt (!). Bei zwei Laufwerken wird ohne Pause überspielt. Bei einem Laufwerk muß nach Erklingen eines Pieptons die Diskette gewechselt und eine Taste betätigt werden. R/O-Files werden als R/W- Files abgespeichert. Wenn auf der Zieldiskette schon ein File mit dem gleichen Namen wie das zu überspielende vorhanden ist, fängt die Bildschirmumrandung an zu blinken. Dies ist eine grafische Sicherheitsabfrage. Beim Drücken der Taste 'j' wird das File überschrieben, jede andere Taste unterbindet dies. Vor dem eigentlichen Start wird der Zieluser ausgegeben. Wenn die Einstellung richtig ist, genügt die Betätigung von 'ENTER', anderenfalls kann eine neue Zieluserebene eingestellt werden (wie bei der Wahl des Quellusers = 2 Zahlen). Beim Kopieren von einer Userebene zur anderen, muß der User nach dem Kopiervorgang von Hand zurückgestellt werden (U). Nach der INSERT FILENAME-Aufforderung und dem anschließendem Laden kann die Abspeicherung durch Drücken von 'ESC' übersprungen werden. 'N' wird benutzt, um ein oder mehrere Files umzubenennen. Nach dem Start wird zunächst der alte Name ausgegeben und der neue Name abgefragt. Nach Eingabe von acht Buchstaben oder Zeichen wird automatisch die Extension abgefragt. Ist der Name kürzer als acht Buchstaben, muß 'ENTER' betätigt werden. Für die Extension gilt das gleiche. Diese Funktion führt, im Gegensatz zum Original, keine Überprüfung auf ungültige Zeichen durch. Es ist also Vorsicht geboten, da nicht zulässige Zeichen verhindern, daß AMSDOS oder CP/M ein solches File laden können. 'T' ist ein Ersatz für den TYPE- Befehl. Es wird nur das erste angekreuzte File bearbeitet. Man kann sich mit 'T' auch BASIC- oder Binärprogramme ansehen, ohne daß der Bildschirm durcheinander gerät. Will man ASCII-Dateien betrachten, muß auf den alten TYPE-Befehl zurückgegriffen oder mit Taste 'O' die Optionen angewählt werden. 'T' ist nach dem Start auf Binärdateien eingestellt. Die Ausgabe kann durch Drücken einer Taste angehalten werden. Bei erneutem Tastendruck wird mit der Ausgabe fortgefahren. Drückt man nach dem ersten Tastendruck 'ESC', wird ganz abgebrochen.

Funktionen, die keine angekreuzten Files benötigen

'SPACE' führt einen Catalog auf dem aktuellen Laufwerk und der aktuellen Userebene durch und setzt das Diskettensystem zurück, so daß auch bei mehrmaligem Diskettenwechsel keine R/O- Fehlermeldungen auftreten können. 'C' führt einen extended Catalog durch. Es werden alle im Directory befindlichen Files mit der dazugehörigen USER-Nr. angezeigt (auch die gelöschten). 'W' setzt den Textbildschirm zurück. Wenn durch eine Unachtsamkeit (Atype) der Bildschirm durcheinander geraten ist (z.B. kein Cursor mehr), kann er durch Drücken von 'W' wieder beruhigt werden. 'G' wird dazu benutzt, um gelöschte Files zurückzuholen. Frage-/Antwort- Spiel wie bei Filecopy. Um die Veränderungen festzuhalten, muß die letzte Frage mit Y beantwortet werden. 'O' verzweigt in den Programmteil Options. Mehr dazu weiter unten. Mit 'D' kann das aktuelle Laufwerk eingestellt werden. Es müssen zwei Buchstaben eingegeben werden. Falls später mit der Kopierfunktion und zwei Laufwerken gearbeitet werden soll, ist die Reihenfolge: Quellaufwerk – Ziellaufwerk ansonsten ist die zweite Angabe egal. 'U' fragt den User ab. Es müssen in jedem Fall zwei Zahlen eingegeben werden, z.B. : User 1 = 01 Die Zahl muß dezimal angegeben werden.

Die Optionen

'1' schaltet die beschleunigte Bildschirmausgabe ein (in Verbindung mit dem TYPE-Befehl ein echter Genuß!). Sie bleibt auch nach der Rückkehr aus dem erweiterten CCP erhalten. '2' schaltet Option 1 wieder ab. Dies kann nötig sein, wenn auf inverse Buchstabendarstellung Wert gelegt wird (die allerdings bei der beschleunigten Wiedergabe nicht funktioniert). '3' verzweigt in den Notizblock, mehr dazu unten. '4' Vorbereitung für den neuen TYPE-Befehl 'T'. Sorgt dafür, daß bei einem Aufruf von 'T', ASCII-Dateien in ihrer ursprünglichen Form wiedergegeben werden (Mit CR, LF etc). '5' bereitet die Ausgabe von BIN-, BAS-, COM- und allen anderen Dateien mit der T-Funktion vor, wobei sämtliche Bildschirmsteuerzeichen unterdrückt werden. '6': Da der ERA-Befehl nicht mehr ansprechbar ist, kann diese Option anstelle von ERA *.* benutzt werden.
'7' Wie '6' , mit dem Unterschied, daß alle BAK-Dateien gelöscht werden. 'A' schaltet Turbo-DOS ein. Es handelt sich mit Rücksicht auf Disk-Missing-Meldungen um eine gemäßigte Beschleunigung, die aber trotzdem etwas schneller ist, als das Original. Sie wird nach Eingabe von CCP automatisch eingestellt. 'B' schaltet Turbo-DOS wieder aus.
DIE OPTIONEN WERDEN DURCH BETÄTIGUNG VON 'ESC' WIEDER VERLASSEN !!!!!
(Es sei denn, man wählt den Notizblock an.)
'4': der Notizblock (erreichbar durch Anwahl der Option 3).
Funktion 1: Eingabe eines maximal 255 Zeichen langen Textes. Eingabe wird nach dem 255. Zeichen oder nach Betätigung von 'ENTER' abgebrochen. Der Text bleibt bis zum RESET oder der neuen Eingabe einer Notiz erhalten. Funktion 2: Hiermit kann der Text auf dem Bildschirm ausgegeben werden. Funktion 3: Führt dazu, daß der Text auf den Bildschirm ausgegeben wird und nach Betätigung einer Taste auch auf den Drucker. Vor dem Drucken wird eine Steuerzeichensequenz an den Drucker ausgegeben. In der Grundeinstellung führt dies beim DMP-2000 zur Ausgabe des Textes in NLQ. Etwaige Veränderungen können Sie dem BASIC-Lader entnehmen. Der Notizblock ersetzt natürlich keine Textverarbeitung, aber um z.B. einem Bekannten die wichtigsten Anleitungspunkte für das soeben kopierte Programm mit auf den Weg zu geben, ist er mehr als geeignet. Die Anwahl von 1.Input, sofortige Betätigung von Enter und drücken von 3.Print bewirken einen LINEFEED. 'ESC' führt zu einem Rücksprung, ohne daß der Text beeinträchtigt wird. Zur Einarbeitung empfiehlt es sich, die einzelnen Punkte nacheinander durchzuspielen. Um das Programm auf neue Disketten zu übertragen, kann entweder der BASIC-Lader, der als MC-Programm abgespeicherte Lader oder das Programm DISKTOOL.COM von der Firma Vortex benutzt werden (Übertragen der CP/M-Systemspuren). Für Besitzer des CPC 6128 müssen geringfügige Änderungen durchgeführt werden, auf die im BASIC-Lader hingewiesen wird. Desweiteren befinden sich im BASIC-Lader Hinweise für experimentierfreudige Geister, anhand derer verschiedene Abwandelungen vorgenommen werden können (BORDER-Farben, Namen der neuen Befehle, Grundeinstellungen, Steuersequenzen für die Druckerausgabe etc. pp.). Und nun Hinweise für diejenigen, die selbst programmieren. Der entscheidende Programmteil ist ein Lader, der in den Eingabepuffer des CCP eingebaut ist (die COPYRIGHT-Zeile von DIGITAL RESEARCH wird an den Anfang des Puffers verschoben!). Nach jedem Booten wird, bevor das Prompt ausgegeben wird, dieser Lader angesprungen und die Sectoren #43 bis #47 von Track 1 nachgeladen. Sie werden ab Adresse #8D00 abgelegt. Sobald man den Befehl "CCP" eingibt, verzweigt das Programm nach #8D00. Eigene MC-Programme können auf den oben erwähnten Sektoren abgelegt werden und, sofern Sie mit CP/M zusammenarbeiten, als Erweiterung benutzt werden. Wer nur ein kleines Programm einbauen will, hat die Möglichkeit, es direkt hinter meiner Erweiterung abzulegen. Die Taste 'x' leitet einen Sprung an das Ende des Programms ein (im Basic-Lader steht hier ein RET). Ab dieser Adresse sind noch etwa 100 Bytes frei. Freunde und Kenner der Maschinensprache mögen mir den Aufbau der Erweiterung verzeihen. Ich programmiere nämlich mehr nach Gefühl und weniger nach Lehrbüchern, wodurch das Ganze vielleicht etwas unübersichtlich wirkt. Wer Fragen hat, kann sich aber gerne mal melden. Ich suche momentan ein kommentiertes Source-Listing für CCP, BIOS und BDOS, vielleicht weiß jemand weiter.

Empfehlungen zur Eingabe

Es ist zu empfehlen, den BASIC-Lader zunächst in der vorliegenden Form abzutippen, auch wenn Sie einen CPC 6128 besitzen. Dabei ist peinlich genau auf die richtige Eingabe der Leerzeichen in den REM-Zeilen zu achten. In den meisten dieser Zeilen wird auf verschiedene Bytes hingewiesen, die verändert werden müssen, um das Programm für den CPC 6128 anzupassen. Die zu verändernden Bytes befinden sich immer in der DATA-Zeile über der REM-Zeile und sind durch nach

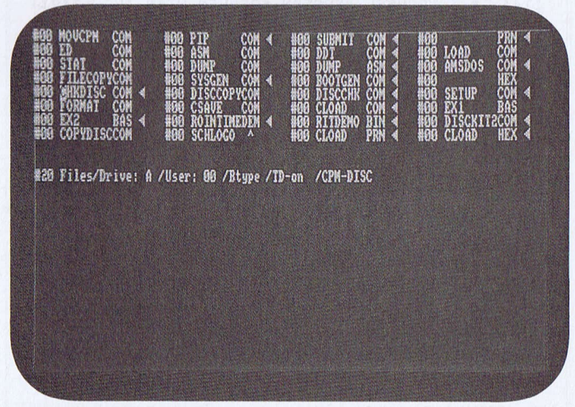


Abb. 2: Das Menü-File unter CCP.

oben zeigende Pfeile markiert (im Abdruck des Laders erscheint das 'kleine Dach'. Zu finden ist dieser Pfeil unter dem Pfundzeichen.) Wenn also eine falsche Anzahl von Leerzeichen in der REM-Zeile eingegeben wird, zeigen die Pfeile auf das falsche Byte und Sie können anpassen, soviel Sie wollen, das Programm wird nicht laufen! Die Pfeile in den REM-Zeilen erscheinen im vorliegenden Lader etwas fehlplaziert, da sie nicht, wie oben erwähnt, auf bestimmte Bytes zeigen. Dies liegt an der Beschränkung des Ausdrucks auf 35 Zeichen/Zeile. Tippen Sie das Listing ab und schauen Sie es sich bei 80 Zeichen/Zeile an. Dann hat alles wieder seine Richtigkeit. Übrigens, bei Veränderungen in den DATA-Zeilen sollte auch die Checksumme angepaßt werden, weil sonst Fehler angezeigt werden, die eigentlich nicht vorhanden sind.

(M. Grams/jb)

für 464-664-6128

```
1000 '  BASIC-LADER ZUM GENERIEREN EINES P [3847]
ROGRAMMS,DAS DEN CP/M 2.2
1010 '  CONSOLE-COMAND-PROCESSOR (CCP) PAT [2548]
CHT.
1020 '  DAS PROGRAMM IST IN DER MOMENTANEN [3525]
 VERSION AUF DEN CPC 464 EINGESTELLT.
1030 '  ANPASSUNGEN AN DEN CPC 6128 STEHEN [3291]
 IM DATA-TEIL.
1040 '  BEI AENDERUNGEN IST UNBEDINGT AUF  [6911]
DIE ANPASSUNG DER CHECKSUMME ZU
1050 '  ACHTEN.                            [462]
1051 '                                     [117]
1060 '                                     [3602]

1070 '                                     [117]
1071 '************************************ [2234]
**********************
1072 '*                                    [175]
                *
1080 '*  KAEPT'N KOMAS INTERGALACTIC ENTER [7524]
PRISES//BETEIGEUZE    *
1081 '*                                    [175]
                *
1090 '*           in irdischer Zusammenarbe [1213]
it mit          *
1100 '*                                    [175]
                *
1110 '*        MICHAEL GRAMS//E'FOERDE// 1 [2599]
1.11.86         *
1111 '*                                    [175]
                *
1112 '************************************ [2234]
**********************
1113 '                                     [117]
1120 DATA 3E,84,32,A5,8C,CD,58,8C,21,18,97 [2862]
,11,08,97,01,24,00,ED,B0,3E,  1878
1130 DATA 2D,32,91,9A,21,C3,37,22,85,9A,3E [2894]
,97,32,87,9A,21,00,8D,22,C1,  1951
1131 '*   ^^   PROMPT-ZEICHEN A-           [1244]
1132 '                                     [117]
1140 DATA 9A,21,00,01,22,C3,9A,21,AB,8C,11 [3965]
,10,9A,01,07,00,ED,B0,21,B2,  1734
1141 '*     ^^ ^^ AN DIESE ADRESSE WIR     [2764]
D NACH EINGABE VON "RUN"
1142 '    VERZWEIGT (#0100).FREUNDE VON TU [3806]
RBO-PASCAL,DIE SICH UEBER
1143 '*   STAENDIGE FEHLERMELDUNGEN BEI UN [3556]
ANGEMELDETEM DISKETTEN-
1144 '*   WECHSEL AERGERN,KOENNEN HIER AUC [3491]
H #38,#22 (FUER VERS.3.0)
1145 '*   EINSETZEN.DADURCH WIRD NACH EINE [2744]
M "ABHEBEN" VON TURBO.COM
1146 '*   EIN WARMSTART DURCHGEFUEHRT,DER  [3325]
DEN IM SPEICHER BEFINDLICHEN
1147 '*   QUELLTEXT ERHAELT.(WERTE FUER TU [3101]
RBO-PASCAL 2.0 #35,#21).
1148 '*   QUELLE c't 1.86!                 [813]
1149 '                                     [117]
1150 DATA 8C,11,37,97,01,4D,00,ED,B0,3E,85 [3713]
,32,A5,8C,CD,58,8C,21,52,8C,  2092
1160 DATA CD,D4,BC,22,53,8C,79,32,55,8C,21 [2527]
,00,8D,22,56,8C,1E,00,16,00,  1744
1170 DATA 0E,43,CD,3D,8C,1E,00,16,00,0E,44 [3446]
,CD,3D,8C,1E,00,16,00,0E,45,  1162
1180 DATA CD,3D,8C,1E,00,16,00,0E,46,CD,3D [3434]
,8C,1E,00,16,00,0E,47,CD,3D,  1351
1190 DATA 8C,C9,2A,56,8C,DF,53,8C,CD,47,8C [3162]
,C9,2A,56,8C,11,00,02,19,22,  2018
1200 DATA 56,8C,C9,85,00,00,00,00,00,21,A5 [3867]
```

Listing "CP/M-Verbesserungen"

```
,8C,CD,D4,BC,22,A6,8C,79,32,  2014
1210 DATA A8,8C,21,00,97,22,A9,8C,1E,00,16 [4655]
,00,0E,48,CD,90,8C,1E,00,16,  1514
1220 DATA 00,0E,49,CD,90,8C,1E,00,16,01,0E [4082]
,41,CD,90,8C,1E,00,16,01,0E,  1264
1230 DATA 42,CD,90,8C,C9,2A,A9,8C,DF,A6,8C [3070]
,CD,9A,8C,C9,2A,A9,8C,11,00,  2704
1240 DATA 02,19,22,A9,8C,C9,85,00,00,00,00 [4363]
,00,43,43,50,20,52,55,4E,3E,  1257
1250 '*   HEX-WERTE DER BEFEHLSNAMEN       [1768]
    ^C ^C ^P    ^R ^U ^N
1260 '*   BEI UMBAU IST ZU BEACHTEN,DASS M [5059]
AX. 4 BUCHSTABEN EINGEGEBEN WERDEN
1270 '*   KOENNEN (BEI "RUN" NUR 3!),UND D [2939]
ASS GROSSBUCHSTABEN VERWENDUNG
1280 '*   FINDEN.                          [305]
1290 '                                     [117]
1300 DATA 08,32,84,97,21,00,8D,22,82,97,01 [3179]
,00,00,CD,1E,AD,18,0E,CD,21,  1515
1310 DATA AD,CD,68,97,CD,D8,98,CD,27,AD,18 [3010]
,F0,3A,84,97,4F,FE,1C,28,14,  2649
1320 DATA 3C,32,84,97,06,00,18,E2,ED,5B,82 [4151]
,97,21,80,00,19,22,82,97,C9,  1960
1330 DATA 21,CD,98,22,85,9A,3E,97,32,87,9A [3706]
,C3,85,9A,00,00,00,31,AB,9E,  2123
1340 ' *                  BEGINN DER ERW [2616]
EITERUNG          ^
1350 '                                     [117]
1360 DATA CD,E3,95,CD,2A,94,0E,0D,CD,05,00 [4248]
,3A,17,90,CD,14,90,0E,1F,CD,  2057
1370 '*   ^1 ^2 ^3 ^4 ^5 ^6 :1-3 AUFRUF TU [5704]
RBO-DOS 4-6:AUFRUF FAST-SCREEN
1380 '*   DIE ROUTINEN WERDEN MIT DEM AUFR [3572]
UF VON "CCP" INITIALISIERT.
1390 '*   WENN DIES NICHT STATTFINDEN SOLL [5523]
,ANSTELLE DER WERTE NULLEN EIN-
1400 '*   SETZEN UND DIE CHECKSUMME AENDER [3619]
N!
1410 '                                     [117]
1420 DATA 05,00,11,0D,00,19,7E,32,40,8D,01 [3604]
,06,06,CD,9B,BE,38,BC,CD,35,  1506
1430 '*                  ^^ ^^ DIE ZU BEGI [2031]
NN EINGESTELLTEN BORDER-FARBEN.
1440 '*   MUESSEN BEI AENDERUNG WEITER UNT [5212]
EN NOCHMAL ANGEPASST WERDEN.
1450 '                                     [117]
1460 DATA 8D,CD,70,8D,CD,12,8E,C3,09,8D,AF [4084]
,32,3D,72,21,1F,82,22,1C,94,  2113
1470 DATA 01,02,00,CD,1E,AD,18,0E,CD,21,AD [3910]
,CD,64,8D,CD,A5,8F,CD,27,AD,  2236
1480 DATA 18,F0,3A,3D,72,4F,FE,10,C8,3C,32 [2942]
,3D,72,06,00,18,E3,ED,5B,1C,  1944
1490 DATA 94,21,80,00,19,22,1C,94,C9,21,01 [3200]
,05,22,21,8F,21,3F,8A,22,1F,  1293
1500 DATA 8F,2A,1C,94,06,1E,3E,E5,77,23,10 [3638]
,FC,CD,97,8F,AF,32,1E,8F,32,  2057
1510 DATA 23,8F,21,20,82,7E,FE,E5,C8,E5,2B [2689]
,7E,FE,00,C2,C5,8D,E5,11,0C,  2624
1520 DATA 00,19,7E,E1,B7,20,1B,3E,23,CD,F8 [3664]
,8D,CD,E0,8D,3E,20,CD,F8,8D,  2567
1530 DATA 23,CD,CC,8D,CD,D6,8D,3A,1E,8F,3C [2433]
,32,1E,8F,E1,11,20,00,19,18,  1982
1540 DATA C8,06,0B,7E,CD,F8,8D,23,10,F9,C9 [3858]
,06,05,3E,09,CD,F8,8D,10,FB,  2381
1550 DATA C9,7E,F5,07,07,07,07,CD,EA,8D,F1 [3724]
,E6,0F,F6,30,FE,3A,38,02,C6,  2528
1560 DATA 07,CD,F8,8D,C9,E5,D5,C5,F5,5F,0E [4055]
,06,CD,05,00,F1,C1,D1,E1,C9,  3080
1570 DATA 1E,FF,0E,06,CD,05,00,B7,28,F6,C9 [3440]
,11,24,8F,CD,A0,8F,3E,23,CD,  2191
1580 DATA F8,8D,21,1E,8F,CD,E0,8D,11,28,8F [2794]
,CD,A0,8F,3A,40,8D,FE,02,28,  2432
1590 DATA 08,11,51,8F,CD,A0,8F,18,06,11,5D [3409]
```

Listing "CP/M-Verbesserungen"


```
,8F,CD,A0,8F,2A,21,8F,CD,9B,  2126
1600 DATA BE,75,BB,CD,07,8E,FE,20,C8,FE,79 [4084]
,CA,93,96,FE,77,CA,C2,95,FE,  3380
1610 '*  TASTATURABFRAGE:SPACE=^^          ^^ [4296]
=y(frei)      ^^=w
1620 '                                     [117]
1630 DATA 6F,CA,54,94,FE,75,CA,A8,8F,FE,64 [3837]
,CA,ED,8F,FE,63,CA,1F,90,FE,  3349
1640 '*   ^^=o           ^^=u           ^^ [822]
=d            ^^=c
1650 '                                     [117]
1660 DATA 67,CA,3C,90,FE,65,CA,C7,90,FE,03 [3018]
,CA,66,96,FE,F0,28,3C,FE,F2,  3210
1670 '*   ^^=g           ^^=e           ^^ [2517]
=?? Curs.--- ^^=up//left ^^
1680 '                                     [117]
1690 DATA 28,4C,FE,F3,28,56,FE,F1,28,3A,FE [3234]
,E0,28,5C,FE,72,CA,15,91,FE,  2932
1700 '*      right=^^          ^^=down     [1303]
 ^^=COPY     ^^=r
1710 '                                     [117]
1720 DATA 73,CA,6C,91,FE,6E,CA,BA,91,FE,74 [2751]
,CA,63,95,FE,78,CA,6C,92,FE,  3371
1730 '*   ^^=s          ^^=n           ^^ [896]
=t            ^^=x
1740 '                                     [117]
1750 DATA FC,C2,B1,8E,CD,97,8F,C3,82,9A,3E [3566]
,07,CD,F8,8D,C3,46,8E,2A,21,  2888
1760 '*   ^^=ESC                           [743]
1770 '                                     [117]
1780 '*  MIT HILFE DER OBIGEN ANGABEN,KANN [4461]
 DIE TASTATURABFRAGE AUF BELIEBIGE
1790 '*  BUCHSTABENABFRAGEN UMGEBAUT WERDE [3091]
N.
1800 '*  (Setzen Sie doch mal anstelle der [3413]
 Cursorabfrage eine Joystickab-
1810 '*  abfrage ein!).                    [841]
1820 '                                     [117]
1830 DATA 8F,2B,22,21,8F,C3,3E,8E,2A,21,8F [3814]
,23,22,21,8F,C3,3E,8E,2A,21,  1732
1840 DATA 8F,06,14,25,10,FD,22,21,8F,C3,3E [2962]
,8E,2A,21,8F,06,14,24,10,FD,  1633
1850 DATA 22,21,8F,C3,3E,8E,06,0B,2A,21,8F [4253]
,E5,CD,9B,BE,75,BB,CD,9B,BE,  2477
1860 DATA 60,BB,2A,1F,8F,77,23,22,1F,8F,E1 [3947]
,24,10,E9,3E,09,CD,F8,8D,3E,  2098
1870 DATA 09,CD,F8,8D,3A,23,8F,3C,32,23,8F [2241]
,3E,F7,CD,F8,8D,C3,3E,8E,00,  2429
1880 '* ASCII-WERT ANKREUZPFEIL            [1965]
   ^^
1890 '                                     [117]
1900 DATA 00,00,00,00,00,0A,0A,0D,24,20,46 [3043]
,69,6C,65,73,2F,44,72,69,76,  1052
1910 DATA 65,3A,20,41,20,2F,55,73,65,72,3A [3724]
,20,30,30,20,2F,42,74,79,70,  1430
1920 DATA 65,20,2F,54,44,2D,6F,6E,20,24,20 [3312]
,2F,44,41,54,41,2D,44,49,53,  1296
1930 DATA 43,24,20,2F,43,50,4D,2D,44,49,53 [2895]
,43,20,24,42,74,79,70,65,41,  1391
1940 DATA 74,79,70,65,54,44,2D,6F,6E,20,54 [3204]
,44,2D,6F,66,66,2A,1F,8F,11,  1645
1950 DATA 5D,00,01,0B,00,ED,B0,C9,11,5C,00 [2956]
,C3,CB,97,11,5C,00,C3,DA,97,  2050
1960 DATA 11,9D,8F,C3,A0,8F,04,02,24,0E,09 [3696]
,C3,05,00,C3,D8,98,CD,97,8F,  2142
1970 '*           MODE 2        ^^         [527]
1980 '                                     [117]
1990 DATA 11,E5,8F,CD,A0,8F,CD,07,8E,32,3F [2940]
,8F,CD,F8,8D,D6,30,6F,AF,06,  2655
2000 DATA 0A,85,10,FD,F5,CD,07,8E,32,40,8F [3751]
,CD,F8,8D,D6,30,E1,84,FE,10,  2751
2010 DATA 30,D3,32,E2,8F,32,9C,8D,C3,DE,8F [3858]
,5F,C3,15,98,00,C2,00,55,53,  2410
2020 DATA 52,2E,4E,72,3A,24,CD,97,8F,11,18 [2714]
,90,CD,A0,8F,CD,07,8E,D6,20,  2206
2030 DATA 32,36,8F,CD,F8,8D,D6,41,32,17,90 [4091]
,CD,07,8E,D6,20,CD,F8,8D,D6,  2745
2040 DATA 41,32,1E,94,C9,C3,BD,97,00,44,52 [3089]
,49,56,45,3A,24,3E,CA,32,9D,  1972
2050 DATA 8D,21,00,00,22,9B,8D,CD,70,8D,3E [2157]
,FE,32,9B,8D,2A,E2,8F,22,9C,  2225
2060 DATA 8D,CD,07,8E,C9,CD,97,8F,21,20,82 [3806]
,7E,FE,E5,28,3B,E5,2B,7E,FE,  2750
2070 DATA E5,C2,7B,90,E5,11,B4,90,CD,A0,8F [3470]
,E1,23,CD,CC,8D,11,A7,90,CD,  3111
2080 DATA A0,8F,CD,07,8E,FE,6A,20,0A,3E,0A [3846]
,CD,F8,8D,E1,E5,2B,AF,77,06,  2522
```

Listing "CP/M-Verbesserungen"

```
2090 DATA 1B,3E,08,CD,F8,8D,10,FB,E1,11,20 [4015]
,00,19,18,C0,11,B9,90,CD,A0,  2184
2100 DATA 8F,CD,07,8E,FE,79,28,06,FE,78,28 [1518]
,13,18,ED,2A,51,8D,E5,23,23,  2175
2110 DATA 23,22,51,8D,CD,35,8D,E1,22,51,8D [3184]
,C9,20,55,4E,45,52,41,3F,3C,  1906
2120 DATA 4A,2F,20,3E,24,23,45,35,20,24,0A [4177]
,0A,0D,53,41,56,45,3F,3C,59,  1024
2130 DATA 2F,58,3E,24,CD,F3,93,3A,23,8F,B7 [3431]
,28,27,3D,32,23,8F,2A,1F,8F,  1831
2140 DATA CD,CC,8D,CD,7F,8F,CD,E8,93,CD,FA [2935]
,90,FE,FF,28,08,11,0B,91,CD,  3143
2150 DATA A0,8F,18,DB,11,00,91,CD,A0,8F,18 [3640]
,D3,CD,07,8E,11,5C,00,C3,EF,  2348
2160 DATA 97,20,2A,4B,4F,54,5A,2A,0A,0A,0D [3624]
,24,20,20,62,79,65,21,0A,0A,  1101
2170 '*     ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ ^^ [4345]
 =10 BUCHSTABEN FEHLERMELDUNG
2180 '                                     [117]
2190 DATA 0D,24,CD,F3,93,3A,23,8F,B7,20,01 [3699]
,C9,3D,32,23,8F,CD,7F,8F,2A,  2103
2200 DATA 1F,8F,CD,CC,8D,3A,65,00,CB,7F,20 [3647]
,04,CB,FF,18,02,CB,BF,32,65,  2278
2210 DATA 00,CD,E8,93,CD,57,91,FE,FF,20,09 [4083]
,11,00,91,CD,A0,8F,C3,55,91,  2666
2220 DATA 11,5F,91,CD,A0,8F,18,C1,11,5C,00 [3171]
,0E,1E,C3,05,00,20,52,2F,57,  1583
2230 DATA 2F,52,2F,4F,21,0A,0A,0D,24,CD,F3 [3492]
,93,3A,23,8F,B7,20,01,C9,3D,  1666
2240 DATA 32,23,8F,CD,7F,8F,2A,1F,8F,CD,CC [3245]
,8D,3A,66,00,CB,7F,20,04,CB,  2198
2250 DATA FF,18,02,CB,BF,32,66,00,CD,E8,93 [3590]
,CD,57,91,FE,FF,20,08,11,00,  2414
2260 DATA 91,CD,A0,8F,18,06,11,AD,91,CD,A0 [3535]
,8F,18,C2,20,53,59,53,2F,44,  2146
2270 DATA 49,52,21,0A,0A,0D,24,CD,F3,93,06 [3510]
,0D,21,5E,92,3E,20,77,23,10,  1408
2280 DATA FC,3E,03,32,66,92,3A,23,8F,B7,20 [3969]
,01,C9,3D,32,23,8F,CD,7F,8F,  2032
2290 DATA 11,45,92,CD,A0,8F,2A,1F,8F,CD,CC [3292]
,8D,11,4E,92,CD,A0,8F,0E,0A,  2279
2300 DATA 11,5C,92,CD,05,00,06,20,3E,09,CD [2831]
,F8,8D,10,FB,11,55,92,CD,A0,  2048
2310 DATA 8F,0E,0A,11,66,92,CD,05,00,06,08 [3099]
,21,5E,92,11,6D,00,CD,30,92,  1454
2320 DATA 06,03,21,68,92,CD,30,92,11,5C,00 [3550]
,0E,17,CD,05,00,FE,FF,28,12,  1614
2330 DATA CD,E8,93,18,8D,7E,FE,61,38,02,D6 [4009]
,20,12,23,13,10,F4,C9,11,00,  2080
2340 DATA 91,CD,A0,8F,18,E6,0A,0A,0D,4F,4C [2961]
,44,3D,20,24,3A,4E,45,57,3D,  1645
2350 DATA 20,24,20,45,58,54,3D,2E,24,08,00 [4085]
,00,00,00,00,00,00,00,00,00,  492
2360 DATA 00,00,00,00,00,CD,F3,93,11,58,96 [4010]
,CD,A0,8F,CD,07,8E,FE,0D,28,  2019
2370 DATA 24,32,61,96,CD,F8,8D,D6,30,6F,AF [4293]
,06,0A,85,10,FD,F5,CD,07,8E,  2492
2380 DATA 32,62,96,CD,F8,8D,D6,30,E1,84,FE [3150]
,10,30,D2,32,E4,8F,3A,17,90,  2685
2390 DATA 5F,3A,1E,94,BB,28,07,3E,C9,32,4E [4188]
,94,18,05,3E,CD,32,4E,94,01,  1677
2400 DATA 06,06,CD,9B,BE,38,BC,21,00,01,22 [3194]
,1C,94,3A,23,8F,B7,C8,3D,32,  1780
2410 '*   ^^ ^^=   2.ANPASSUNG DER BORDER- [1915]
FARBEN
2420 '                                     [117]
2430 DATA 23,8F,CD,7F,8F,11,21,94,CD,A0,8F [3615]
,21,5D,00,CD,CC,8D,CD,D8,93,  2603
2440 DATA CD,4E,94,0E,0D,CD,05,00,3A,17,90 [3789]
,CD,14,90,3A,E2,8F,CD,DE,8F,  2259
2450 DATA 21,3F,8A,22,A7,93,0E,0F,11,5C,00 [3742]
,CD,05,00,FE,FF,CA,FD,93,CD,  2246
2460 DATA 60,93,11,1F,94,CD,A0,8F,CD,4E,94 [3242]
,FE,FC,C8,CD,D8,93,0E,0D,CD,  2884
2470 DATA 05,00,3A,1E,94,CD,14,90,3A,E4,8F [4314]
,CD,DE,8F,3A,65,00,CB,BF,32,  2212
2480 DATA 65,00,21,3F,8A,22,A7,93,0E,11,11 [2777]
,5C,00,CD,05,00,FE,FF,C2,04,  1740
2490 DATA 94,0E,16,11,5C,00,CD,05,00,FE,FF [3289]
,CA,FD,93,21,00,01,22,1C,94,  1858
2500 DATA CD,7C,93,CD,E8,93,C3,B6,92,CD,AC [2383]
,93,0E,14,11,5C,00,CD,05,00,  2460
2510 DATA 2A,1C,94,77,B7,C2,E0,93,CD,9C,93 [3386]
,D4,B6,93,C3,60,93,CD,AC,93,  3096
2520 DATA 2A,1C,94,7E,B7,C2,E0,93,0E,15,11 [4454]
,5C,00,CD,05,00,B7,C2,FD,93,  2223
2530 DATA CD,9C,93,D4,C3,93,C3,7C,93,2A,1C [3146]
```

Listing "CP/M-Verbesserungen"

```
,94,11,81,00,19,22,1C,94,11, 2144
2540 DATA 3F,8A,ED,52,C9,2A,1C,94,23,EB,0E [3934]
,1A,C3,05,00,21,FF,FF,22,A7, 2193
2550 DATA 93,21,00,C0,22,1C,94,C9,21,00,C0 [3826]
,11,00,01,01,00,40,CD,9B,BE, 1641
2560 DATA 1B,B9,21,00,01,22,1C,94,C9,AF,32 [3166]
,68,00,32,7C,00,C9,0E,10,11, 1408
2570 DATA 5C,00,C3,05,00,2A,1F,8F,11,0B,00 [2862]
,19,22,1F,8F,C9,CD,97,8F,21, 1502
2580 DATA 3F,8A,22,1F,8F,C9,11,00,91,CD,A0 [3777]
,8F,C9,01,1C,00,CD,9B,BE,38, 2116
2590 '*                                    [785]
        ^^ ^^ FUER COPY-
2600 '*       PRG:WENN FILE AUF ZIELDISK SC [2745]
HON VORHANDEN,BLINKT DIE BORDER
2610 '*       IN DIESEN FARBEN.            [681]
2620 '                                     [117]
2630 DATA BC,CD,9B,BE,06,BB,FE,6A,C0,CD,FA [2992]
,90,C3,44,93,00,00,00,00,00, 2492
2640 DATA 07,24,0D,49,4E,53,45,52,54,3A,24 [3520]
,2A,D4,BD,22,4C,94,21,37,94, 1556
2650 DATA 22,D4,BD,C9,E5,CD,D3,12,EB,E1,CD [4151]
,64,0B,0E,08,41,1A,77,7C,81, 2560
2660 '*                          ^^ ^^     [2838]
 ^^ ^^    ANPASSUNG VON FASTSCREEN
2670 '*   AN CPC 6128.=      D4 12          [1510]
 6A 0B
2680 '*                    TXT GET MATRX    [1348]
SCR CHR POS
2690 '*   DA ICH KEINEN 6128 BEZITZE KONNT [3386]
E ICH DIE WERTE NICHT
2700 '*   UEBERPRUEFEN.                    [1425]
2710 '                                     [117]
2720 DATA 67,1C,10,F8,C9,00,00,CD,9B,BE,06 [3025]
,BB,C9,11,96,94,CD,A0,8F,CD, 2568
2730 DATA 07,8E,FE,31,CC,2A,94,FE,32,CC,8F [3042]
,94,FE,33,CA,EC,94,FE,34,CC, 3046
2740 DATA C8,95,FE,35,CC,D2,95,FE,36,CA,1C [2821]
,96,FE,37,CA,24,96,FE,61,CC, 3159
2750 DATA E3,95,FE,62,CC,EE,95,FE,FC,C8,18 [4284]
,C5,2A,4C,94,22,D4,BD,C9,04, 3152
2760 DATA 02,31,3A,53,70,2D,6F,6E,20,32,3A [4045]
,53,70,2D,6F,66,66,20,33,3A, 1406
2770 DATA 4E,4F,54,45,20,34,3A,41,74,79,70 [2892]
,65,20,35,3A,42,74,79,70,65, 1626
2780 DATA 20,36,3A,45,72,61,20,41,6C,6C,20 [3839]
,37,3A,45,72,61,20,2A,2E,42, 1348
2790 DATA 41,4B,0A,0A,0D,41,3A,54,44,2D,6F [3448]
,6E,20,42,3A,54,44,2D,6F,66, 1280
2800 DATA 66,0A,0A,0D,24,11,00,96,CD,A0,8F [2531]
,CD,07,8E,FE,31,CC,09,95,FE, 2119
2810 DATA 32,CC,16,95,FE,33,CC,2C,95,FE,FC [3809]
,C8,18,E3,3E,FF,32,1A,B6,0E, 2673
2820 DATA 0A,11,1A,B6,C3,05,00,3A,1B,B6,16 [2711]
,00,5F,21,1C,B6,E5,19,3E,24, 1414
2830 '*          ^^ ^^          ^^ ^^     [1133]
        ^^ ^^
2840 '* 6128=    76 B2          77 B2     [855]
        78 B2
2850 DATA 77,E1,EB,CD,A0,8F,C3,07,8E,CD,16 [3744]
,95,21,8F,96,7E,FE,24,28,0B, 2600
2860 DATA E5,5F,0E,05,CD,05,00,E1,23,18,F0 [3448]
,21,1C,B6,7E,FE,24,28,0B,E5, 2016
2870 '*                                    [516]
   ^^ ^^
2880 '*                                    [184]
    78 B2
2890 '*   DIE OBIGEN AENDERUNGEN MUESSEN D [3526]
URCHGEFUEHRT WERDEN,
2900 '*   DAMIT DIE NOTIZBLOCKFUNKTION AUF [3332]
 DEM 6128 LAEUFT.
2910 '*   AUCH HIER FEHLT MIR DIE MOEGLICH [3532]
KEIT DIE FUNTIONSFAEHIGKEIT
2920 '*   ZU UEBERPRUEFEN.BEIM 464 BEFINDE [3067]
T SICH DER NOTIZSPEICHER
2930 '*   IM SOUND-BUFFER AB ADRESSE #B61A [3455]
.M.M.N. ENTSPRICHT ADRESSE
2940 '*   #B276 DEM SOUND-BUFFER BEIM 6128 [3083]
.
2950 '                                     [117]
2960 DATA 5F,0E,05,CD,05,00,E1,23,18,F0,1E [4085]
,0D,0E,05,CD,05,00,1E,0A,0E, 1174
2970 DATA 05,C3,05,00,CD,F3,93,3A,23,8F,B7 [3506]
,C8,CD,D8,93,CD,7F,8F,CD,8B, 2806
2980 DATA 8F,11,00,50,CD,A5,8F,0E,14,11,5C [3640]
,00,CD,05,00,B7,28,04,CD,07, 1545
2990 DATA 8E,C9,3E,80,32,C1,95,11,00,50,3A [3463]
```

Listing "CP/M-Verbesserungen"

```
,C1,95,B7,28,E3,1A,FE,20,38, 2240
3000 DATA 07,FE,7F,30,03,CD,F8,8D,21,C1,95 [4243]
,35,13,D5,CD,AF,95,D1,18,E2, 2681
3010 '*          ^^ FUER TYPE BEFEHL.#7F ST [2234]
ELLT DEN OBERSTEN ASCII-WERT
3020 '*   DAR.ZEICHEN, DIE GROESSER SIND,W [2682]
ERDEN NICHT AUSGEGEBEN.
3030 '*   WER AUF DIESE ZEICHEN WERT LEGT, [4264]
SETZT ANSTELLE VON #7F DEN WERT
3040 '*   #FE EIN.                         [1522]
3050 '                                     [117]
3060 DATA 0E,06,1E,FF,CD,05,00,B7,C8,CD,07 [4137]
,8E,FE,FC,C0,D1,D1,C9,00,CD, 2774
3070 DATA 9B,BE,4E,BB,C9,AF,32,99,95,21,6E [3538]
,8F,18,09,C9,3E,20,32,99,95, 2304
3080 DATA 21,69,8F,11,43,8F,01,05,00,ED,B0 [3627]
,C9,21,46,96,CD,0D,C6,21,73, 1945
3090 DATA 8F,18,09,21,4F,96,CD,0D,C6,21,79 [4654]
,8F,11,4A,8F,01,06,00,ED,B0, 1805
3100 DATA C9,0A,0D,31,3A,49,4E,50,20,20,20 [4368]
,32,3A,56,49,45,57,20,20,20, 1177
3110 DATA 33,3A,50,52,54,0A,0A,0D,24,21,30 [3130]
,96,22,1F,8F,18,06,21,3B,96, 1135
3120 DATA 22,1F,8F,CD,7F,8F,C3,FA,90,3F,3F [4049]
,3F,3F,3F,3F,3F,3F,3F,3F,3F, 1965
3130 DATA 3F,3F,3F,3F,3F,3F,3F,3F,42,41,4B [2427]
,23,00,C8,00,01,01,0A,00,03, 960
3140 '*                                    [988]
   ^ANFANG TD-TAB.    ENDE^
3150 '                                     [117]
3160 DATA 32,00,FA,00,AF,0F,0C,01,03,5A,69 [2204]
,65,6C,2D,55,73,72,3A,30,30, 1423
3170 '*   ^ ANFANG SLOW-DOS-TAB.  ^ENDE    [2089]
3180 '                                     [117]
3190 DATA 08,08,24,21,75,96,7E,C6,01,FE,E0 [3814]
,C8,CD,F8,8D,23,18,F4,03,00, 2255
3200 DATA 2E,A3,1F,61,78,1F,4C,2D,46,71,60 [5216]
,6C,72,2E,30,30,2D,30,30,2D, 1438
3210 DATA 37,35,2E,DF,1B,78,01,24,C9, 762  [1596]
3220 '*                    ^^ ^^ ^^ ^^  STEUERS [3049]
EQUENZ FUER DRUCKERAUSGABE
3230 '*   KANN DURCH BELIEBIGE SEQUENZ ERS [4032]
ETZT WERDEN.ALLE ZEICHEN,
3240 '*   DIE AB #1B STEHEN,WERDEN AUSGEGE [6150]
BEN,BIS DAS PRG. AUF DEN WERT
3250 '*   #24 TRIFFT.                      [611]
3260 '*   MOMENTANE EINSTELLUNG IST NLQ-EI [5450]
N(#1B,#78,#01 UND ALS ABSCHLUSS #24.
3270 '                                     [117]
3280 MODE 2:PRINT"!     41 Sekunden Pause. [3328]
MCode wird gepoked       !"
3290 dat=0:sz=0:dz= 1120                   [548]
3300 FOR adr=-29777  TO-26989              [573]
3310 READ byte$:dat=dat+1                  [1348]
3320 sz=sz+VAL("&"+byte$)                  [1203]
3330 POKE adr,VAL("&"+byte$)               [745]
3340 IF dat < 20 AND adr < -26989 THEN 338 [1873]
0
3350 READ chksum                           [1222]
3360 IF chksum<>sz THEN PRINT "Fehler in Z [3503]
eile :";dz:z=z+schr
3370 dz=dz+ 10 :sz=0:dat=0                 [843]
3380 NEXT adr                              [547]
3390 PRINT:PRINT"!Legen Sie eine ungepatch [5080]
te CP/M 2.2-Diskette ein !"
3400 '                                     [117]
3410 '*Es muss unbedingt eine ungepatchte  [13580]
Diskette verwendet werden,weil es
  *sonst Probleme mit dem Copyright-Vermer
k von DIGITAL RESEARCH gibt(Es          *g
eht natuerlich auch mit Verzicht auf die E
intragung,aber das ist ver-         *boten
!).
3420 '                                     [117]
3430 PRINT:PRINT"!              Und druecke [3485]
n Sie eine Taste        !"
3440 CALL &BB06                            [393]
3450 PRINT:PRINT"!       CP/M-Patch wird   [3643]
durchgefuehrt            !"
3460 CALL &8BAF'START DER PATCH-ROUTINE    [2659]
3470 |CPM' *oder SAVE"MCCODE",b,&8baf,&969 [3299]
3-&8baf
10000 BYTE=0                               [121]
10010 INPUT BYTE$                          [797]
10020 BYTE=VAL("&"+BYTE$)+BYTE             [1090]
10030 PRINT BYTE;                          [440]
10040 GOTO 10010                           [500]
```

Listing "CP/M-Verbesserungen"

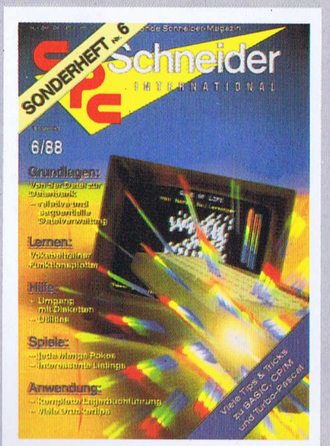
Wissen à la carte

Themen-Sonderhefte von DMV

Joyce Sonderheft 2
Das JOYCE-Sonderheft ist eine Sonderpublikation aus der Redaktion der PC International für JOYCE und JOYCE Plus. Es enthält ausschließlich noch nicht veröffentlichte Beiträge und Programme.
Aus dem Inhalt:
PASCAL-Compiler, Adress-und Archivverwaltung, Turbo-PASCAL Grafikerweiterung ohne GSX, Funktionstastenbelegung, 3D-Funktionsplotter
Best.-Nr.: 302 — 20,– DM*



Sonderheft 6/88
Grundlagen, viele Tips, nützliche Anwendungen und tolle Spielprogramme – ein Muß für jeden CPC-Anwender. Dieses Sonderheft behandelt das gesamte Spektrum möglicher CPC-Einsatzgebiete und bietet Informationen für jedermann. U.a. komplette Lagerbuchführung, relative und sequentielle Datei, neues Disketten-DOS, viele Lernprogramme, CP/M und Turbo Pascal und, und, und...
Best.-Nr.: 312 — 14,– DM*



Sonderheft 5/87
Power für Ihren CPC! Über 500 KByte leistungsstarke Software aus vielen Anwendungsbereichen bringen Ihren CPC auf Trab. Ob Einsteiger, Fortgeschrittene oder Profi – hier finden Sie alles, was Ihr Computer braucht. Viele Tips & Tricks, Grundlagenartikel, Spiele und Anwendungen lassen dieses Sonderheft zu einer unentbehrlichen Arbeitshilfe werden.
Best.-Nr.: 311 — 14,– DM*



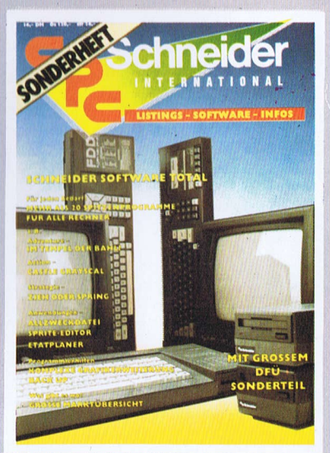
Sonderheft 4/87
Programmiersprachen – Anwendungen in Turbo-Pascal und mannigfaltige Informationen stehen im Mittelpunkt des 4. CPC-Sonderheftes. Mit über 200 Seiten praller CPC Informationen, Tips und wertvollen Programmen das ideale Sammelsurium für jeden CPC-Anwender. Interessiert!? – dann sollten Sie sich schnell entscheiden, denn es sind nur noch wenige Restposten verfügbar.
Best.-Nr.: 310 — 14,– DM*



Sonderheft 3/86:
Reviews – Spiele – Anwendungen – ein wahres Hit-Sammelsurium birgt das CPC-Sonderheft 3/86. Die 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.
Best.-Nr.: 309 — 14,– DM*



Sonderhefte 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 1/86!
SH-Nr.1 Best.-Nr.: 307 — 14,– DM*
SH-Nr. 2 Best.-Nr.: 308 — 14,– DM*



* Ohne Rücksicht auf die Anzahl der bestellten Zeitschriften berechnen wir für das Inland 3,– DM bzw. für das Ausland 5,– DM Porto und Verpackung. Bitte benutzen Sie den Coupon auf Seite 129.

DMV Verlag · Fuldaer Str. 6 · Postfach 250 · 3440 Eschwege

SPECIAL OFFERs!

für CPC 464-664-6128, nur auf 3"- Disketten

Original CPC-Software im Paket zu stark herabgesetzten Preisen

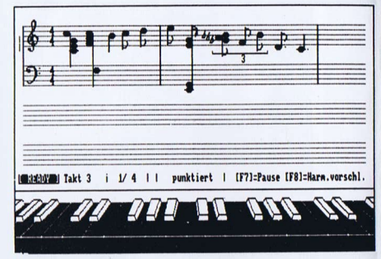
CPC SPECIAL OFFER No. 1
vier Programme
Best-Nr 204 nur

NUR 69,-DM *

COMPOSER-STAR (664/6128)

Ein Musikprogramm für alle Musikfans

Mit COMPOSER-STAR können Sie komplette Musikstücke oder nur ein paar Taktfolgen auf einfachste Weise erstellen! Natürlich stehen Ihnen auch eine Menge Korrekturhilfen zur Verfügung, um Ihrem Stück den letzten Schliff zu geben. Sie können auch mehrere Stücke verbinden, transponieren, die Tonhüllkurven und die Lautstärke verändern, den Rauschgenerator benutzen, Notenblätter drucken ... Das Programm ist sehr einfach zu bedienen; das beiligende Handbuch gibt auch dem Anfänger einen Einblick in die bislang fremde Welt der Musik. Das wichtigste jedoch ist, das Sie Ihre Werke in ein Basicprogramm umwandeln können, um es in eigenen Programmen zu verwenden.



COPY-STAR II

ist die ideale Befehlserweiterung für Druckerbesitzer, denn es stellt für alle gängigen Drucker Hardcopyfunktionen in verschiedenen Größen zur Verfügung. Sogar Farbbilder lassen sich schattiert ausgeben. COPY-STAR II können Sie leicht in eigene Programme einbinden.

MATHE-STAR

Vom Lehrer für Schüler

- lin. Gleichungssysteme
- Gleichungen 4. Grades
- Bruchrechnen
- Primfaktorenzerlegung
- Polynome
- Kurvendiskussion
- Integralrechnung
- Vektorrechnung
- Matrixrechnung

etc.

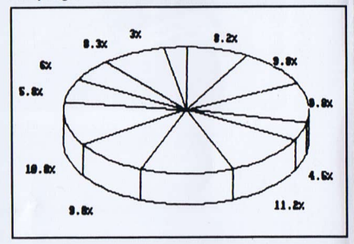
STATISTIC-STAR

Eine Grafik sagt mehr als 1000 Zahlen

Ein professionelles Grafik- und Statistic-programm zum Auswerten von Daten aller Art (Schule, Studium, Beruf, Hobby, Haushalt ...).

- Linien-, Balken- und Tortengrafik
- Betiteln von Grafiken
- 400 Daten direkt im Speicher
- Umfangreiche Editierfunktionen
- Umfangreiche statistische Berechnungen
- Hardcopyfunktion u.v.m.

Beispielgrafik STATISTIC-STAR



CPC SPECIAL OFFER No. 2
vier Programme
Best-Nr.205

NUR 69,-DM *

STAR-MON

Das Entwicklungssystem für Profis

- Assembler
- Editor
- Disassembler
- Monitor
- vier Breakpoints
- Trace-Funktion
- Bankswitch
- Memory Dump
- Diskettenmonitor
- u.v.m.

CREATOR-STAR

Ein Trickfilmdesigner für alle Hobbyregisseure auf dem CPC!

- Sprite-Designer
- Laufschrift
- Utilities
- Kulissendesigner
- Sprites mit 4 Unterpositionen
- Verbinden von Sprites
- Kulissen auch übereinandergelegt
- Eigene Programmiersprache mit Editor und Compiler

DISKSORT-STAR

Leistungsstarke Diskettenverwaltung, die keinem CPC-Benutzer fehlen sollte. DISKSORT-STAR verwaltet, archiviert, katalogisiert, druckt, ... Ihre Diskettensammlung auf einfachste Weise. Neben der reinen Diskettenverwaltung ist unter anderem noch ein kompletter Diskettenmanager enthalten. Auch in punkto Bedienungskomfort ist DISKSORT-STAR kaum zu schlagen.

DESIGNER-STAR

Grafikprogramm, mit dem man Bildschirmgrafiken komfortabel erstellen kann. Hilfsmenü auf Tastendruck – kein Joystick oder Maus notwendig.

CPC SPECIAL OFFER No. 3 (Games)
neun Programme
Best-Nr.107

NUR 49,-DM *

9 Spiele für Ihren CPC auf 3" Disketten

1) Stan und der Zauberstab – Ein deutsches Grafikadventure besonderer Art
2) Mr. PAC – Version des beliebten PACMAN-Spieles
3) Puzzle – Bringen Sie ein durcheinandergefallenes Bild wieder in Ordnung
4) Black Jack **5) Orion** **6) Labyrinth** **7) Memory** **8) Zick-Zack** **9)Slalom**

* Ohne Rücksicht auf die Anzahl der bestellten Programme berechnen wir für das Inland 3,– DM bzw. für das Ausland 5,– DM Porto und Verpackung. **Bitte benutzen Sie den Coupon auf Seite 129.**

SPECIAL OFFERS sind nur erhältlich bei: **DMV Verlag · Postfach 250 · 3440 Eschwege**