

SCHNEIDER

CPC-WELT

DM6/ÖS/48/SFR6



BTX-Modul

Neue Zukunft für den CPC

Im Test:

Drucker LQ 3500

Neu dabei:

C-Splitter

Neue Spiele

im Test:

*Basil
Firetrap
Trantor
und viele andere*



SCHNEIDER

IMPRESSUM

SCHNEIDER CPC-WELT erscheint monatlich in der CA-Verlags GmbH (i.G.)

VERANTWORTLICH FÜR DEN INHALT: Gert Seidel

GESCHÄFTSFÜHRER (und verantwortlich für Anzeigen): Werner E. Seibt

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN: Postfach 1161, 8044 Unterschleißheim
Tel.: 089/12980 11
Telex: 5214428 cav-d

Es gilt Preisliste Nr. 8 vom 1.1.1988
Media-Unterlagen bitte anfordern.

© 1988 by CA-Verlags GmbH (i.G.)
Heßstraße 90, 8000 München 40.
Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung.
Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für den Abdruck und die Aufnahme in den Softbox-Service zu den Honorarsätzen des Verlages.
Das Copyright und das Recht der wirtschaftlichen Verwertung gehen auf den Verlag über.
Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt.
Jedwede Verwendung ist untersagt. Namentlich gezeichnete Beiträge unserer Mitarbeiter stellen nicht unbedingt die Meinung der Redaktion dar.

Leser-Ecke

- Leserecke**
Preisberichtigung zu MS-DOS-Kinderprogrammen:
Billiger als angenommen
Anleitung zu Puzzle:
Nachtrag zum Listing in 12/87
Finanzgenie ist korrekt:
*Menüpunkt 1 ohne Drucker-
ausgabe*
Wo steckt der Tippfehler?
*Tips für einwandfreie Einlese-
routinen*
Verbesserungen für den ewigen Kalender:
Druckausgabe wird vereinfacht
„Karteikasten“ meldet
MEMORY FULL:
Wieviel paßt in die Datei?
Laufwerk-Selbstbau ist gefragt:
Noch einmal die Bezugsquellen
Einbindung einer Hardcopy:
Wichtig für alle Programme
MERGE oder CHAIN MERGE:
Tips gegen Betriebssystemfehler
HiSoft ist kein Turbo-Pascal:
*Gegen Irrtümer ist auch kein
Lehrer gefeit*
Listings als Barcode:
Zukunftsweisende Idee?
CPC-Monitor am Schneider PC:
*Es geht nur mit großen
Schwierigkeiten*
„Budget Manager“ nur für
Diskette:
Empfehlung als Alternative
Leser lassen „Dampf ab“:
*Berechtigte und unberechtigte
Kritik*
**CP/M-Programmtausch zwischen
CPC und C64:**
Zusätzliche Kosten entstehen
Bedienungshinweise zur
Diskettenbibliothek:
*Speicherkarte muß abgeschaltet
werden*
ab Seite 12

Nachrichten

*Neuheiten aus der
Computerszene*
ab Seite 10

Hardware

Schneider BTX-Modul
Lohnt die Ausgabe?
ab Seite 8
Der Drucker LQ 3500
Kein echter Schneider
ab Seite 44

Editorial

Computer in der Arbeitswelt
*Muß man um seinen
Job fürchten?*
ab Seite 50

Bericht

Programmiersprachen
für den CPC
Der Turm zu Babylon
ab Seite 90

Basic-Grundlagen

Das Terminal
*Input- und Tastatur-
abfrage*
ab Seite 86

**Problem
& Lösung**

Die Säulengrafik wird
verbessert
*Eingabe und automatische
Ausrichtung*
ab Seite 17

Ein Code wird geknackt
*Nicht nur für Geheim-
agenten*
ab Seite 32

**Tips —
Tricks — Kniffe**

Die Basic-Trickkiste
*Funktionstasten werden
erweitert*
ab Seite 20

Pascal

HiSoft-Pascal ausgebaut
Sogar Kreisdiagramme
sind möglich

ab Seite 29

C-Splitter

Sprechen Sie C?
Einzelheiten über die
modernste Programmier-
sprache

ab Seite 26

MC-Kurs

Der verständliche
Checksummer
CPC-Welt's Checker kann
nachvollzogen werden

ab Seite 35

Service

PC 1512
GEM contra GEM
Gibt es einen Fortschritt?

ab Seite 52

Bezugsquellenverzeichnis
Was gibt es wo?

Seite 16

Spiele-Hilfe

Die erste Hilfe gegen
Spielefieber
Mitmachen und gewinnen

ab Seite 48

Spiele-Reviews

Guild of Thieves
Edle Diebe

ab Seite 4

Renegade
Kampf im Straßen-
dschungel

Seite 6

Wild West Games
Wer kann am besten melken?

Seite 7

Firetrap
Der Feuerwehrmann als
Retter

Seite 89

Basil – Der Mäusedetektiv
Disneys Filme im CPC

Seite 92

Trantor
Der letzte Held

Seite 93

Classix 1
Spieletip für Sammler

ab Seite 94

Listings

Checker
Der Checksummer der
CPC-Welt

Seite 58

Privat-Manager
Ordnung im Privatleben

ab Seite 59

Invertierungsroutine
Die Grafik wird zum
Negativ

ab Seite 65

Solitär
Bekanntes Brettspiel
im Computer

ab Seite 66

Hyperinversi
Ein Spiel für Denker

ab Seite 70

Musikdemo
Wieder einmal etwas
zum Hören

ab Seite 72

Speed-Up
Beschleunigte Bild-
schirmausgabe

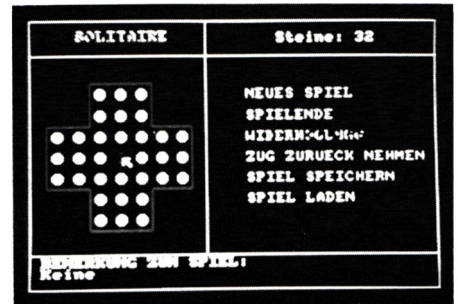
ab Seite 76

Anti-Era
Löschungen rückgängig
gemacht

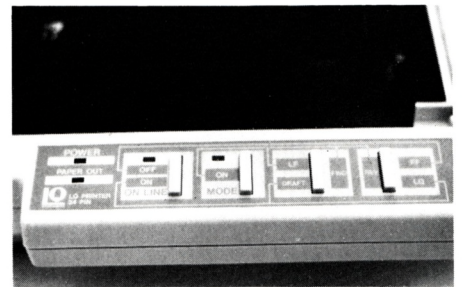
ab Seite 77

Polygraf
Diagramme für alle
Gelegenheiten

ab Seite 78



Unser Listing gegen Langeweile



Neue Hardware im Test –
Der Schneider LQ 3500



Bei Renegade herrscht Anarchie auf
den Straßen. Ist die brutale Spiel-
idee noch zu tolerieren?

Abenteuern Sie mal wieder!

Oft ist die Gemeinschaft der Computergamer in zwei Lager aufgeteilt: Für die einen sind Spiele nur dann halbwegs interessant, wenn sie einen möglichst heißen Sound und eine Spitzengrafik bieten. Vor allem die Bewegung von Raumschiffen, Karatekämpfern oder Rennwagen auf dem Bildschirm bedeutet für diesen Teil der Spielerschar das Höchste. Der andere Teil der Gamer bevorzugt das stille Glück der gelösten Aufgabe. Nach dem Motto: „Ich gegen den Parser“ oder „Wie überliste ich mein Programm?“ kämpft der Spieler nun oft nächtelang mit diversen Schwierigkeiten.

Guild of Thieves	
Grafik	
Sound	
Motivation	
Bedienung	englisch
50%	
Von Rainbird bei ariolasoft	

GUILD OF THIEVES

Den großen Sprung nach vorn taten die Entwickler der Firma Rainbird mit dem sehr bekannten Adventure „The Pawn“. Dieses Game war 1986 wohl die größte Neuentdeckung auf dem Adventure-Markt.

Mit seiner unglaublich schönen Grafik und dem starken Parser war es monatelang das Adventure schlechthin. „The Guild of Thieves“ heißt nun der Nachfolger von „The Pawn“. Dieses Adventure ist noch ein bißchen ausgereifter als sein Vor-

DAS ADVENTURE SCHLECHTHIN

gänger und verspricht dem Adventurer viel Vergnügen. Kaum erwähnenswert ist wohl die Tatsache, daß „The Guild of Thieves“ in englischer Sprache geschrieben ist. Der User sollte schon ein gediegenes Maß an Schulwissen mitbringen. Aber keine

Angst, zur Not tut's intensives Nachschlagen in den Dictionaries auch. Das Spiel wird in einer sehr ansprechend gestalteten Verpackung geliefert. Bei der neuesten Ausgabe ist neben der englischen Originalbeschreibung auch noch eine weitere Hilfe vorhanden: eine

ENGLISCHE UND DEUTSCHE ANLEITUNG

deutschsprachige Anleitung. Der Adventurer findet in der Packung auch sehr wichtige Tips, die ihm unter Umständen weiterhelfen können. Diese Hilfen und weitere Informationen zur Aufgabe allgemein findet er in der Verpackung enthaltenen ca. 40seitigen Magazin „What Burglar“. Es stellt für den Benutzer die wichtigsten Hilfe für die Lösung des Adventures dar.

WIE WERDEN SIE EIN ERFOLGREICHER DIEB?

Aufgabe des Spielers ist es, ein erfolgreicher Dieb zu werden. Wenn die zahlreichen Hürden der Aufnahmeprüfung bewältigt sind, so wird er in die altherwürdige Gilde der Diebe aufgenommen. Bis dahin ist es natürlich ein weiter Weg. Nach dem Laden erscheint ein sehr schönes Bild auf dem Screen. Es kann in seiner Größe variabel gestaltet werden. Durch Betätigen der Cursortasten läßt sich so der Bildschirm individuell aufbauen. Die Möglichkeit, verschiedene Textgrößen anzuwählen, unterstützt vor allem die Benutzer eines Farbmonitors.

DER PARSER

Ein Leser aus Duisburg berichtete der Redaktion, daß er nach einigem Herumprobieren Fehler im Parser von „The Guild of Thieves“ entdeckt habe. Dies schmälert aber die Leistung des Parsers nur wenig. Insgesamt ist er sicher im oberen Drittel der besten zehn anzusiedeln. Sätze wie: „Drink all that is on the tray except the gentlemen's drink“ werden mühelos verstanden und abgearbeitet. Das Spiel versteht viele verschiedene Verben und Wörter, mit denen der User mitteilt, was er möchte. Stehen z.B. verschiedene Gläser auf dem Tresen und der

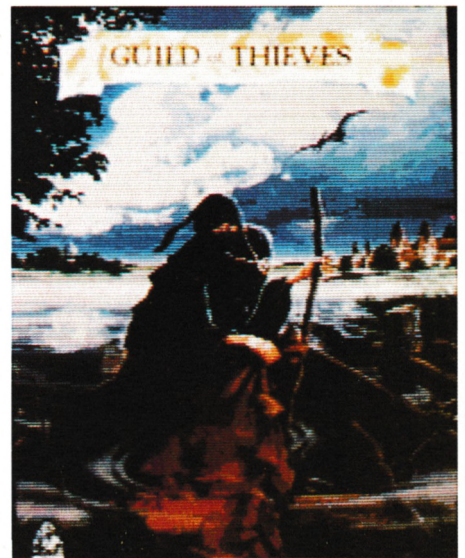
WELCHES GLAS DARF'S DENN SEIN?

Abenteurer sagt: „Get Glas“, so fragt das Programm: „Which one? The beer glas, the sherry glas or the wine glas?“.

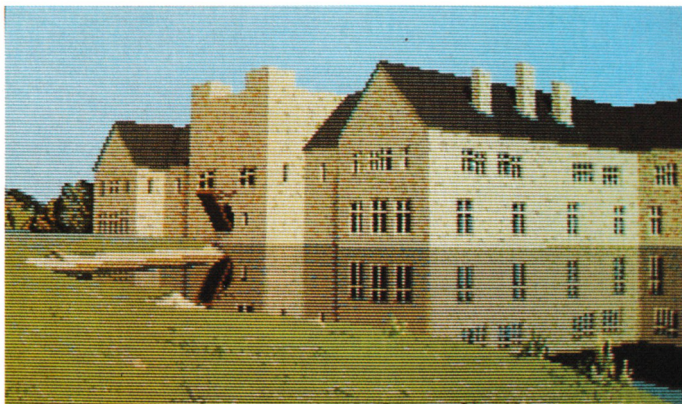
Nun hat der User die Möglichkeit, seinen Wunsch noch einmal zu differenzieren. Mit der Eingabe:

EINFACHE KOMMANDOS UND KOMPLIZIERTE SÄTZE

„The wine glas“ kann er so die richtige Auswahl treffen. Diese Nachfragen erlauben unseres Wis-

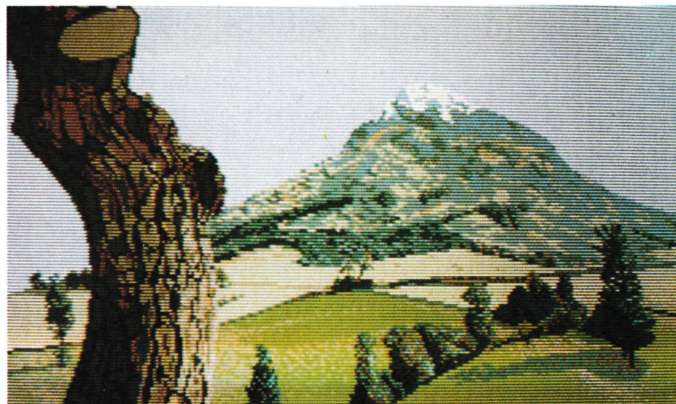


sens sonst nur die Infocom-Adventures. Der Parser versteht selbst komplizierte Sätze. Aber auch die simpelsten Ein-Wort-



Ausdrücke werden von ihm erkannt. Dies erspart dem Dieb in spe oft lästige Tipparbeit.

Durch diesen guten Parser sind der Fantasie kaum Grenzen gesetzt. „The Guild of Thieves“ bietet dem Adventurer außerdem eine



jede Softwaresammlung empfohlen werden. Durch den starken Parser und die ansprechende Grafik bleibt dieses Spiel sicher lange Zeit die

Nummer eins unter den aktuellen Adventures. Wir sind schon gespannt auf den Nachfolger. ■

GRAFIK DER SONDERKLASSE

Grafik der Sonderklasse. Die Bilder vermitteln genau die Atmosphäre, die für Adventures so wichtig ist. Im Laufe des Spiels werden bis zu 31 Bilder in den Arbeitsspeicher geladen. Durch die schon angesprochene Möglichkeit, sie in ihrer Größe variabel darzustellen, bleibt es dem User überlassen, wieviel von dem

VARIABLE BILDSCHIRMDARSTELLUNG

jeweils aktuellen Bild er sehen will. So kann die Grafik quasi ganz abgeschaltet werden. Ihre Ausgefeiltheit und das Ausgabe-Handling sind mit ein Grund für den Erfolg dieses Spiels.

SPEZIALBEFEHLE

„The Guild of Thieves“ kennt einige Spezialbefehle, die den Umgang mit dem Programm erleichtern. Mit Again wird der letzte eingetippte Befehl wiederholt. Das Kommando Exits zeigt dem Spieler sämtliche begehbaren Richtungen an. So behält er jederzeit den Überblick über vorhandene Ausgänge. Mit Score wird der aktuelle Punktestand abgefragt. Restart bringt den Abenteurer wieder an die Ausgangsposition des Spiels zurück. „The Guild of Thieves“ kann uneingeschränkt für

Kerovnia Guild Of Discreet Entry And Removal Operatives¹ (Dornbrook Branch)

INDENTURE AND CONTRACT OF SERVICE

Wherein and heretofore as otherwise previously notwithstanding the said applicant postulant apprentice novice trainee or otherwise hereinafter designated PERSON² has heretofore formally and earnestly applied for membership of to for by and with the said Guild

AND wherein and heretofore this contract has been previously drawn up drafted engrossed and otherwise written and inscribed most especially bereft of excluding and otherwise without any form of COMMA or SEMI COLON or any other form of punctuation which would otherwise allow the said applicant postulant apprentice novice trainee or otherwise to make HEAD OR TAIL of what this is about

AND WHEREIN the said contract indenture agreement and earnest of good intent is DELIBERATELY AND POMPUSLY written in difficult obscure archaic or otherwise out of date language which we do NOT UNDERSTAND either

IT IS THEREFORE AGREED covenanted contracted bargained engaged stipulated promised arranged settled signed sealed and executed that I, _____ do hereby covenant contract agree and do we REALLY have to go through all that again? to go off and rob all sorts of really great stuff from this weird place and moreover to hang about and not moan while we think of a more POMPUS and FRIGHTENING way of expressing that last bit³

¹Hereinafter named called described as and otherwise else um er notwithstanding and um so forth referred to as "Guild of Thieves". Plus also where we wish hereinafter etcetera, also heretofore, um, where were we? Do you know what it's like to have to WRITE this stuff? Everyone laughs at lawyers, oh yes, ha ha ha, but it's no fun, not really, not compared with just hanging round on street corners talking dirty. Okay, you get rich, but money can't buy you happiness. Mind you, happiness can't buy you money, so it's as broad as it's long, hereinafter.

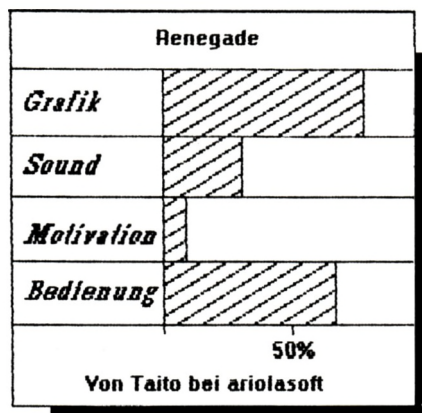
²Mirabilis condidisti et mirabiliter redemisti.

³Hereinafter referred to as the pre-existing condition of whoops oh rats, bum, there was a hyphen there, hyphens aren't legal jargon, oh RATS neither are commas, right, that's it, I've had enough, stuff the law, I'm going to Malibu. Sort it out yourselves.

Renegade

Anarchie auf den Straßen!

Der Alptraum einer Großstadt: In dunklen Straßen und natürlich ganz besonders in einsamen U-Bahnhöfen lauert die Gefahr. Wenn Sie derlei Dinge vermissen, dann sollten Sie zu dem neuesten Spiel von Imagine mit dem Titel „Renegade“ greifen. Dort werden Sie nämlich ständig überfallen und beziehen Prügel nach Strich und Faden.



Stellen Sie sich vor, Sie wollen zu einem Stelldichein gehen und fahren mit der U-Bahn. Kaum steigen Sie aber aus dem Zug, sehen Sie sich schon den Vertretern der Halbwelt gegenüber. Und die sind Ihnen gar nicht freundlich gesinnt! Sollte Ihnen so etwas tatsächlich einmal passieren, dann nehmen Sie besser die Beine in die Hand. Mit Ihrem CPC dürfen Sie sich jedoch einmal auf eine Prügelei einlassen.

Es beginnt noch recht einfach: Erst sehen Sie sich reinen Schlägern gegenüber, mit denen Sie aber durch geschicktes Verteilen von Fußtritten und Kinnhaken fertig werden. Dann kommen die Lederjackenträger, die mit Knüppeln auf Sie einschlagen wollen, um Sie zu Boden zu bringen. Haben Sie auch die überwunden, kommt der Anführer der Outlaws, die nur ihren eigenen Gesetzen folgen. Hier müssen Sie Ihre ganze Geschicklichkeit einsetzen, um nicht zu unterliegen.

STELLEN SIE SICH MIT DEM RÜCKEN AN DIE WAND!

Am besten ist es, wenn Sie Ihre Helden mit dem Joystick so lenken, daß er mit einer Seite an einer

Wand steht. Dann ist es für die Angreifer nicht möglich, Sie von zwei Seiten zu traktieren und Sie haben bessere Chancen, mit heiler Haut davon zu kommen.

Leider ist das Spiel so programmiert, daß Sie den Joystick nur zum Laufen in die vier Richtungen benutzen können. Wollen Sie sich jedoch verteidigen, müssen Sie f0 bzw. die kleine Enter-Taste drücken, um sich nach links oder rechts zur Wehr zu setzen. Zumindest trifft dies bei der Kassettenversion für den 464 zu. Die Suche nach den richtigen Funktionstasten ist ein bißchen umständlich und trägt nicht gerade zum Spielgenuß bei. Wobei der Ausdruck „Spielgenuß“ sowieso in Anführungszeichen stehen sollte, denn ein erheiterndes Spiel ist es wegen der Brutalität sicher nicht. Es dürfte kaum zur Entspannung beitragen.

ÜBERALL LAUERN SCHLÄGER

Haben Sie den U-Bahnhof heil überstanden, kommen schon die nächsten Schlägertypen auf Sie zu, diesmal im Hafenviertel. Hier werden Sie von Motorrad-Rockern angegriffen. So geht das Spiel dann weiter; über dunkle Seitenstraßen zu einem Trainingsplatz für Kampfsporttruppen. Und überall wird Ihnen aufgelauert.

Wer dieses Spiel überstehen will, muß über besonders trainierte Finger verfügen, um in rasender Geschwindigkeit die kleine Enter- oder die f0-Taste zu drücken.

Was haben die Programmierer nur für einen Geschmack? Warum werden nicht mehr Spiele in der Art von „Hexenküche“ oder „Sorcery“ geschrieben? Oder verkaufen sich diese brutalen Spiele besser? Diese Annahme liegt nahe, wenn man sich

die Hitlisten über die umgesetzten Spiele ansieht. Oder wollen die Käufer solche Programme haben, damit sie die Brutalität im Alltagsleben auch noch nach Feierabend am Bildschirm, vom bequemen Sessel aus, verfolgen können? Besonders abstoßend die Szene, wenn ein Gegner, der mehrmals niedergeschlagen wurde, sich in einer Blutlache wälzt.



SCHLECHTE IDEE, GUTE GRAFIK

Doch wie so oft bei brutalen Spielen, ist auch dieses von der Grafik her ziemlich weit oben anzusiedeln. Da haben die Programmierer mit Details nicht geizigt. Ihnen ist eine gute grafische Gestaltung gelungen. Auch der Sound ist recht flott.

Das Spiel kostet als Diskette DM 44,95. In der Kassettenversion müssen Sie DM 29,95 auf den Ladentisch legen. Gerade die Mischung aus guter Programmieretechnik und einer schlechten Idee zwingt dazu, das Spiel nicht zu empfehlen. Handelt es sich am Ende gar um ein Produkt, welches mit einer Indizierung werben will? JE

Imagine by Ariolasoft

Motivation: 40 %

Grafik: 90 %

Sound: 80 %

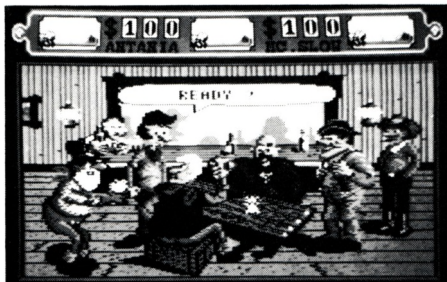
Handhabung: 50 %

Western Games

Wer kann am besten melken?

Ein neues Spiel ist auf dem Markt, bei dem neben der eigentlichen Joystick-Akrobatik auch der Humor nicht zu kurz kommt. Nicht das sture Abschießen von Außerirdischen ist gefragt, sondern das geschickte Verhalten am Freudenknüppel, ohne die ganze Sache allzu ernst zu nehmen.

Verstaubt durch den langen Ritt, erreichen Sie ein kleines Städtchen am Rande der Prärie. In der Bar, in der Sie sich erfrischen wollen, stehen einige wild aussehende Cowboys auf und machen Ihnen klar, daß jeder Fremde, der in dieses Städtchen kommt, zuerst einige Kraftproben durchstehen muß, bevor er aufgenommen wird. Mißmutig stimmen Sie zu, denn Sie wollen ja hier einige Tage bleiben, um Ihrem Pferd eine Erholungspause zu gönnen.



WER GEWINNT BEIM „ARMDRÜCKEN“?

So beginnt die Story zu dem neuen Game von „Magic Bytes“. Und nachdem Sie sich so richtig eingestellt haben, geht es dann gleich los.

Als erstes müssen Sie Ihre Kräfte mit dem stärksten Kerl des Ortes messen, und zwar durch das im Wilden Westen so beliebte „Armdrücken“. Dazu sitzen Sie Ihrem Gegner an einem Tisch gegenüber, den Ellenbogen des rechten Arms aufgestützt. Sie packen die Hand Ihres Gegners, und auf ein Kommando hin versucht jeder, den Arm des anderen auf die Tischplatte zu drücken.

Doch dies ist nur die erste Probe, die Sie bestehen müssen. Es fol-

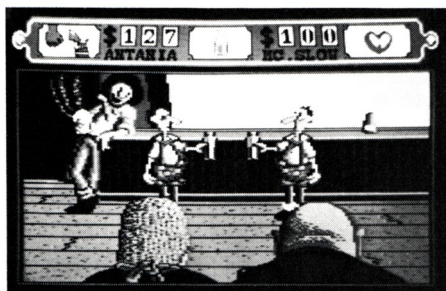
gen noch einige andere: Bierkrugzerschießen, Priemspucken, Tanzen, Kuhmelken und Wettessen. Bei jedem dieser Wettbewerbe müssen Sie möglichst gewinnen, um eine hohe Punktezahl zu erreichen.

GEFRAGT SIND JOYSTICKAKROBATEN!

Bei den einzelnen Wettbewerben kommt es auf das richtige Bewegen des Joysticks in einer ganz bestimmten Reihenfolge an. Wenn Sie die nicht einhalten, können Sie auch keine Punkte sammeln. Beim Kuhmelken ist dies ganz besonders wichtig, denn sonst haut Ihnen die Kuh mit ihrem Schwanz immer wieder auf den Kopf. Und das sollte ja möglichst nicht sein!

Beim Wettessen achten Sie darauf, nicht so viel Luft zu schlucken, denn sonst müssen Sie rülpfen, und dies bringt Ihnen sicher keine Punkte ein.

Beim Tanzwettbewerb führt eine Lady auf der Bühne des Saloons einige Tanzschritte vor, die Sie mit

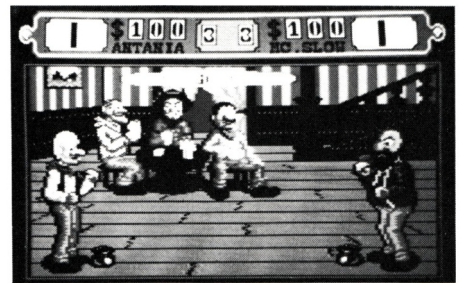


Ihrem Männlein dann nachvollziehen sollen. Auch das ist nicht ganz einfach, denn einen Walzer auf einer Westernbühne zu tanzen, ist nicht jedermanns Sache.

Auch beim Bierkrugabschießen heißt es aufpassen. Denn wenn Sie

nicht genau zielen, kann es sein, daß Sie versehentlich den Mann „umlegen“, der den Krug hält. In diesem Fall haben Sie dann sicherlich Schwierigkeiten zu erwarten.

Es gibt bei jedem Wettbewerb immer einen kleinen Gag, auf den Sie besonders achten müssen und der das ganze Programm etwas amüsant macht. Wir jedenfalls konnten uns während des Tests ein kleines Schmunzeln nicht verkneifen.



FAZIT

Ein nettes Spiel, so richtig zum Entspannen. Die Grafik ist recht gut gelungen; besonders die Figuren, die Sie auf dem Bildschirm sehen, sind ganz lustig. Der Sound ist ebenfalls gut anzuhören. Nicht ganz leicht ist die Steuerung zu handhaben. Sie müssen schon ganz schön mit dem Joystick umherrühren, um Erfolg zu haben. Aber schließlich wollen Sie ja gewinnen, so bleibt Ihnen nichts anderes übrig, als sich zu bewähren. *JE*



Magic-Bytes by micro-Partner

Motivation: 90 %

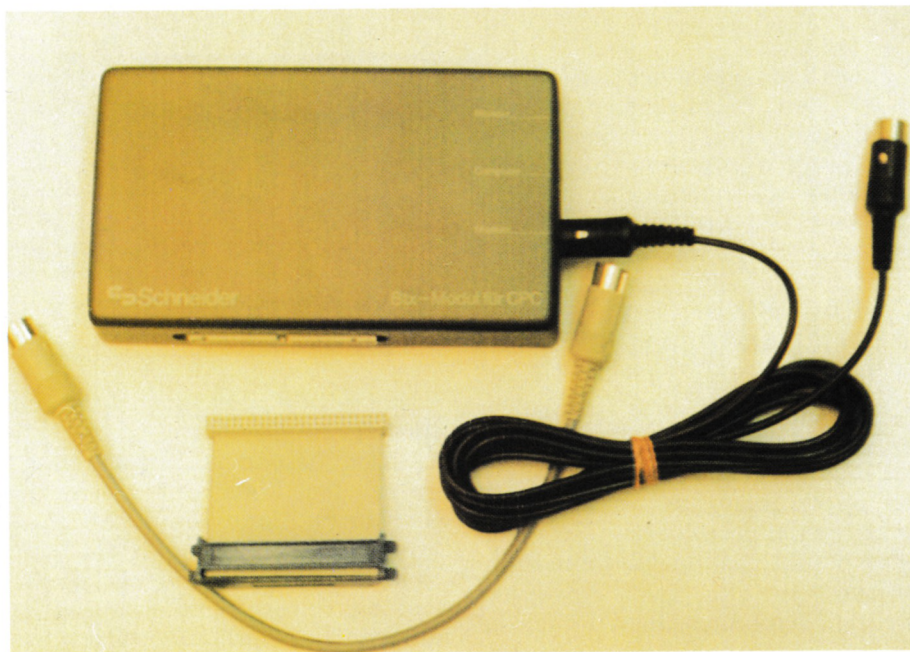
Grafik: 80 %

Sound: 80 %

Handhabung: 50 %

Spät entdeckt: BTX Modul für den CPC

BTX (Bildschirmtext) soll nach dem Willen der Bundespost endlich das Zeitalter der Telekommunikation einleiten. In anderen Ländern (Beispiel: England) sind Bankabbuchungen oder Einkäufe am heimischen Bildschirm schon längst Alltag. Hierzulande kommt nun endlich das Schneider Modul zum CPC heraus. Ob es technisch in der Lage ist, dem Mediendienst zu neuem Aufschwung zu verhelfen, soll der Test der CPC-Welt klären.



Relativ spät – es sind bereits 300 000 CPCs in Deutschland verkauft – bringt Schneider ein BTX-Modul für den populären Homecomputer auf den Markt. Es dürfte allerdings so sein, daß nicht einmal der Entschluß zur Entwicklung dieses Gerätes in Türkheim gefaßt wurde. Wer den Aufkleber an der Unterseite beachtet, der wird neben der Postprüfnummer den Namen der Firma Siemens, Hausentwickler der Bundespost, bemerken. Wichtiger ist aber noch der Hinweis, daß es sich hier um einen Universal-Decoder handelt. Da stellt man sich sofort die Frage, unter welchem Firmennamen und in welcher Farbe das Gerät sonst noch zu erhalten ist.

KOMPLETTE LIEFERUNG MIT FOLGEKOSTEN

Doch bleiben wir bei Schneider. hier wird komplett geliefert, sogar

das Antragsformular für einen Anschluß ist enthalten. Aber eines ist klarzustellen: Mit dem Kaufpreis des Decoders ist man noch lange nicht „dabei“. Die Post verlangt ebenfalls eine Anschlußgebühr (65,- DM) und eine monatliche Grundgebühr von 8,- DM. Diese Kosten entstehen zusätzlich zum eigentlichen Telefonanschluß, der Grundvoraussetzung ist. Im Betrieb kommen dann noch die Einheiten eines Nahtarif-Gesprächs dazu. Darüber hinaus verlangen einige Firmen auch Gebühren für bestimmte Dienstleistungen. Schneiders Lieferung ist also bestenfalls technisch komplett, die Gesamtkosten sind nicht abgedeckt. Mit dabei sind die Verbindungskabel zum Monitor, zum Postmodem und das Flachbandkabel zum Floppy-Anschluß. Das Diskettenlaufwerk kann trotzdem noch angeschlossen werden.

Die Frage nach dem Hersteller des BTX-Moduls wird zweifelsfrei mit dem Aufdruck „Siemens“ auf

der Platine beantwortet. Noch zwei Dinge fallen auf: zum einen das Eprom, welches wahrscheinlich das einzige Schneider-spezifische Teil sein dürfte. Es ist auf einem Sockel installiert und kann beim Hersteller sicherlich schnell gegen die Software anderer Firmen ausgetauscht werden. Zum anderen Nachteil: die Buchsen der DIN-Stecker. Sie haben nur eine Plastikhalterung und sind sehr fest mit der Platine verlötet. Häufiges Anschließen des Gerätes dürfte hier zuallererst zum Leitungsbruch führen. Also möglichst nur einmal installieren!

VOM STANDARD UND LUXUS EINER BTX-STATION

Bildschirmtext unterliegt einer gewissen Normung, die Software muß damit zurecht kommen. So werden Befehle mit dem Multiplikations-Stern (*) eingeleitet und mit einem Rautensymbol (statt ENTER) abgeschickt. Für letzteres muß das Nummernzeichen (#) herhalten, es findet sich ja bislang nur auf BTX-Konsolen. So unwesentlich dieses Detail erscheint, man sollte sich die Befehlskürzel gut merken. Während des Betriebes erscheint ja kein großes Menue, sondern man wird von der Anbieterseite her bestenfalls zum Weiter- oder Zurückblättern aufgefordert, aber eben mit den falschen Tastensymbolen. Allerdings ist es nicht sehr schwierig, mit dem System zurecht zu kommen. Man kann eine Seitennummer aufrufen, blättern, bei Übertragungsfehlern (kostenlos) wiederholen lassen, Eingaben machen und Dialogseiten nach Beendigung abschicken. Selbst der Aufruf der Zentrale (mit dem jede Verbindung beginnt) und die Abwahl werden durch Funktionstasten geregelt. Die typische Verbindung sieht dann wie folgt aus:
I btx (Einschalten des BTX-Modus)
Shift-f1 (Anwahl der BTX-Zentrale).

Danach kann man sich durch das Angebot durchblättern oder man verkürzt die Zeit, indem man eine bereits bekannte Seite anwählt. Zur

Beendigung bzw. zur Abwahl der BTX-Zentrale, steht die Funktionstaste f4 zur Verfügung.

DER VORTEIL EINES COMPUTERS

Bisher wurden jene Funktionen beschrieben, die ein BTX-Terminal ohnehin erfüllen muß, und es wäre schade gewesen, hätte es Schneider dabei belassen. Ein Computer kann eine ganze Menge mehr, und die Software des Eproms unterstützt diese Fähigkeiten. Da wäre zunächst einmal der ASCII-Modus, der zwar nichts mit BTX zu tun hat, aber zum Beispiel eine Datex-P Verbindung erlaubt. Wie der Name schon sagt, werden dabei als Steuerzeichen nur Carriage Return (CR), Line Feed (LF) und Form Feed (FF) zugelassen. Weiterhin wird in eine 80-Zeichen-Darstellung umgeschaltet und einige Optionen, etwa die deutsche Tastaturbelegung, gelten nicht mehr.

Der Capture Modus ist für den BTX-Betrieb besonders wichtig, hilft er doch, teure Gebühreneinheiten zu sparen. Mit dieser Funktion, im DFÜ-Umgang als „Download“ bezeichnet, können bis zu zwölf Bildschirmseiten im RAM des CPC auf Diskette oder Band abgespeichert werden. Da zuerst mit dem verfügbaren Speicher des CPC gearbeitet wird, sollte man sich hin und wieder die noch vorhandene Kapazität anzeigen lassen. Bedauerlicherweise endet der Capture-Modus ganz automatisch, wenn der Speicher überzulaufen droht. Während des BTX-Betriebes wird der Aufbau einer Seite natürlich nicht beschleunigt. Der finanzielle Vorteil entsteht erst dann, wenn man sich diese Texte in aller Ruhe anschauen kann, ohne „On Line“ zu sein. Die Dateien werden unter einem frei zu vergebenen Namen auf der Diskette abgelegt.

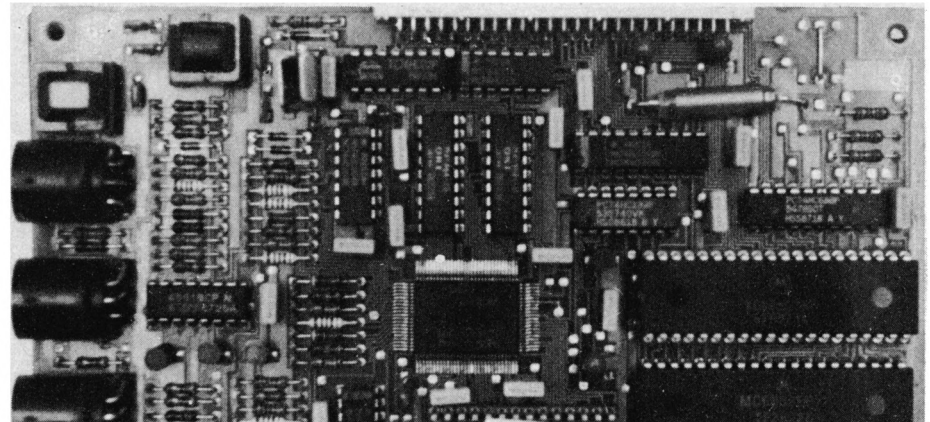
Daß man diese Bilder wieder laden kann, ist selbstverständlich. Zu erwähnen ist noch, daß im BTX-Modus des CPC auch die Optionen eines Diskettenkataloges nicht fehlen. Darüber hinaus kann man sich auch eine Pausenfunktion zunutze machen, bei der die „Vorzeigedauer“ der Bildschirmseite zwischen einer und neun Sekunden vorprogram-

mierbar ist. Damit ist die Wirkung einer Diashow zu erreichen. Soweit es sich um Textinformationen handelt, können diese auch auf den Drucker ausgegeben werden.

Bei der Eingabe von Daten stehen zwei verschiedene Tastaturbelegungen zur Verfügung, zwischen denen auch während des Betriebes umgeschaltet werden kann. Es handelt sich dabei um die amerikanische ASCII-Belegung und um eine Schreibmaschinen-ähnliche Zuordnung mit deutschen Umlauten.

ANLEGEN UND BENUTZEN VON MAKRODATEIEN

Gerade bei BTX bewahrheitet sich der Spruch „Zeit ist Geld“. Wer den Postdienst wirklich nutzen will, der muß aus dem Hauptmenü schnell das Angebot oder Formular seiner Firma herausfinden. Wenn man sich dabei von Verzeichnis zu Verzeichnis durchwählt, werden die Kosten unnötig erhöht. Beim CPC-Modul



kann man sich durch die Definition von Makros behelfen. Bis zu 36 dieser Befehlsketten sind möglich, und jede einzelne kann durch ihre Kennzahl aufgerufen werden. Einmal erstellte Makros sollten abgespeichert und vor Bedarf eingelesen werden. Unter dieser Option bietet die Software sogar noch Raum für ein Lernprogramm, es kann also nichts schiefgehen.

TELESOFTWARE

Als letzter Punkt nun die Telesoftware-Option: Theoretisch kann ein Programm über BTX geholt und auf Diskette abgelegt werden. Normalerweise wird in der „On-Line“ Verbindung gespielt und dadurch stei-

gen die Gebühren; Telefoneinheiten, die man sich leicht durch ein Download-Verfahren ersparen kann. Das Verfahren gleicht dem der Seitenspeicherung, so daß weiter nichts darüber gesagt werden muß. Allerdings sei vor großem Optimismus und dem Glauben an einen „Software-Pool“ gewarnt. Sicherlich gibt es einige kleine Programme, aber deren Spielwert und Niveau sind gering, der Spaß ist schnell dahin.

FAZIT: TECHNISCH EINWANDFREI, ABER ...

Am BTX-Modem der Firma Schneider wird es nicht liegen, wenn dieser Postdienst zu wenig in Anspruch genommen wird. Technisch ist der Decoder einwandfrei, und die eingebaute Software macht sich den Computer zunutze. Makrodefinitionen und Download-Möglichkeiten machen die Arbeit leicht und

schnell. Schwieriger wird es, einen wirklichen Nutzen aus der Sache zu ziehen. Der Decoder kostet 398,- DM, die Einrichtung des Postmodems weitere 65,- DM. Eine Summe, die man mit dem momentanen Angebot der Post nicht mehr wettmachen kann. Tatsache ist, daß BTX nicht den rechten Anklang in der Bevölkerung fand und kaum genutzt wird. Die Post hat richtig erkannt, daß man Freaks benötigt, die BTX nur als technische Spielerei betreiben, um das System in Schwung zu bringen. Besitzer von Homecomputern kommen da gerade recht. Solange allerdings das Angebot trotz der hohen Einstiegspreise gering bleibt, sollte man jenen User allerdings mehr zur DFÜ raten. G.S.

Nachrichten & Neuheiten Nachrichten

DIE ATARI-FÜHRUNGSMANNSCHAFT wurde verstärkt. Die außergewöhnlichen Markterfolge von Atari Corp. Deutschland GmbH und die sich daraus ergebenden Aufgaben haben die Gründung des neuen Geschäftsbereiches Technologie, Forschung und Entwicklung erforderlich gemacht. Als Geschäftsführer wurde Helmut Joswig gewonnen. Seit 1980 war er für die Produktion der Commodore-Büromaschinen verantwortlich. In der Branche gilt der diplomierte Ingenieur als Experte für die Großserienproduktion



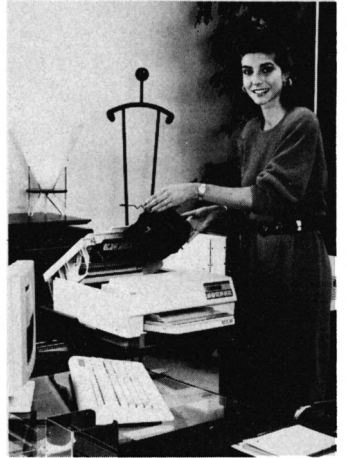
von Computern. Atari dokumentiert mit der Einrichtung dieses neuen Geschäftsbereiches in Deutschland die Wichtigkeit des deutschen und europäischen Marktes für den Gesamtkonzern. *DK*

WORDSTAR, WORDSTAR und kein Ende. Das wohl best verkaufte (und bestkopierte) Textverarbeitungsprogramm liegt in einer neuen Version namens 4.0 EXTRA vor. Selbstverständlich ist auch MailMerge mit dabei. Eine deutsche Rechtschreibhilfe soll mit einem Wortschatz von 90000 Wörtern über die Hürden helfen. Hinter EXTRA verbirgt sich ein Programmelement, welches die Textverarbeitung zum Publisher macht. Dem Benutzer bisheriger Versionen wird dabei sofort das WYSIWYG-Prinzip (What You See Is What You Get) auffallen. In die Druckerpalette sind für diesen Programmteil Laserdrucker und 24-Nadel-Drucker hinzugefügt worden. Alle gängigen Grafikkarten der MS-DOS-Computer werden vom neuen Wordstar 4.0 EXTRA unterstützt. Micropro ist es damit relativ spät gelungen, auf dem DTP-Markt zuzugreifen. Aber immerhin kann man auf ein Kundenpolster von über drei Millionen zurückgreifen. Möglicherweise heißt der zukünftige Standard von DTP-Programmen ebenfalls Wordstar. *GS*

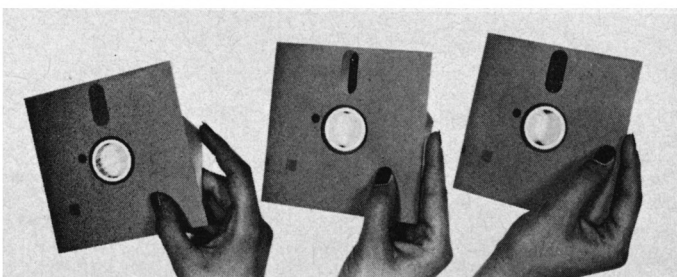
BEIM START DER EUROPAPARAKETE Ariane haben erstmals auch Amigas mitgemischt. Zur Errechnung der Startphase und Veranschaulichung der Daten beim 19. Ariane-Start in Kourou/Französisch-Guayana wurden unter anderem auch aufgerüstete Amiga 2000 eingesetzt. Vom Start bis zum Aussetzen der beiden Satelliten zeigt der Amiga die jeweiligen Raketenpositionen auf der Flugbahnkurve. Der 2000er wurde natürlich entsprechend aufgerüstet, ein Motorola 68020 sowie, als Mathematik Koprozessor, ein 68881 sorgen für eine rasante Taktschwindigkeit des Gerätes von über 14 MHz. *TS*

WO SOLL MAN PRODUZIEREN, fragte sich der Druckerhersteller Star Micronics. Die zunehmende Automatisierung der Fabrikanlagen macht den Produzenten relativ unabhängig vom Lohnniveau des Landes. Die Erhöhung des Yen-Kurses und die japanische Problematik des Handelsüberschusses machen das Land des Lächelns für Investoren uninteressant. So kommt das gute alte Europa wieder in Mode. Hier ganz besonders Großbritannien, wo den Garantien zukünftiger Arbeitsplätze steuerliche Vergünstigungen geboten werden. Ab März 1988 möchte Star also die Drucker, die für Europa bestimmt sind, in Tredekar, 250 km westlich von London, bauen. Ab Januar will man bereits den neuen Laserprinter über ausgewählte Fachhändler vertreiben. Das Gerät mit

Ein-MByte-Speicher kostet zwar 6500,- DM, soll aber die Betriebskosten niedrig halten, da ein Ausdruck lediglich mit 7,5 Pfennigen zu berech-



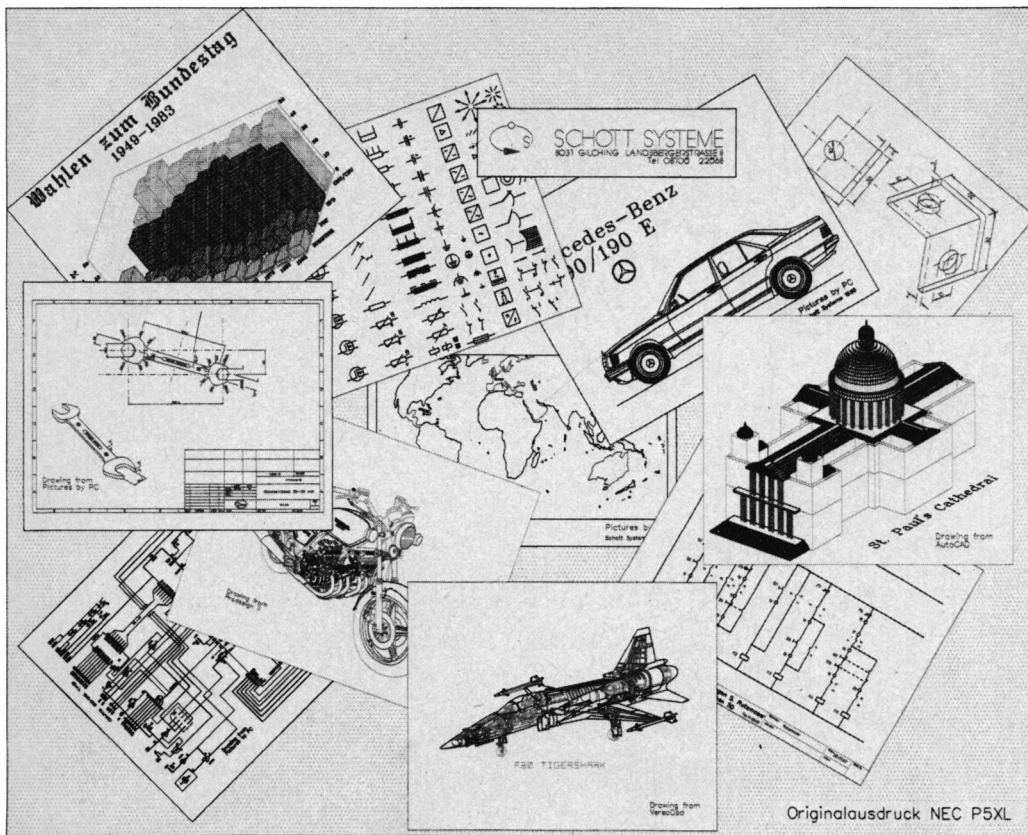
nen ist. Den Markt für Laserprinter sieht man gar nicht so günstig, man prophezeit bestenfalls eine Wachstumsrate von 3,5 %. Der Einstieg in die neue Technologie war aber unvermeidlich, alleine schon, um die Produktpalette der Printer abzurunden. Für die Besitzer des NB 24, dem bisherigen Spitzenprodukt der Firma, gibt es übrigens neue Softwaretreiber, die von Star kostenlos an alle Fachhändler ausgegeben wurden. *GS*



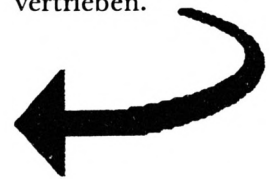
EINE LÄNGST FÄLIGE Neuerung ist im September 1987 beim deutschen Patentamt angemeldet worden. Es handelt sich um einen automatischen Verschluss der Schreib-Lese-Öffnung bei 5.25-Zoll-Disketten. Besonders für teurere Dis-

ketten empfiehlt sich diese Sicherung. Ob und wann die Diskette auf den Markt kommt, hängt davon ab, wie rasch sich Produktion und Vertrieb auf die Entwicklung stürzen. Die Mehrkosten dürften unter einer Mark liegen. *GS*

& Neuheiten Nachrichten & Neuheiten



SPEZIELL FÜR DIE NEC-Drucker hat die Firma Schott einen Universal-Druckertreiber entwickelt, der die bestmögliche Auflösung garantiert. So kann zum Beispiel der P6 – einer der meistverbreiteten Farbdrucker – nun auch im Farbmodus 360 Pixel pro Zoll erreichen. Voraussetzung ist ein beliebiges CAD- oder Grafikprogramm, welches Dateien im Format von HPGL (Hewlett Packard Graphic Language) oder als Vektor-Datei ablegt. PinPlot wird für alle 24-Nadel-Drucker und für grafische Laserdrucker vertrieben. **GS**



CHESS BASE STELLT sicher eine der interessantesten Anwendungen für Schachspieler dar. Diese Schach-Datenbank ist nun endlich auch für IBM-kompatible Rechner erhältlich. Sie wurde vom derzeitigen Schachweltmeister Gary Kasparov mitentwickelt. Nachdem das Programm für den Atari ST ein großer Erfolg wurde, war es eigentlich nur eine Frage der Zeit, wann auch die PC-User in den Genuß dieser Software kommen würden. Vor allem für Vereinsspieler stellt Chess-Base eine wirkliche Hilfe im Umgang mit den schon einmal gespielten Partien dar. Eine echte Abwechslung im Schachprogramm-Einerlei. **DK**

SOEBEN ERSCHIENEN IST die 9. Auflage des „TTL Data Book, Volume

1“ von Texas Instruments. Auf 1024 Seiten enthält dieses Datenbuch die Daten der bekannten TTL-Familien: Low Power Schottky, Schottky und Standard-TTL. Neben den reinen elektronischen Kennwerten sind dem Buch auch Informationen über Austauschtypen des Wettbewerbs, alle Gehäuseformen, sowie die neuen Logiksymbole zu entnehmen. „TTL Data Book“ gibt es mit der ISBN-Nr.: 3-88078-067-6 (nur in englischer Sprache) zu 36,50 beim Fachbuchvertrieb Wichmann & Partner. **JS**

DER FIRMENKRACH ZWISCHEN Amstrad (CPC, PC1512/1640) und Schneider (AT) nähert sich dem Höhepunkt. Wie unsere Kollegen von der englischen Zeitschrift „Your Computer“ meldeten, ist die Trennung

für Mai 1988 geplant. Zuvor will Amstrad aber schon ein deutsches Vertriebsnetz aufgebaut haben und, wenn möglich, sogar eines, welches die der Konkurrenten Atari, Commodore und Schneider um Längen schlägt. Alan Sugar scheint – wenn man den Meldungen glauben will – nicht zuletzt über den AT verärgert. Aber auch Schneider bereitet sich auf die Stunde der Wahrheit vor. Mit viel Aufwand wird in einer süddeutschen Tageszeitung um Personal geworben. Gesucht werden Informatiker, Nachrichtentechniker und Konstrukteure. Die Zeichen stehen auf „Eigenentwicklung“, und demzufolge müssen sich die zukünftigen Mitarbeiter natürlich besonders im MS-DOS Bereich auskennen. Gefordert werden aber auch Kenntnisse in 68000 Assembler und in der Programmiersprache C. Schließlich

sollen nicht nur Personal-, sondern auch Mikro-Computer entwickelt werden, wenngleich immer noch unklar ist, was man in Türkheim unter letztgenannter Kategorie zu verstehen hat. Das Rennen ist gestartet, und während man in Großbritannien wartet, ob es Schneider-Computer geben wird, darf man hierzulande gespannt sein, mit welchen Produkten Amstrad überraschen will und – selbstverständlich – was Schneider als Gegenargument einfällt. Auf der CeBit in Hannover wird jeder die Möglichkeit zum Vergleich haben. Erstmals stellen beide Firmen auf der gleichen Messe aus. **GS**



DAMPF ABLASSEN

Ich möchte ein wenig Dampf über Schneider aktiv ablassen. Dies betrifft nicht den Inhalt, sondern den schlechten Druck Ihrer Listings. Man kann kein „B“ von einer „8“ unterscheiden und das „&“ ist nur zu erraten. Wie dabei Programme bei den Käufern Ihrer Zeitung zum Laufen kommen, ist mir ein Rätsel. Oder ist es so – diese Vermutung liegt nahe –, daß Sie alle dazu zwingen möchten, Ihren Programmservice in Anspruch zu nehmen? Sollte dies in Ihren Verkaufsüberlegungen eingeplant sein? Und dann noch der Preis des Programmservice von 60,- DM! Ich finde diesen Preis überzogen, da ja eine 3“-Disc den gleichen Grundpreis hat wie eine Kassette, etwa 5,- DM. Außerdem werden alle Programme von den Autoren auf Datenträger geliefert, es sind also keine großen Kopierarbeiten nötig. Wo ist denn da der Preis von 60,- DM gerechtfertigt? Nun zu meinem speziellen Problem: In der Spezial 1/88 auf Seite 22 steht das Programm X-COPY. Dieses sollte einmal bei mir zum Leben erweckt werden. Sie beziehen sich zwar auf das X-Modul von Vortex, geben jedoch keine Versionsnummer an. Ich besitze einen CPC 6128 mit Vortex VDOS 2.0 und meine Frage lautet: Läuft es damit oder nicht? Ansonsten, wenn der Druck verbessert wird, finde ich Ihre Zeitschrift nicht schlecht.

Wilhelm Beese,
7218 Trossingen

Wir freuen uns zwar nicht sonderlich, daß Sie über uns ein wenig Dampf ablassen müssen, müssen Ih-

nen aber teilweise recht geben und geloben auch Besserung. Dies betrifft insbesondere den Listing-Druck. Wenn Sie unser Heft regelmäßig lesen, dann haben Sie sicher gemerkt, daß es verschiedene Experimente gab, bis wir beim jetzigen Kursivdruck angelangt waren. Dabei ist die Schrägschrift eigentlich unwichtig, es war jedoch die einzige Möglichkeit, den korrekten Zeichensatz mit lesbaren Buchstaben abzudrucken. Wie Sie bemerkt haben, war der vorhergehende Druck zwar richtig, die Hersteller des Druckers sorgten aber mit einer eigenwilligen Gestaltung des Zeichensatzes (bei B,8 und &) für Mißverständnisse. Mit der jetzt gültigen Form – Kursivschrift mit besserem Zeichensatz, Checksummer und besseres Papier – sind wir eigentlich ganz zufrieden. Wir freuen uns trotzdem über Lesermeinungen.

Mit unseren zahlreichen Versuchen ist auch das Argument beseitigt, daß wir nur den Umsatz der Softbox steigern wollen. Es gab auch in der Vergangenheit viele Leser, die – entgegen Ihrer Annahme – die Programme abtippen konnten. Der von Ihnen bemängelte Preis von 60,- DM trifft nur auf die Sonderhefte zu, die ja immerhin „randvoll“ mit Programmen geliefert werden. Bei regulären Heften kostet die Softbox 30,- DM. Rechnen Sie jetzt bitte einmal nach: Statt der üblichen 30 Seiten Listings für DM 30,- erhalten Sie 100 Seiten für 60 DM, das sind 31 Programme, wobei allerdings für uns Qualität über Masse geht. Dabei machen wir bewußt keinen Unterschied zwischen Kassetten und Disketten. Wie schon

mehrfach erklärt, kosten uns Kassetten aufgrund der Kopierauflage mehr als eine Diskette und wir möchten diesen Service nur noch als Ausnahme beibehalten. Damit, daß unsere Programmautoren Datenträger einschicken sollen, hat dies nichts zu tun. Diese werden selbstverständlich umgehend zurückgesandt und senken nicht etwa die Entstehungskosten der Softbox. Für das Programm X-Copy gibt es jedoch tröstliche Nachrichten: Es läuft mit ihrer Konfiguration, ist allerdings auch Bestandteil eines Artikels, der in der regulären Ausgabe mittlerweile erschienen ist.

Auch wenn wir Ihnen in einigen Punkten widersprochen haben, möchten wir Sie bitten, weiterhin kritisch zu bleiben. Wie Sie sehen, reagieren wir – diese Seiten heißen nicht ohne Grund „Dialog“ – und wir möchten, daß die CPC-Welt Ihre Zeitschrift ist und bleibt.

LISTING ALS BARCODE

Die Firma Bühler bietet zur Zeit einen Barcodeleser zum niedrigsten Preis an. Außerdem läßt sich so etwas preiswert selbst bauen. Wäre die CPC-Welt bereit, die Listings mit dem Barcode zu drucken und weiter den Disketten/Kassettenservice bereitzuhalten? Zusammengekommen erspart dies jedem Leser das Abtippen.

Name unbekannt

Unser Softbox-Service bleibt auf jeden Fall. Wem das Abtippen zu lästig ist, der hat also bereits eine Möglichkeit, sich das zu ersparen. Die Idee mit dem Barcodele-

ser ist sicherlich gut. Wir glauben jedoch nicht, daß sich die Technik kurzfristig durchsetzen wird. Bedenken Sie: Abtippen kann jeder, aber nicht jeder CPC-Besitzer wird über einen Barcodeleser verfügen, auch wenn dieser noch so billig ist.

HISOFT IST KEIN TURBO-PASCAL

Ich habe mir aus dem Schneider aktiv Spezial 1/87 das Statik-Programm abgeschrieben und es läuft nicht. Ich habe das Programm nun einem Programmierlehrer gezeigt und der hat gesagt, an dem Listing stimme etwas nicht. Könnten Sie mir eine Korrektur schicken?

Reiner Preys,
7311 Ohmden

Ihr Lehrer hat recht. Aus der Sicht eines Turbo-Pascal - Programmierers stimmt etwas nicht mit dem Listing. Es handelt sich um HiSoft-Pascal. Obwohl jeder meint, Pascal unterliege einer festen Normung, gibt es leider doch Unterschiede.

CPC-MONITOR AM SCHNEIDER PC

Ich beabsichtige, mir einen Schneider PC zuzulegen. Ich bin zwar noch nicht sicher, ob es ein 1512- oder ein 1640-Modell sein wird, möchte mich aber vorab davon überzeugen, ob es möglich sein wird, meinen Farbmonitor vom CPC 464 auch am zukünftigen PC zu betreiben. In diesem Zusammenhang hätte ich gerne die Pinbelegung der Monitoranschlüsse beider Rechner, um ein passendes Kabel

zu basteln. Eine letzte Frage am Rande: Kann ich einen Star NL10 mit dem Centronics-Interface für den CPC 464 auch am PC benutzen?

Chris Arnu,
6464 Altenhasslau

Vorweg muß an das Netzteil erinnert werden, welches beide Rechner mit Strom versorgt. Volt- und Wattzahl sind nicht identisch, deshalb könnte der CPC-Monitor nur als Zweitbildschirm dienen. (Strom aus dem Originalmonitor, Bild über CPC-Gerät). Gegebenenfalls müssen die RGB-Signale und das Sync-Signal noch invertiert werden, dies haben wir noch nicht geprüft. Möglicherweise ist ein separates Netzteil aufzutreiben, das den PC mit Strom versorgen kann. Bedauerlicherweise unterliegen die Schaltpläne (und die darin beschriebenen Pinbelegungen) dem Copyright, wir dürfen diese nicht verbreiten. Sie können jedoch bei Ihrem Fachhändler Einsicht in die Unterlagen nehmen. Informationen über Maskentyp, „pitch“ und so weiter liegen uns nicht vor. Es existieren jedoch verschiedene Monitortypen, auch hier wäre der Händler die beste Adresse für Rückfragen. Aber Ehre, wem Ehre gebührt: In der Zeitschrift c't erschien vor einiger Zeit ein Artikel, in dem die von Ihnen gewünschte Konfiguration behandelt wurde.

Der Star NL10 verfügt über einen Epson- und IBM-Modus und kann deshalb am PC eingesetzt werden. Eventuell muß der Druckertreiber einiger Programme angepaßt werden, um deutsche Umlaute zu erreichen. Aus dem gleichen Grund kann es passieren, daß die MS-DOS Hard-

copy (PrtSc) kleine Fehler produziert.

DISKETTEN-BIBLIOTHEK: EIN GUTES PROGRAMM MIT EIN PAAR SCHWIERIGKEITEN

Läuft das Programm „Diskettenbibliothek“ aus 9/87 nur mit DISBOS oder auch anders, und wie kann die erstellte Liste gespeichert oder geladen werden? Wie kann ich nach Ausdruck der Liste bzw. der Seite den zusätzlichen großen Seitenvorschub vermeiden?

W. Schubert,
6604 Brebach

Das Programm „Diskettenbibliothek“ läuft nur ohne die Speichererweiterung richtig. Die Vortex-Karte muß also mit dem Befehl /DISBOS ausgeschaltet werden. Die erstellte Liste kann über das Diskettenmenü abgespeichert und wieder eingelesen werden; allerdings werden nur die Filenamen abgespeichert, die Sie zuvor markiert haben. Bei der Eingabe des Namens dürfen nur acht Zeichen benutzt werden, also kein Punkt und keine Extension. Diese Erweiterung fügt das Programm automatisch hinzu (dsc). Um einen Seitenvorschub zu vermeiden, entfernen Sie bitte in Zeile 30 das Steuerzeichen CHR\$(12).

IMPROPER ARGUMENT BEI "SENSO"

Seit kurzem arbeite ich mit einem CPC 6128 und habe versucht, das Listing „Senso“ abzutippen. Leider meldet der Computer IMPROPER ARGUMENT IN 140. Da ich noch zu

wenig Erfahrung habe, bitte ich Sie, mir mitzuteilen, wo der Fehler liegt. Ich habe die Zeile durchgeprüft und kann keinen Fehler finden.

Ewald App,
7177 Wellendingen

Auch hier gilt das gleiche wie im Brief des Lesers K. Weiland. Der Fehler muß nicht in Zeile 140 stecken, es ist sogar wahrscheinlicher, daß irgendeine Datazeile falsch ist.

BUDGET-MANAGER AUF KASSETTE

Ich habe das Heft Schneider aktiv Spezial 1/88 gekauft. Ich besitze einen CPC 464 ohne Diskettenlaufwerk, möchte aber trotzdem das Programm nutzen. Bitte geben Sie mir Hinweise zur Erstellung einer Kassettenversion.

Wolfgang Schmal,
5810 Witten 1

Der Budget Manager wurde speziell für den Diskettenbetrieb ausgelegt. Viele Routinen, die die Qualität des Programmes ausmachen, wären nicht mehr lauffähig. Ganz abgesehen davon, daß es sicherlich zuviel Mühe macht, ein derart komplexes Programm auf Diskettenzugriffe zu durchforsten. Wir empfehlen Ihnen deshalb das Programm „Haushaltsbuch“ aus dem Sonderheft 2/87.

EILT – DRINGENDE BERICHTIGUNG

In Ihrer Ausgabe 12/87 unterlief Ihnen ein Fehler beim Artikel „PC 1512 auch für sechs- bis 14jährige“. Die Diskette mit allen zehn Program-

men kostet 20 DM; nicht wie Sie schreiben, 20,- DM je Programm. Das wären ja 200,- DM, und wir wollen doch gute Software für Kinder so preiswert wie möglich machen.

Firma Freier,
Engelbert-Kraus-Str. 25,
8709 Rimparr

CP/M PROGRAMM-AUSTAUSCH ZWISCHEN COMMODORE UND CPC

Wie kann ich ein Programm, das über Maschinensprache geladen wird, von Kasette auf Diskette kopieren? Gibt es eine Möglichkeit, die Startadresse und die Länge des Programmes im Speicher festzustellen? Es handelt sich um die Programme „Nonterraqueous“ und „Soul Of A Robot“, beide von Mastertronic. Ich besitze einen CPC 6128.

Frage 2:

Ich versuche, mit einem Freund, der einen Commodore besitzt, CP/M Programme (Public Domain) auszutauschen. Dies funktioniert jedoch nicht. Ich besitze die Vortex Floppy F!X (5.25 Zoll) und wollte mir nicht erst für 200 DM ein Programm kaufen, um tauschen zu können. Gibt es eine Möglichkeit, ohne Hardwareeingriffe die Floppy auf 40 Spuren umzuschalten?

Frage 3:

Wie kann ich unter Maschinensprache Interrupts einbinden?

Marcus Rehbein,
2000 Hamburg

Das Übertragen der Maschinensprache von Kasette auf Disk ist im Regelfall sehr einfach; die Übertragung von kommerziell vertriebenen Programmen leider nicht.

Das liegt vor allem daran, daß die meisten Kassettenprogramme eigene Lader enthalten und die Hauptteile der Programme nicht mit Firmware-Routinen eingelesen werden. Wir kennen zwar die angeführten Programme nicht, aber Sie werden sich von dieser Tatsache überzeugen können, wenn Sie das Programm „Header“ aus Schneider aktiv 8/87 einsetzen. Dieses Programm gibt Ihnen die gewünschte Information von ungeschützten Files.

Zu 2:

Soweit uns bekannt ist, kommen bei der Firma Vortex die BASF-Laufwerke zum Einsatz. Bei diesen Laufwerken wissen wir leider nicht, ob Sie durch einen Hardware-Eingriff auf 40 Tracks umgestellt werden können. Aber dies war keine Frage von Ihnen. Beim C128 von Commodore kommt CP/M Plus, Version 3.0 zum Einsatz. Das Commodore-Laufwerk 1571 ermöglicht es, z.B. unter CP/M 80, Disketten in den Formaten KAPYRO II, EPSON, QX10 VALDOCS und OSBORNE zu verarbeiten. Wenn Sie also Public-Domain-Programme an Ihren Bekannten weitergeben wollen, dann geht es nur, wenn Sie mit derart formatierten Disketten arbeiten. Leider haben Sie dieses Format nicht zur Verfügung, es sei denn, Sie haben z.B. die Programme PARA Plus (mit dem Vortex-Controller) oder DISC-PARA (ohne Vortex-Controller). Ihre Frage, ob Sie ohne Hardwareeingriffe die Möglichkeit haben, andere Formate zu verarbeiten, kann also mit einem eindeutigen „Ja“ beantwortet werden, aber Sie brauchen eines dieser Konvertierungs-

programme. Eine weitere Möglichkeit wäre die Übertragung über die RS-232-Schnittstelle.

Zu Frage 3:

Die Einbindung von Interrupts in Maschinenprogramme ist zwar nicht weiter schwierig, aber ein Kapitel, welches nicht mit wenigen Sätzen abgehandelt werden kann. Wir werden dies im Rahmen unserer Artikelreihe über Maschinensprache bringen. Vorab nur in Kürze folgendes: Im CPC arbeitet der Z80 im Interrupt-Modus 1, wodurch alle Unterbrechnungen einen RST7 ausführen. Eine eigene Interrupt-Routine muß deshalb ab 3B installiert werden. Siehe hierzu auch das Firmware-Handbuch des CPC.

KEINE DRUCKAUSGABE IM ERSTEN MENU EPUNKT DES FINANZGENIES

Ihr Finanzgenie weist Fehler auf. Auf Seite 77 mußte ich in der Routine Kapital (Wertpapiere) noch Datas schreiben, damit ich dieses ausdrucken konnte. Der Rücksprung ins Hauptmenü ist nicht möglich. Eine Frage habe ich noch: Könnte man auch eine Originalkassette aus Heft 2 haben oder führen Sie die nicht mehr?

Frank Michalik,
6551 Argenschwank

Wahrscheinlich liegt bei Ihnen ein Fehler in den Zeilen der Rücksprungbedingungen vor. Das Hauptmenü wird nach jeder abgearbeiteten Unteroutine korrekt ausgegeben. Allerdings ist es völlig richtig, daß zum Menüpunkt 1 keine Druckerausgabe vorgesehen war. Zu Ihrer Frage

nach der Softbox 2/87: Alle Programme der Hefte ab 10/86 sind lieferbar. Damit sparen Sie sich die Tipparbeit und können sicher sein, daß alle Programme laufen.

KARTEIKASTEN MELDETE "MEMORY FULL"

Vor kurzem habe ich Ihr Sonderheft 4/87 und auch die Diskette hierzu von Ihrem Verlag erworben. Mich interessierte besonders das Listing „Kartei“ von Rudolf Bodinger. Das Programm begeisterte mich sehr, zumal es auch sehr bedienerfreundlich angesetzt und die Druckroutine ebenfalls ausgezeichnet ist. Ich habe dieses Programm gleich für die Speicherung meiner CD-Platten eingerichtet und habe dabei eine unangenehme Überraschung erlebt. Dem Menü folgend habe ich sieben Felder eingerichtet und 200 Datensätze vorgegeben. Funktionierte sehr gut. Nun habe ich mich an die Eingabe herangemacht. Als ich dann bei Eingabe 102 (Datensätze) angekommen war und speichern wollte, streikte das Programm und links oben erschien – man staune – „Memory Full in 1400“. – Die Arbeit war leider umsonst.

Ich könnte mir vorstellen, daß neben den Informationen in Ihrem Heft auch vielleicht ein Hinweis über die Speicherkapazität ganz nützlich wäre. Oder erwarten Sie, daß man dies selbst weiß oder gar aus dem Programm lesen kann?

Eberhard Geßler
2740 Bremervörde

Zu diesem Problem eine kleine Hilfe des Autors

Rudolf Bodinger:

Generell läßt sich vom Programmierer kein Hinweis auf die Speicherkapazität eines Karteiprogrammes geben, wenn die Anzahl der Sätze, Felder und Zeichen nicht von vornherein festgelegt werden. Das von mir erstellte Programm wird durch seine umfangreichen Möglichkeiten in seinem Speicherplatz begrenzt. Da das Programm für den CPC 464 mit Rekorder geschrieben wurde, sollten Besitzer einer Floppy nach Symbol After 90 (in Zeile 4490) folgende Zeile einfügen:

4495 Openout "DUMMY"
:MEMORY HIMEN-1:
CLOSE OUT

Wenn Sie das Programm laden (nicht starten) und dann FREE ("") eintippen, erhalten Sie den verfügbaren Speicherplatz von 23382 Bytes angezeigt. Legen Sie eine Diskette in das Laufwerk ein und starten Sie mit RUN. Die Abfragen von Karteinamen sowie Anzahl und Sätze der Felder beantworten Sie nur durch Drücken der ENTER-Taste, bis das Menü erreicht ist. Dann unterbrechen Sie und lassen sich FREE ("") ausgeben. Es müssen 17817 Bytes sein. (einige mehr oder weniger können durch das Hinzufügen oder Weglassen von Kommentaren zustande kommen). 5565 Bytes sind es weniger geworden, davon gehen 4 KByte auf das Konto der Zeile 4495, die für Speichern und Laden der Daten reserviert wird, die übrigen Byte verbraucht die Sortieroutine. Die Reservierung der Felder wird noch einmal Speicherplatz beanspruchen. Das Kernproblem ist: Niemand weiß, wie groß Ihre einzelnen Datenfelder sind und wie groß

dementsprechend die Datei werden darf. Reicht der Speicher nicht für Ihre komplette Sammlung, so müssen Sie in einzelne Teildateien ausweichen, etwa von A bis K und von J bis Z.

TIPPFELER MUSS NICHT IN DER ANGE- GEBENEN ZEILE SEIN!

In Ihrer Ausgabe 10/87 brachten Sie ein Listing von einem Malprogramm — GPAINT —, welches aus mehreren Teilen besteht. Ich habe treu und brav alles abgetippt. Der Grafik-Datalader, der das Binärfile GRAFIK.BIN erzeugen soll, funktioniert nicht. Er meldet immer wieder „Type Mismatch in 700“ und ich bin jetzt ratlos. Das Listing ist korrekt abgetippt, wir haben jetzt mit zwei Personen zum xten Male überprüft.

Karlheinz Weiland,
4408 Dülmen-Buldern

Die Fehlermeldung des Interpreters weist darauf hin, daß ein Schreibfehler existiert. Dieser muß jedoch nicht in Zeile 700 sein, sondern verbirgt sich wahrscheinlich in den Datazeilen. Zeile 700 weist den Datas nämlich nur einen korrekten Zahlenwert zu und poked diese in den Speicher. Mit einer Zahl (oder mehreren) kann diese Routine jedoch nichts anfangen, sie ist falsch. Daß wir mittlerweile einen Checksummer haben, wird Ihnen wenig nutzen. Deshalb hier ein kleiner Trick, wie Sie den Fehler schnell aufspüren können: Fügen Sie den CALL &BB06 ein, so daß die Zeile lautet:

```
700 d$="@"+d$:CALL  
&BB06:POKE i,... usw.
```

Nun wird nur durch Ta-

stendruck eingelesen und gepoked. Mit ein wenig Aufmerksamkeit wissen Sie, welche Zahl gerade bearbeitet wurde, wenn der Fehler auftritt. Nach der Korrektur muß dieser Call selbstverständlich entfernt werden.

MERGE ODER CHAIN MERGE — TIPS GEGEN DEN BETRIEBS- SYSTEMFEHLER

Ich habe meinen CPC 464 noch nicht sehr lange. Können Sie einige meiner Probleme lösen?

1. Wenn ich Programme „mergen“ will, kommt sehr oft die EOF MET-Meldung (Nr.: 24), zum Teil unabhängig von der Programmlänge und trotz richtigem RENUM. Bei längeren Programmen tritt diese Fehlermeldung meines Erachtens allerdings häufiger auf. Beide Listings sind dann zerstört. Wie kann ich schon vor dem „Mischen“ dem Programm ansehen, daß der Fehler auftritt und wie kann ich die Files trotz EOF „mergen“?

2. Wann wähle ich CHAIN oder CHAIN MERGE anstatt LOAD oder MERGE? Besonders bei CHAIN werde ich aus dem Benutzerhandbuch nicht schlau.

3. Wenn sich in eine — über OPENIN zu ladende — Datei ein Direktbefehl eingeschlichen hat (Fehler Nr.: 21), wie kann ich den Direktbefehl aus der Datei löschen, ohne diese zu beschädigen, wie finde ich den Namen des Befehls heraus und wie kann ich aus einem gespeicherten Programm einen Befehl ohne Zeilennummer löschen, damit das Programm wieder geladen werden kann?

4. Kann ich einen Schalter in den Monitor oder das Verbindungskabel setzen, damit der Bildschirm ausgeschaltet, die Stromversorgung für das Keyboard aber aufrecht erhalten wird?

Rolf Carpentier,
6100 Darmstadt

Eine Menge Fragen, und deshalb auch der Reihe nach zuerst zu eins: Der CPC 464 hat einen Fehler im Betriebssystem, der für den MERGE-Ärger verantwortlich ist. Der vielversprechendste Ausweg liegt darin, das zu „mergende“ Programm als ASCII-File abzuspeichern (SAVE“Name“,A). Falls dies durch Textverarbeitungsprogramme geschieht, kann es auch manchmal zu EOF MET kommen, die Programme stehen zu diesem Zeitpunkt aber komplett im Speicher.

Zur zweiten Frage: Wenn Sie MERGE im Direktmodus eingeben, werden zwei Programme miteinander gemischt, aber nicht gestartet. CHAIN MERGE mischt ebenfalls die Programme, startet jedoch sofort danach die Ausführung. Durch die Angabe weiterer Parameter kann veranlaßt werden, daß zuvor Zeilen gelöscht werden oder die Programmausführung bei einer bestimmten Zeile beginnt. Beispiel: CHAIN MERGE “PROG1“,1000,DELETE 200—400

Dieser Befehl bewirkt, daß das Programm „PROG1“ zugemischt wird, der Programmstart in der Zeile 1000 erfolgt und die Zeilen 200 bis 400 gelöscht werden. Der Nutzen dieses Befehls liegt auf der Hand. Bei großen Programmen können bestimmte Routinen auf der Diskette bleiben und

werden nur bei Bedarf eingeladen und ausgeführt.

Bei der dritten Frage verstehen wir das Problem nicht exakt. Selbstverständlich kommt es zu einer Fehlermeldung, wenn Sie versuchen, ein Text-ASCII-File mit RUN“Name des Files“ zu starten. Gleiches kann auch passieren, wenn Sie ein derartiges File laden wollen. Mit dem Befehl OPENIN und dem Einlesen von Strings konnten wir den Fehler “DIRECT COMMAND FOUND“ jedoch nicht erzeugen. Bestimmte abgespeicherte Daten erzeugen jedoch diese Fehlermeldung, es handelt sich dabei allerdings nicht um lauffähige Basic-Programme, sondern z.B. um einen Pascal-Code. Beim Locomotive-Basic — so der zweite Teil dieser Frage — sind Zeilennummern unbedingt notwendig. Es ist jedoch denkbar, daß ein Poke diese Zahlen unsichtbar machte. Vielleicht kommen daher Ihre Probleme, eine Zeile ohne Nummer löschen zu müssen. Laden Sie in diesem Fall erst einmal mit LOAD, um den Poke nicht zu aktivieren. Es ist allerdings auch denkbar, daß Sie versuchen, in einem Textfile zu löschen, oder, daß es sich um das Label-Basic eines anderen Rechners (etwa den PC 1512) handelt.

Zu den letzten Fragen: Sie können unbesorgt den sechspoligen Monitorstecker aus dem CPC ziehen. Hierdurch wird nur das Bild ausgeschaltet, die Stromversorgung bleibt trotzdem erhalten. Sie können natürlich auch durch Lötarbeiten einen Schalter einfügen. Da Sie jedoch noch nicht lange Besitzer Ihres Gerätes sind, möchten wir Ih-

nen davon abraten. Sie verlieren hierdurch den Garantieanspruch, selbst wenn Sie korrekt arbeiten.

BEZUGSQUELLE DES LAUFWERKBAUSATZES

Zu Ihrem Artikel „Laufwerke am CPC – Alle Formate passen“ habe ich folgende Fragen an Sie:

Kann man den Bausatz für eine 5.25-Zoll-Floppy bei Ihnen erwerben und wie hoch wäre der Preis dafür? Falls Sie jedoch den Bausatz nicht zum Verkauf anbieten sollten, wäre ich Ihnen sehr dankbar, wenn Sie mir die Anschriften der Firmen mitteilen würden, die die Einzelteile liefern. Weiterhin wäre ich Ihnen sehr verbunden, wenn Sie mir die ungefähren Preise für die Einzelteile mitteilen könnten. Ich hoffe, daß Sie mir auf meine Fragen eine Antwort geben können, da ich das Laufwerk dringend für mein Maschinenbaustudium brauche.

André Hartmann,
4630 Bochum

Es mag vielleicht mittlerweile üblich sein, daß man nur noch Produkte bespricht, an deren Verkauf man interessiert ist, bei uns war und ist dies allerdings nicht der Fall, unsere Tests sind objektiv. Deshalb können wir Sie nur auf das Bezugsquellenverzeichnis (auf Seite 83 der gleichen Ausgabe) verweisen.

VERBESSERUNGEN FÜR DEN "EWIGEN KALENDER"

Zunächst herzlichen Dank für die prompte Be-

antwortung meiner Anfrage bezüglich nicht funktionierender Programme. Der „Ewige Kalender“ läuft mittlerweile einandfrei und ich hatte tatsächlich vor lauter Sternen in Zeile 500 den Vorsatz `z+a(i)` übersehen. Zu diesem Programm hat der Ausdruck von Fastnacht und Valentinstag bei der Jahreszahl 2001 nicht richtig funktioniert. Hier die Änderung der Zeile 530:

```
530 IF j<=200 THEN
t=t+1
```

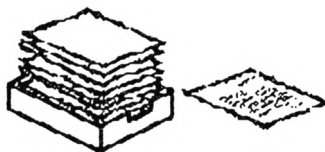
Es wirkt außerdem übersichtlicher, wenn der Zeilenvorschub in Zeile 290 auf `CHR$(10)` gesetzt wird, der Kalender füllt dann eine DIN A4-Seite hoch. Um den bei meinem Drucker mit DIP-Schalter fest eingestellten Zeilenvorschub zu überlisten, habe ich noch die Zeile 293 eingefügt (Abschaltung dieses Vorschubes per Software):

```
293 PRINT#8,CHR$(27);
„0“
```

Ich werde von den Redaktionen anderer Schneider-Magazine mit Beantwortung von Fragen nicht gerade verwöhnt; ich glaube daher, daß Sie auf dem richtigen Weg sind, wenn Sie diese Sparte der Leserbetreuung weiter so prompt ausüben.

Walter Halbmaier,
6806 Viernheim

Vielen Dank für das Lob und die Verbesserungsvorschläge. Da bleibt uns nur noch die Information für alle User südlich des Weißwurstkanals: Fastnacht ist der „Fasching“ der Preußen.



Bezugsquellen:

Was gibt es wo?

Von „Schneider Welt“ getestet.

Wir wollen hier keine Gratiswerbung – es finden sich auch die Vertreter jener Soft- und Hardware, die bei uns nicht auf Gegenliebe stieß – machen, aber ein Testbericht ist völlig sinnlos, wenn wir nicht auch mitteilen, wo das Besprochene zu beziehen ist. Leider können nicht alle Händler berücksichtigt werden, die die Ware im Sortiment führen. Dies soll nur ein letzter Hinweis für denjenigen Interessenten sein, der bei seinem Fachhändler auf leere Regale stieß.

Dirman (11/87)
Tracer(11/87)

D. Gunkel Software
Jülicherstr. 312
5100 Aachen

TAS ? Datenbank (1/88)

Henschke Datentechnik
Aidlinger Weg 6
7034 Gaertingen

Street Hawk (11/87)
Mag Max (11/87)

Waldeck Software
Tulpenstr. 30
2870 Delmenhorst

Musterprogramme in Turbo-Pascal
Mehr Erfolg mit Schneider CPC ...
Aktuelles IC-Datenbuch

Interest-Verlag
Postfach 1150
8901 Kissing

James Bond – Der Hauch des Todes (1/88)

Rushware GmbH
An der Gumpgesbrücke 24
4044 Kaarst 2

Multiface II

Weeske Computer Elektronik
Potsdamer Ring 10
7150 Backnang
und

Romantic Robot
(Goldscheider)
Ben-Gurion-Ring 80
6000 Frankfurt 56

Schneider PC 1640 (11/87)
BTX Modul zum CPC 464 (2/88)
Drucker LQ 3500 (1/88)

Im Fachhandel
Musterprogramme

Cholo (11/87)
Indoor Sports (11/87)
Hybrid (11/87)
Vermeer (11/87)
Tracker PC 1512 (12/87)
High Frontier (12/87)
Balloon (1/88)
Asphalt (12/88)
Fußball Manager (1/88)
Reisende im Wind 2 (1/88)
Sidewalk (1/88)
Renegade (1/88)
Cassix 1 (1/88)
Renegade (2/88)
Guild Of Thieves (1/88)
Trantor (1/88)

ariolasoft GmbH
Postfach 1350
4830 Gütersloh

Firetrap (2/88)
Trantor (2/88)

Activision Deutschland GmbH
Postfach 760680
2000 Hamburg 76

C-Compiler (2/88)

Arnor (Deutschland)
Hans-Henny-Jahn-Weg 21
2000 Hamburg 76

Vereinsverwaltung (11/87)

„Byte me“-Computersysteme
Wilhelmstr. 7
5240 Betzdorf

Faulheit lohnt sich: Die Säulengrafik wird automatisiert

In mageren zehn Zeilen haben wir in der letzten Ausgabe ein Säulendiagramm vorgestellt. Es diente eigentlich nur dazu, den Aufbau einer solchen Grafik zu demonstrieren, und das Listing krankte noch an vielen Kleinigkeiten. Mit einer Eingaberoutine und einer automatischen Platzierung wollen wir das Utility heute vervollständigen. Für den, der sich nur auf das Abtippen beschränkt, wurde das begleitende Listing mit dem CPC-Welt Checksummer versehen.

Die Berechnung des bisherigen Diagrammes war recht einfach. Sowohl die Anzahl der Daten als auch die maximale Höhe standen bereits fest. Die Werte waren ja als Programmbestandteil in Datenzeilen eingegeben. In der Praxis ist diese Vorgehensweise jedoch unmöglich. Man stelle sich vor, es müßte jedesmal das Listing aufgerufen, die Datenzeilen und natürlich auch sämtliche FOR-NEXT-Schleifen geändert werden. Der Computer würde die Arbeit nicht erleichtern, sondern nur neue schaffen. Aber selbst wer sich diese Mühe einmal machen will, wird feststellen, daß der optische Eindruck recht unbefriedigend ist. Wir haben ja, um das Diagramm in der ganzen Breite füllen zu können, einfach vorausgesetzt, daß es sich um zehn Werte handelt. Auf drei reduziert, würden sich diese Balken alle in der linken Ecke drängeln und mehr als zehn wären nicht auf den Bildschirm zu bringen. Gehen wir also ein-

mal davon aus, daß wir alle Werte in gleichmäßiger Breite um die Grundlinie verteilen wollen, egal, ob es sich um drei oder 30 handelt. Diese Aufteilung soll aber der Rechner übernehmen, denn die Rechenarbeit macht er schneller und gründlicher.

Damit sind wir auch schon bei der Anzahl der Werte. Hierfür muß im Speicher erst einmal Platz geschaffen werden, indem wir die DIM-Anweisung (Zeile 30) anwenden. Bei einem Programm dieser Länge ist der Speicherplatz für andere Aufgaben nicht so wichtig, nach oben sind also keine Grenzen gesetzt. Für mehr als 100 Werte bietet allerdings der Bildschirm bei der späteren Programmführung keinen Platz. Im Gegenteil: Bei so großen Mengen kommt es leicht zu Rechenungenauigkeiten, die zu häßlichen Lücken führen. Wollen wir eine vernünftige Darstellung, müssen wir uns bei diesem Listing auf etwa 50 Zahlen bei der Eingabe beschränken. Mit

```

10 '*** Balkendiagramm ***           <1DAC>
20 '***   Eingabe   ***             <1DC7>
30 MODE 2:WINDOW#1,1,80,1,1:DIM we
rt(100):PRINT                        <2374>
40 FOR i=1 TO 100                    <0FA0>
50 PRINT#1,"Ende mit Return (Enter
)   Wert";i;": ";:INPUT#1,"",wer
t$                                     <454F>
60 IF wert$=""THEN GOTO 90 ELSE we
rt(i)=VAL(wert$)                     <302C>
70 IF wert(i)>=grwert THEN grwert=
wert(i)                               <351C>
80 PRINT TAB(60);:PRINT USING"###
###.##";wert(i):NEXT i              <30E5>
90 CLS#1:anz=i-1:PRINT#1,"Anzahl d
er Werte:";anz,                     <36E4>
100 PRINT#1,"Taste = Grafik":CALL
&BB06                                 <1E8F>
110 '*** Berechnung ***             <1971>
120 h%=350/grwert;br%=550/(2*anz) <2CF6>
130 start%=2*(550/(2*anz)):beg%=50
+br%/2                                <3288>
140 '*** Grafik ***                 <1575>
150 CLS:ORIGIN 50,30:DRAWR 550,0    <14F4>
160 ORIGIN 50,30:DRAWR 0,350       <12A2>
170 FOR i=1 TO anz                  <13EA>
180 x%=(i-1)*start%+beg%:y%=h%*we
rt(i)                                 <3AD9>
190 FOR j%=0 TO br%:ORIGIN x%+j%,3
0:DRAWR 0,y%:NEXT j%               <2E3E>
200 NEXT i                          <0A01>
210 CALL &BB06                      <09F0>
220 '*** Hardcopy einfüegen ***    <214D>
230 RUN                             <06F6>

```

DIM (100) wird ein wenig hochgestapelt. In der gleichen Zeile sollte ein Window definiert werden. Es dient ebenfalls nur der Optik, denn die Eingabe wird dadurch ansprechender gestaltet. Der letzte Befehl, die PRINT-Anweisung, hat eine besondere Bedeutung, dazu jedoch später mehr.

EINGABEROUTINE MIT ABBRUCHBEDINGUNG: ZAHLEN ALS TEXTSTRING EINGEBEN

Entsprechend dem dimensionierten Speicherplatz reicht die FOR-

NEXT-Schleife zur Eingabe bis 100 (Zeile 40). Der nächste Schritt ist die Eingabe im festgelegten Fenster Nummer 1. Hierfür würde die Input-Abfrage genügen, doch um etwas Übersicht zu bekommen, darf dabei auch angegeben werden, um welchen Wert es sich handelt. Am besten benutzen wir dazu die Schleifenvariable i, die hinter dem Text ausgegeben wird. In diesem Eingabetext ist auch die Abbruchbedingung als Hilfe für den Benutzer genannt.

Stört Sie auch oft das Fragezeichen nach einem INPUT-Befehl? Wenn ja, sehen Sie hier eine Mög-

lichkeit, wie es unterdrückt werden kann. Achten Sie auf die Kommas und den kleinen Trick, durch den ein gar nicht vorhandener Text in Anführungsstriche gesetzt wird. Ganz nebenbei wird Ihnen auch auffallen, daß wir gar nicht nach einer Zahl fragen, sondern nach einem Text, einem String, den das /-Zeichen kennzeichnet. Warum dies?

Auf diese Weise kann in Zeile 60 abgefragt werden, ob eine Leereingabe gemacht wurde (IF wert S/= " "). Wird diese Bedingung erfüllt, dann verläßt das Programm die Eingabeschleife. Andernfalls müßten wir ja immer 100 Werte eingeben. Überlegen Sie doch einmal, wie eine Zahlenkette auf eine Abbruchbedingung geprüft werden kann. Es ist gar nicht so einfach. Die bequemste Lösung wäre, eine bestimmte Zahl zu vereinbaren, etwa 1E11 ("1 hoch 11"). Als Wert kommt diese Zahl wahrscheinlich nicht vor; was aber, wenn ein anderer das Utility zum ersten Mal benutzt? Sollen wir deshalb gleich eine Bedie-

nungsanweisung schreiben? Die Werte zuerst einmal als Text einzugeben, ist also schon richtig, erfordert aber auch, daß wir sie in der zweiten Hälfte der Zeile 60 in Zahlen umwandeln, nachdem die Abbruchbedingung geprüft wurde. Mit der VAL-Anweisung ist dies leicht zu machen.

Erinnern Sie sich noch an das erste Listing zum Balkendiagramm? Den Maximalwert haben wir einfach bei 30 festgelegt und danach die Höhe aller Balken berechnet. Durch die Eingaberoutine besitzt dieser Wert allerdings keine Berechtigung mehr, der größte Wert muß erst einmal ermittelt werden, am einfachsten, indem wir ihn mit dem Maximalwert vergleichen. Er wird in Zeile 70 an die Variable grwert (für größter Wert) übergeben und später kann damit weiter gerechnet werden. Sicherlich hat der CPC dafür die Befehle MIN und MAX, aber hier wird wenigstens vorgeführt, auf welcher Logik sie beruhen.

Kleine Nebensächlichkeiten lassen das Programm übersichtlicher

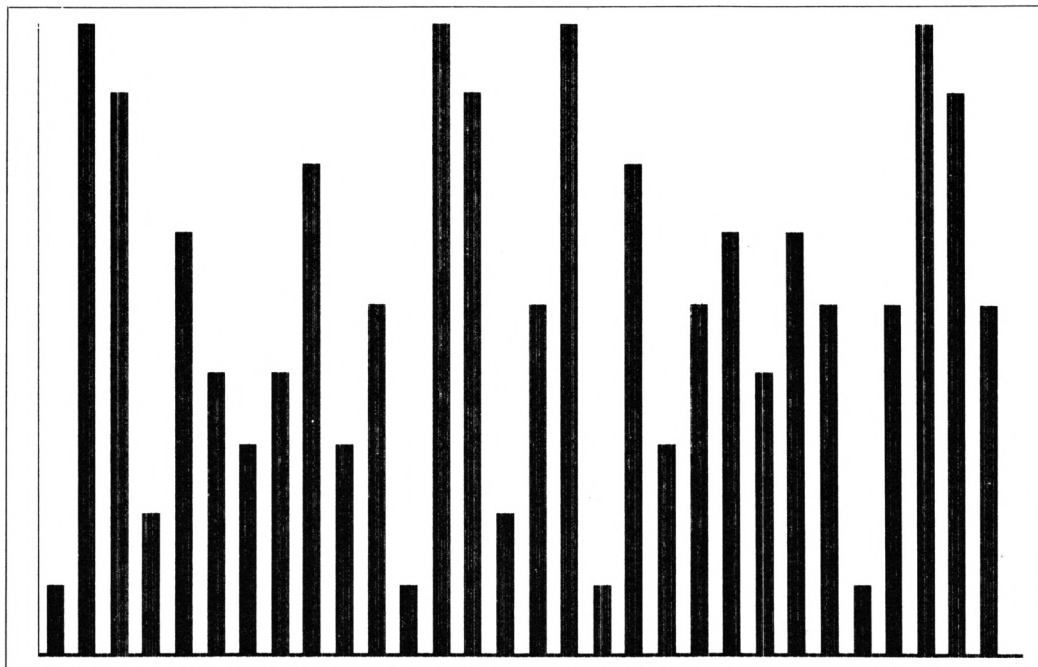
werden. So lassen wir uns schon bei der Eingabe die Zahlen auflisten. Und nun wird auch der PRINT-Befehl aus Zeile 30 wichtig. Fehlt er, dann wird die erste Zahl in der ersten Zeile des Bildschirms gedruckt und gleich darauf durch das Window 1 überschrieben. Entweder müssen wir also ein zweites Fenster definieren, welches erst in Zeile zwei beginnt, oder wir sparen uns die Mühe und wenden den kleinen Trick an. Auch die Bequemlichkeit muß berücksichtigt werden. Mit dem NEXT in Zeile 200 ist dann die Schleife zur Werteingabe beendet. Die Anzahl der Werte ist leicht zu ermitteln, wir müssen nur darauf achten, daß die Variable i der Eingabezahl ja immer um 1 voraus ist, also ist $anz=i-1$.

VORBEREITENDE BE- RECHNUNGEN ZUR GRAFIKAUSGABE

Stellen wir noch einmal kurz klar, welche Werte wir für unsere Balkendarstellung benötigen: Ganz wichtig, denn darum geht

es ja, ist die Höhe der Säulen. Absolutes Limit dürfte hierbei der Maximalwert sein. Aber wir wollten auch eine gefällige Optik haben und dazu sämtliche Säulen „gerecht“ auf der Grundlinie verteilen. Also benötigen wir auch die Breite jedes Wertes und darüber hinaus die Breite des Zwischenraumes. Dabei muß mit Variablen gearbeitet werden, denn sämtliche Werte ändern sich bei jedem Programmdurchlauf. Beginnen wir mit der Höhe:

Der verfügbare vertikale Platz ab der Grundlinie steht ja schon fest, es sind 350 Bildpunkte, 50 haben wir als unteren Rand „verschwendet“. Genau so hoch darf dann auch der größte Wert dargestellt werden, alle anderen werden daran gemessen. Eine Größeneinheit ergibt sich dann aus der Formel $h=350/grwert$. Im Verhältnis dazu muß jede andere Zahl nur mit h multipliziert werden. Wer es schon vergessen hatte: In der Schule nannte man dies „Dreisatz-Rechnung“. Probieren Sie es einmal aus. Wenn der Maximalwert 50 ist und damit 350 Punkte Höhe hat, dann ist, nach $h=350/grwert$, $h=7$. Der Wert 25 umfaßt mit $25*7$ genau 175 Bildpunkte und ist nur halb so hoch. So einfach sich das hier auch liest, wir haben damit immerhin eine Automatik geschaffen, die immer das richtige Verhältnis berechnet. Dabei ist es unwichtig, ob die Zahlen nun im Tausenderbereich oder unter Null im Komabereich liegen. Lediglich allzu große Abweichungen innerhalb der Wertgruppe lassen sich grafisch nicht mehr darstellen. Aber so ist das nun mal. Den Unterschied



zwischen einem Milliarden und einem Normalverdiener kann keiner in ein Diagramm zwingen.

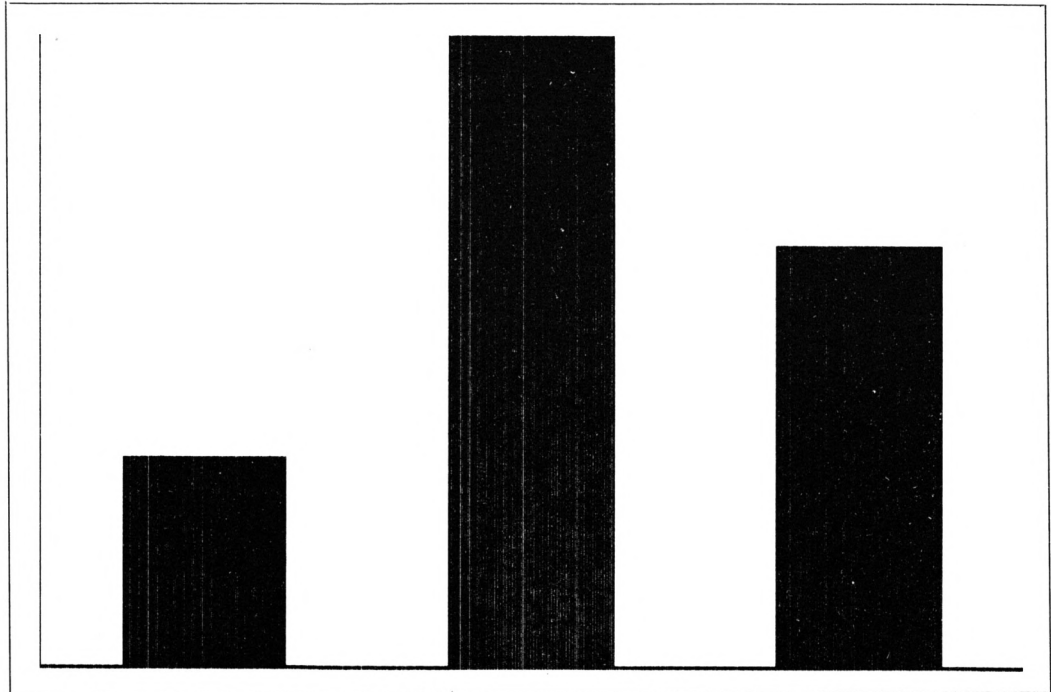
ABSTAND ZWISCHEN DEN SÄULEN VARIABLEN BERECHNEN

Zur Berechnung der Säulenbreite ist die Anzahl der Werte wichtig. Wir haben diesen Wert in Zeile 90 ja nicht nur ermittelt, damit er kurz auf dem Bildschirm ausgegeben werden kann, sondern er wird jetzt benötigt. Darüber hinaus müssen wir uns aber noch überlegen, wie groß der Abstand zwischen zwei Werten sein darf oder soll. Im Grunde genommen bleibt es jedem selbst überlassen, normalerweise ist der Zwischenraum jedoch immer genauso groß wie der Balken selbst. So kann es dann auch einmal recht eng hergehen und es ist doch noch zu sehen, daß es sich um einzelne Balken handelt. Für die Ermittlung des Zahlenwertes können wir also so tun, als hätten wir einen Platzbedarf; jeder davon benötigt ja noch einmal den gleichen Platz als Abstand. Die Berechnung lautet demnach

$$br=550/(2*anz),$$

wobei 550 Bildpunkte der größtmögliche Platz ist, den wir nach Einzeichnen des Koordinatenkreuzes noch zu vergeben haben.

Bleibt nur noch die Frage, an welcher Stelle der Grundlinie ein Balken beginnt und wo der absolute Anfang dieser Grundlinie liegt. Ersteres wird unter dem Variablennamen `start` errechnet. Wenn jedem Wert der gleich große Freiraum folgt, dann ist der Standpunkt jedes Balkens bei



`br*2` zu setzen. Im Listing haben wir diese Formel etwas komplizierter mit `start=2*(550/(2*anz))` dargestellt; warum, lesen Sie unten. Der absolute Startpunkt auf der Achse läge eigentlich bei 50, denn hier beginnt die Linie. Dies ist zwar auch machbar, zählen wir jedoch noch einmal die Hälfte eines Zwischenabstandes (`beg=50+br/2`) hinzu, dann wird das Diagramm etwas aufgelokkter.

INTEGERZAHLEN BRINGEN TEMPO

Das Listing liest sich etwas anders, als der Text hierzu. Das Programm sollte etwas beschleunigt werden, denn es gibt immerhin viel zu zeichnen. Am besten, wir ersparen dem CPC überflüssige Rechenarbeit. Da die Bildschirmkoordinaten ja nur ganzzahlig sind, eine Nachkommastelle also nicht berücksichtigt werden kann, bieten wir zur Weiterverarbeitung auch nur Ganzzahlen (sogenannte Integerzahlen) an. Den Variablen `h`, `br`, `start`

und `beg` wird hierzu das %-Zeichen angehängt. Aber wir dürfen nicht den Fehler machen und uns zu früh auf Integerzahlen festlegen, denn dadurch kommt es zu großen Ungenauigkeiten durch Auf- und Abrundungen. Aus diesem Grund sollte die Variable `start%` auch nicht durch `2*br%` berechnet werden, sondern durch die komplette Formel. Nur das Endergebnis wird gerundet.

Rein rechnerisch ist damit alles getan und es kann ans Zeichnen gehen. Dank der Vorarbeit gibt es keine Probleme mehr. Die Schleife verläuft jetzt bis `anz`, der Anzahl aller Werte. Es wird dann der Startpunkt des Balkens festgelegt, eine zweite Schleife zeichnet einzelne Linien und zwar genau `br` Stück. Nach getaner Arbeit wartet der CPC auf einen Tastendruck und beginnt dann mit der Programmausführung von vorn.

An dieser Stelle des abgedruckten Listings wäre der beste Platz eine Hardcopyroutine unterzubringen. Die CPC-Welt hat mehrere dieser Utilities veröffentlicht, so daß es ein leichtes ist, die passende herauszusuchen. Aber es gibt noch mehr Möglichkeiten zur Verbesserung. Bislang

ANREGUNGEN ZUM WEITERMACHEN

fehlt ja völlig die Beschriftung des Diagramms, etwa eine Skalierung oder eine Bildunterschrift. Weiterhin lassen sich auch die Werte numerieren, sofern nicht übermäßig viele Zahlen dargestellt werden sollen. Selbstverständlich kann auch mit der Farbgebung experimentiert werden.

Worauf es ankam, war, zu demonstrieren, wie kurz, aber komfortabel ein solches Utility sein kann. In der gegenwärtigen Form läßt es sich in fast jedes Programm einbauen und vermittelt einen anschaulichen Überblick zu Zahlenwerten.

Kleine Hilfen für den Programmierer

Neulinge in der Maschinensprache sollten nicht gleich zurückschrecken, dies ist kein Assembler-Kurs. Aber gerade mit Hilfe der Sprungleiste erscheint die Programmierung viel, viel einfacher. Also dann, rein ins Vergnügen ...

Am einfachsten ist die Textausgabe mit der Routine TXT OUT ACTION an der Adresse #BBSA. Man übergibt im Z80-Register A den ASCII-Code des auszugebenden Zeichens und ruft die Routine mit einem CALL auf. Das Zeichen wird dann in der aktuellen Schreibfarbe an der aktuellen Cursorposition ausgegeben, der Cursor wird auf die nächste Stelle gesetzt. Die Z80-Register werden nicht verändert, Steuerzeichen (Zeichen mit Code kleiner als 32) werden nicht dargestellt, sondern ausgeführt.

Um Steuerzeichen darzustellen, wählt man die nächste Routine, TXT WR CHAR (#BBSA). Mit diesem Unterprogramm, das nebenbei auch die Register AF, HL, DE und BC verändert, werden alle Zeichen auf dem Bildschirm ausgegeben, also auch die Steuerzeichen.

Um nun den Cursor zuvor an die richtige Stelle zu setzen, gibt es in der Sprungleiste gleich drei Routinen: TXT SET COLUMN (#BB6F), TXT SET ROW (#BB72) und TXT SET CURSOR (#BB75). Wie der Name schon sagt, wird mit TXT SET COLUMN nur die Spalte auf dem Bildschirm angegeben, es entspricht also in etwa der TAB() Funktion in BASIC. Die Spalte wird hierbei in A angegeben, das Register HL wird im Unterprogramm verändert. Genau so funktioniert TXT SET ROW, nur, daß hier nicht

die Spalte, sondern die Zeile in A angegeben wird.

Mit TXT SET CURSOR wird, genau wie beim BASIC Befehl LOCATE, die Cursorposition komplett neu gesetzt. Die Position wird in HL angegeben, wobei die

man diese Routine aufruft, kommt im Register HL die aktuelle Cursorposition zurück.

Auch die Befehle PEN und PAPER gibt es der Sprungleiste. Dort heißen sie TXT SET PEN (#BB90) und TXT SET PAPER (#BB96). Die Stiftfarbe wird jeweils im Register A angegeben.

Eine Funktion, mit der sich die aktuelle Stiftfarbe ausfindig machen läßt, gibt es unter BASIC nicht, dafür aber in der Sprung-

Parameter in HL den Anfang des Textes und ein Null-Byte als Endmarkierung und wird mit einem CALL

#C341 beim CPC464
#C38E beim CPC664
#C38B beim CPC6128

aufgerufen, nachdem das ROM mit CALL #B 900 eingeschaltet wurde. Die zweite Methode ist nicht ganz so gut, da die BASIC Routine zum anderen ja auch sehr kurz ist.

1;---Stringsausgabe unter Assembler			
5;---Abbildung 1			
8000	10	org #8000	
8000	11	ent\$	
9000	210D90	13 start: ld hl,text	; Zeiger auf Text setzen
8003	7E	14 print: ld a,(hl)	; Ein Textzeichen holen
8004	FE00	15 cp 0	; Ist Zeichen die Endmarkierung ?
8006	CB	16 ret z	; Ja, zurück
8007	CD5ABB	17 CALL #BB5A	; sonst Zeichen ausgeben
800A	23	18 inc hl	; Zeiger auf nächstes Zeichen
800B	18F6	19 jr print	; und weiter
800D	0401	21 text: defb 4,1	; Mode 1
800F	1FOADA	22 defb 31,10,10	; Locate 10,10
8012	0F02	23 defb 15,12	; Pen 2
8014	0E03	24 defb 14,3	; Paper 3
8016	204886572	25 defm "Here we go"	; Print "Here..."
8023	0F01	26 defb 15,1	; pen 1
8025	0E04	27 defb 14,4	; Paper 0
8027	1E	28 defb 30	; Home
8028	00	29 defb 0	; Endmarkierung

Spalte in L und die Zeile in H steht. Weitere Register werden nicht zerstört.

I. DIE AKTUELLE CURSORPOSITION WIRD GEMELDET

Es gibt aber auch eine Routine, die den BASIC-Funktionen POS (#0) und VPOS (#0) entspricht: TXT GET CURSOR (#BB78). Wenn

man diese Routine aufruft, kommt im Register HL die aktuelle Cursorposition zurück.

Um nun ganze Texte auszugeben, gibt es mehrere Möglichkeiten: Zum einen kann man sich eine eigene, kurze Routine zur Ausgabe schreiben (siehe Abbildung 1), oder man benutzt die BASIC-Ausgabe. Diese verlangt als

Ein Tip: Cursor positionieren, Pen-Farbe wählen – all das kann man auch mit Steuercodes erledigen. Damit kann man Ausgaben für den Programmierer etwas übersichtlicher gestalten, und es spart im Programm zudem noch Platz. Im Programm von Abbildung 1 sieht man schon, wie das geht, und Abbildung 2 gibt eine Übersicht über alle CPC-Steuercodes und

deren Funktion sowie der entsprechenden BASIC-Befehle.

Wer bisher noch die Funktion TAG vermisst hat, dem kann geholfen werden: GRA WR CHAR (#BBFC) schreibt das in A angegebene Zeichen mit der aktuellen Grafikfarbe an die aktuelle Grafikposition. Aber Vorsicht: Die Register AF, HL, DE und BC werden von dieser Routine auf andere Werte gesetzt!

II. ZEICHENMATRIZEN SPEICHERN

Mal wieder ein Standard-Problem: Da hat man einen ganz tollen alt-deutschen Zeichensatz entworfen (wahlweise auch einen chinesischen, griechischen, russischen usw....), tja, und nun steht er da rum im RAM, und keiner weiß was damit anzufangen.

Natürlich gibt es jetzt die Möglichkeit, ein BASIC-Programm zu speichern, in dem die ganzen Zeichen definiert werden, neun KByte auf der Diskette, und bis die erst mal alle definiert sind, hat der Anwender den Computer schon längst zurückgesetzt.

Nein, das Ei des Columbus ist in diesem Fall das Abspeichern des Zeichensatzes als Binarprogramm. Das ist schnell, übersichtlich und braucht nur den dritten Teil des Platzes auf der Diskette. Ist das nichts?

Aber erst die Arbeit, dann das Vergnügen! Also: Um die Zeichen abzuspeichern, muß man wissen, an welcher Speicherstelle sie anfangen und wie viele Zeichen es sind. Da es aber für solche Sachen im ganzen CPC Speicherstellen gibt, sollte das kein Problem sein (die folgenden Werte be-

Abbildung 2: Übersicht über die SteuerCodes des CPC

Code	Funktion	Prameter	BASIC-Analogie
00	ignoriert		
01	nächsten Code drucken, auch wenn Steuerzeichen		
02	TXT CUR DISABLE, Cursor aus		
03	TXT CUR ENABLE, Cursor an		
04	SCR SET MODE, Modus setzen	Modenummer	MODE n
05	TAG für ein Zeichen	Zeichen	TAG/TAGOFF
06	TXT VDU ENABLE, Ausgabe an		
07	Glocke		
08	Cursor ein Zeichen zurück		LOCATE h,y
09	Cursor ein Zeichen vor		LOCATE h,y
10	Cursor ein Zeichen ab/LF		LOCATE h,y
11	Cursor ein Zeichen hoch		LOCATE h,y
12	TXT CLEAR WINDOW		CLS
13	Carriage Return (CR)		LOCATE h,y
14	TXT SET PAPER	Hintergrundfarbe	PAPER n
15	TXT SET PEN	Stiftfarbe	PEN n
16	Zeichen unter Cursor löschen		
17	Zeile bis Cursor löschen		
18	Zeile ab Cursor löschen		
19	Bildschirm bis Cursor löschen		
20	Bildschirm ab Cursor löschen		
21	TXT VDU DISABLE, Ausgabe aus		
22	Hintergrundmodus anwählen (transparent an/aus)	1=an 0=aus	
23	SCR ACCESS: Grafikmodus	Modus	
24	TXT INVERS: PEN und PAPER vertauschen		PEN n / PAPER n
25	Zeichenmatrix definieren	Nr, n1...,n8	SYMBOL,nr
26	Window definieren	n1,n2,y1,y2	WINDOW n1,...
27	ignoriert		
28	Farbstift definieren	Nr,Farbe_1,...	INK nr,...
29	Rand definieren	Farbe _1,...	BORDER f1,f2
30	Cursor home (links oben)		LOCATE h,y
31	Cursorposition bestimmen	h,y	LOCATE h,y

ziehen sich auf den CPC 664/6128): In #B734 steht die Nummer des ersten definierten Zeichens, dem bei SYMBOL AFTER angegebenen Zeichen also. Diese Nummer nennen wir im weiteren einmal NUMMER. Die Adresse dieses ersten Zeichens im RAM steht in #B736/B737. Die nennen wir einmal START.

ALLE ZEICHEN WERDEN ABGESPEICHERT

Da jedes Zeichen nun acht Byte belegt, kann man mit SAVE „NAME“, b, START, (255-NUMMER)*8 den gesamten definierten Zeichensatz abspeichern. Nun soll es aber Leute geben, die definieren nicht 255, sondern nur zehn Zeichen, und zwar die von 65 bis 75. Irgendwie verständlich,

gisch, nicht?). ERSTE muß größer oder gleich NUMMER sein, LETZTE darf nicht größer als 255 sein. Den Zeichensatz speichern wir dann mit SAVE„Name“, b, START+(ERSTE-NUMMER)*8, (LETZTE-ERSTE)*8

Alles klar? Nun haben wir also den Zeichensatz auf Diskette stehen. Es soll aber Leute geben, die ihre Dateien auch wieder gerne einladen würden. Naja, seien wir mal nicht so und zeigen auch das:

```
SYMBOL AFTER
ERSTE
LOAD„Name“, PEEK
(&B736)+256*PEEK(&B
737)
```

und schon steht der Zeichensatz wieder in alter Pracht im CPC. Toll, nicht?

CPC464-Besitzer sollen nun auf das alles nicht verzichten müssen. Sie ersetzen alle Adressen wie folgt:

```
#B734 → #B294
#B736 → #B296
#B737 → #B297
```

und schon können sie auch ihre Zeichensätze speichern.

```
10 'Abbildung 3 <1294>
20 MODE 2:PRINT"Vergrößerung des
Funktionstastenbuffers":PRINT <35CC>
30 INPUT"Gewünschte Buffergröße
in Byte";c <2EC4>
40 MEMORY HIMEM-c-1-0 <11FD>
50 READ a$:FOR i=1 TO 9:POKE HIMEM
+i,VAL("&"MID$(a$,(i-1)*2+1,2)):N
EXT <3E59>
60 DATA 110000210000c315bb <1923>
70 POKE HIMEM+3,INT((HIMEM+10)/256
):POKE HIMEM+2,(HIMEM+10)-PEEK(HIM
EM+3)*256:POKE HIMEM+6,INT(c/256):
POKE HIMEM+5,c-PEEK(HIMEM+6)*256 <60BA>
80 CALL HIMEM+1 <0A61>
90 PRINT:PRINT"Funktionstasten Buf
fer definiert, Länge(hex)=",HEX$(c
,4)" Byte." <4AC9>
```

Die Funktionstasten am CPC sind zwar eine ganz nette Sache, aber einfach zu inkonsequent durchgeführt, und das ist schade. Inkonsequent? Nun ja, zuerst einmal dürfen alle Texte zusammen nur maximal 100 Byte lang sein – das ist nur et-

III. FUNKTIONSTASTEN ERWEITERN

was mehr als eine Bildschirmzeile –, und von den ursprünglich 31 zur Verfügung stehenden Funktionstasten sind von der BASIC-Tastatur aus erstmal nur 13 erreichbar. Das ist doch inkonsequent, oder?

Nun, wollen wir einmal Schritt für Schritt diese Mißstände wenigstens teilweise beheben: Als erstes schaffen wir dem Benutzer mehr Platz für seine Funktionstastenbelegungen. Dafür gibt es prinzipiell eine Routine in der Sprungleiste, die sich KM EXPAND RESET nennt. Mit ihr kann ein neuer, völlig unbelegter Tastaturbuffer angelegt werden. Angegeben werden müssen dabei Startadresse und Länge des Buffers.

Der geübte Leser erkennt jetzt schon: „Hier ist doch was faul, klingt viel zu gut!“. Klar, er hat recht. Erst einmal geht natürlich freier Speicherplatz vom System ab, und zwar genau so viel, wie als Tastaturbuffer reserviert wird. Und zum

```

10 'ABBILDUNG 4                                <125F>
20 READ i:WHILE i<>-1:READ i1,i2,i
3:KEY DEF i,1,i1,i2,i3:READ i:WEND          <49B5>

30 FOR i=0 TO 9:KEY 128+i,CHR$(48+i)
i):KEY 140+i,CHR$(48+i):NEXT                <3A59>
40 KEY 138,".":KEY 150,".":KEY 139
,CHR$(13):KEY 151,CHR$(13):KEY 152
,"RUN"+CHR$(34)+CHR$(13)                   <422F>
50 DATA 15,128,140,48                       <1482>
60 DATA 13,129,141,49                       <14FB>
70 DATA 14,130,142,50                       <149D>
80 DATA 5,131,143,51                       <1350>
90 DATA 20,132,144,52                      <14DA>
100 DATA 12,133,145,53                     <1411>
110 DATA 4,134,146,54                     <1354>
120 DATA 10,135,147,55                    <144B>
130 DATA 11,136,148,56                   <14A3>
140 DATA 3,137,149,57                    <1349>
150 DATA 7,138,150,46                    <13E6>
160 DATA 6,139,151,152                   <14E9>
170 DATA -1                                <0915>
    
```

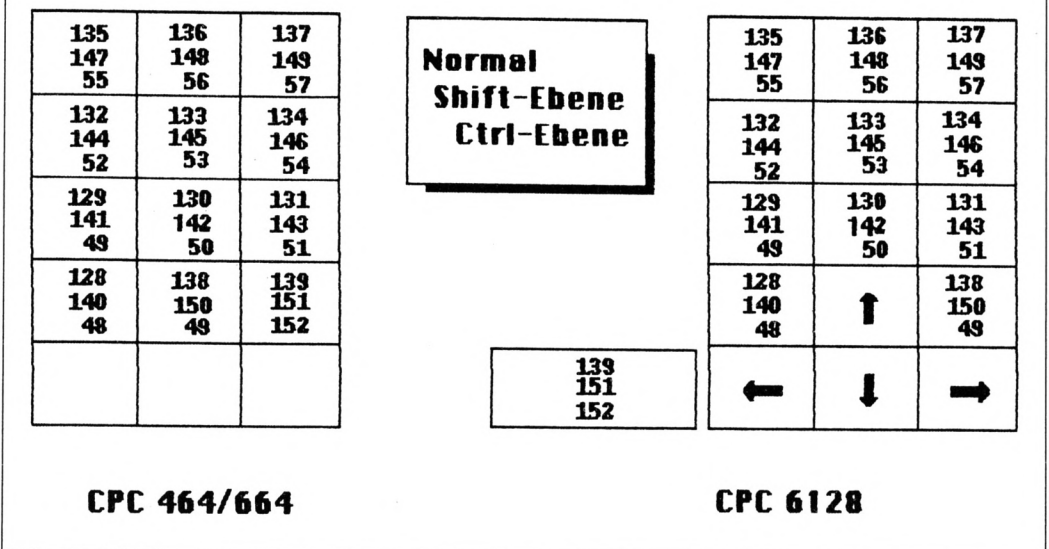
ändern sind alle Tastaturerweiterungen, die Funktionstasten also, nach der Initialisierung ohne jede

Belegung; das heißt, der Ziffern/Funktionstastenblock ist erst einmal völlig unbelegt. Doch es

sollte noch zumutbar sein, die Funktionstasten neu zu belegen. Und der Speicherplatzverlust ist ja so riedig auch wieder nicht, also: sei's drum. Ein Programm zur Belegung der Funktionstasten steht in Abbildung 3. Es ist kurz genug, daß Sie es auch in eigenen Programmen verwenden können.

Nun zum zweiten Problem: die 18 brachliegenden Funktionstasten. Das läßt sich mit Hilfe eines ganz einfachen BASIC-Programms lösen: Das Programm in Abbildung 4 belegt den Ziffernblock zweifach, so daß mit der <SHIFT>-Taste dann wenigstens 25 der 31 erweiterten Tasten nutzbar sind. Die Zuordnung der Tasten an die Erweiterungs Strings entnehmen Sie bitte Abbildung 5.

Abbildung 5: Belegung des Funktionstastenblocks



Verk. CPC 6128 grün, Am-drive, DMP 2000, MP2, Dart-scanner, Lightpen, Joy, Datenrec., Prowort, Maxam, Discovery, Discology, CPC Dictionary, Printmaster; Hefte: Aktiv, Magazin, PC Int., kompl.; Software D/K u.v.m. Kompl. 2.000,- DM. Mutuhaus, Tel. 07461/77217

*** NOTVERKAUF ***
Schneider PC 1640 SD, EGA, 40 MB, noch 4 Mon. Garantie inkl. Monitor, Zubehör u. div. Büchern. Neupreis 4.700,- DM, jetzt 3.500 DM. Tel. 089/981213
Supertausch!! Scuhe DR Graph, biete Wordstar 3.0 m. Mailmerge. P. Marquard, Tel. 0621/859140

Orig. Spiele u. Programme f. Schneider CPC zu verk. Wordstar 3.0 f. 464/664 100,- DM. Aierwolf, Combat Lynx, Hunter Killer, Infiltrator, Red Arrows, Marsport, Strike Force Harrier, je Disk 25,- DM. Lotto 6 aus 49, Disk. 30,- DM. Game Box 1, Kass. 10,- DM. Tel. 089/6370808 ab 18 Uhr

Achtung!! Tausche Programme f. Schneider CPC 464. R. Kissling, Spitalhalde 14, CH-4310 Rheinfelden od. Mailbox 004161/874788. PS. Suche Schneider Mailbox-Prog.
Gesucht: Zweitfloppy f. 6128 u. Grafik. D. Maron, Bahnstr. 26, 2085 Quickborn

Bücher — Bücher

Neue Wege bei der Veröffentlichung von Computerliteratur ging der Interest-Verlag, von dem wir hier drei Neuerscheinungen besprechen wollen. Dabei ist fragwürdig, ob Idee, Ausführung und Inhalt den recht hohen Preis rechtfertigen.

Die Methode ist bei allen drei Neuerscheinungen gleich: Die Bücher erscheinen als Einzelblatt-Sammlung in einem Ringordner, und wenn es nötig wird, verspricht der Verlag Update-Lieferungen, die dann einfach nur dazu geheftet werden müssen. Die Bestell-Postkarte liegt jedem Ringbuch bei. Bei der Kaufentscheidung kommt es natürlich auf das Thema an. Zur Hardware eines Computers, der seit drei Jahren ausgeliefert wird, läßt sich sicherlich nicht viel Neues sagen, aber es können neue Tips aus der Softwareumgebung kommen. Es gibt in der Computerliteratur sicherlich Bücher, die in sich abgeschlossen sind und die ihren Wert über lange Zeit behalten. Bei dem technischen Fortschritt in Sachen Microelektronik sind die meisten Werke jedoch schnell veraltet. Den Zweck, als Nachschlagewerk zu dienen, können sie nur noch bedingt erfüllen. Aus dieser Sicht ist die Update-Ringbuch-Methode sicherlich begrüßenswert. Der Vertrieb macht jedoch einen ziemlich hohen Preis notwendig, denn auch die Nachlieferungen kosten Geld. Wir wollen prüfen, ob dieser gerechtfertigt ist.

AKTUELLES IC-DATENBUCH

Beginnen wir mit dem schwersten Werk — womit nicht nur das Gewicht gemeint ist —, dem „Aktuellen IC-Datenbuch“. Es ist klar, daß die umfangreiche Sammlung nichts für die übliche Computerbeschäftigung ist. Sie müssen schon begeisterter Hobby-Bastler sein oder gar von Berufs wegen mit IC's zu tun haben. Wenn Sie sich bei einem neuen Computer fragen, wo der LötKolben anzusetzen ist, dann sind Sie mit diesem Buch gut bedient. Da sich der Autor Helmut Weidner direkt an den Fachmann richtet, kann er getrost auf eine allzu lange Einleitung verzichten.

Statt dessen enthält das Buch ein ausführliches Inhaltsverzeichnis, Unterkapitel werden weiter aufgliedert. Die grobe Einteilung befaßt sich mit digitalen und linearen integrierten Schaltungen, Schaltungen zur Datenerfassung und Wandlung und der Sammelrubrik der Sonder-IC's. Die ersten beiden Kapitel dürften wohl am wichtigsten sein: Bei den digitalen Schaltungen wird sowohl die Standardlogik der TTL- als auch jene der CMOS-Typen berücksichtigt. Besonders letz-



tere erhalten durch ihre stromsparende Entwicklung seine große Bedeutung bei den zunehmenden Lap Top-Rechnern.

Wer aber nun Schaltungen darstellen will, der muß auf eine Norm zurückgreifen. Die Internationale Elektrotechnische Kommission (IEC) hat dazu eine symbolische Sprache entwickelt, die vom Autor ausführlich erklärt wird. Da diese Norm auch vom DIN-Verband angenommen wurde, dürfte es keine Schwierigkeiten beim Verständnis der Datenblätter geben. Mit der Beschreibung dieser Symbolsprache und der damit verbundenen Norm

kann sich zwar niemand einarbeiten (es wird erhebliches Grundwissen vorausgesetzt), der Fachmann kommt damit jedoch wieder auf den neuesten Stand.

Das umfangreiche Buch schließt mit einem Hersteller- und Bezugsquellenverzeichnis. Für den Bastler eine große Erleichterung, da es bei der ausführlich beschriebenen Speicherchips, Prozessoren und Verstärkerschaltungen sicherlich Beschaffungsprobleme geben könnte. Alles in allem dürfte das „Aktuelle IC-Datenbuch“ dem Fachmann (und nur dem) eine große Hilfe sein. Der Update Service beweist gerade hierbei seine Vorteile, da die Entwicklung neuer Chips ständig vorangeht.

Interest Verlag
Best.-Nr. 1500 / 92,— DM

MEHR ERFOLG MIT SCHNEIDER CPC 464/664/6128

Um es gleich vorweg zu sagen: So hätte man sich das Handbuch zum CPC gewünscht. Und das, obwohl gerade die Auflistung und Erläuterung aller BASIC-Befehle, wie man sie aus dem Original-Handbuch kennt, fehlen. Irritierend ist eigentlich nur der Titel. Mancher Leser wird unter erfolgreichem Programmieren wohl eher die finanzielle Seite sehen und weniger darauf achten, daß ein leistungsfähiges Programm schon Erfolg an sich ist.

Die Reihenfolge der Kapitel ist etwas unüblich. Anfänger sollten sich das Inhaltsverzeichnis vornehmen und mit den Kapiteln beginnen, die sie besonders interessieren. Ansonsten kann es passieren, daß man ausgerechnet gleich zum Start mit einem ausführlichen, deshalb aber auch sehr trockenen Text über die Hardware samt interner Software (damit ist das ROM gemeint) konfrontiert wird. Es ist sinnvoller, die ersten Seiten nur zum Nachschlagen zu benutzen, als Lesestoff eignen sie sich kaum.

Da wird es bei der Software-Erstellung schon interessanter. Wenn schon lernen, dann so. Von der „Idee zum Programm“ heißt das wesentlichste Kapitel, in dem nichts ausgespart wurde. Es geht vom Programmwurf, beginnend bei der symbolischen Ablaufsprache,

über Problemanalyse bis hin zur Programmieretechnik. Die vielen Beispielprogramme sind in diesem Stadium zwar recht kurz, erläutern allerdings den Lernstoff bestens. Was fehlt, ist der Hinweis, daß diese wissenschaftliche Genauigkeit bei der Programmstrukturierung so

mittlung und Tilgungsplan verschrecken sicherlich den einen oder anderen Spiel-Freak, geben dem, der an Anwenderprogrammen interessiert ist, jedoch endlich einmal Lehrstoff in die Hand.

Unter der Überschrift „Programmiersprachen“ vollführt der Autor einen kleinen Seitensprung zu Pascal. Den Text als Pascal-Kurs zu bezeichnen, ist zwar etwas hochgegriffen, doch einen Einstieg zu einer anderen Programmiersprache als BASIC hat man damit schon erreicht. Mit Sicherheit ist man nach dem Lesen des Kapitels kein Pascal-Programmierer. Auf weitere Literatur muß deshalb noch gewartet werden.

Ganz wichtig war uns das Kapitel Hardware- und Software-Ergänzungen. Schließlich gibt es einige Kleinigkeiten, in denen der CPC nicht perfekt ist. Mit den Bauanleitungen für einen Reset-Schalter und einer Druckerschnittstelle mit acht Bit hat der Interest Verlag sicherlich einen guten Anfang gemacht. Die beiden Anleitungen sind einfach nachzuvollziehen und funktionieren ausgezeichnet. Da es allerdings noch viele Details gibt, an denen der Freak basteln könnte, wären schnellstmögliche Nachlieferungen angebracht. Wie wäre es, wenn sich endlich einmal jemand an das Problem einer seriellen Schnittstelle wagen würde? Notlösungen gab es ja schon.

Genauso verhält es sich mit der Rubrik „Spezielle Einsatzbereiche“. Hier dient der CPC bislang ausschließlich zur Eisenbahnsteuerung. Andere Themen wurden noch nicht aufgegriffen, man muß auf die Update-Blätter warten. Tabellen und Diagramme beschließen das Buch, auch hier wurde nur ein Anfang gemacht, und es findet sich eine Liste der BASIC-Befehle mit der kürzestmöglichen Funktionsbeschreibung. Diese Liste dient sicherlich nur dem, der sich im BASIC-Dialekt des CPC schon auskennt.

Fazit: Wir haben das große Lob schon vorweggenommen und können „Mehr Erfolg mit dem CPC“ nur empfehlen. Die bereits vorhandenen Kapitel sind ausführlich und erklärend beschrieben. Mit Ausnahme von Pascal wurde jedes Thema vermieden, welches nicht erschöpfend behandelt werden konnte. Sollte sich dieser Stil in den Nach-

lieferungen fortsetzen, dann dürfte es momentan kein besseres Lehr- und Nachschlagewerk für den CPC auf dem Markt geben.

Interest Verlag
Best.-Nr.: 2400 / 92,- DM



zwingend nun auch wieder nicht ist. Bei einem kurzen Listing darf man durchaus einmal ein Auge zudrücken, der Autor Hans Lorenz Schneider bleibt jedoch streng, um sich nicht selbst zu widersprechen. Das Kapitel Software-Erstellung schließt mit einem ausgezeichneten Grafikkurs, der keine Fragen offenläßt. Der Interest Verlag verspricht, noch einen Kurs zur Sound- und Mausprogrammierung folgen zu lassen.

Ein dicker Packen Seiten beschäftigt sich mit Musterprogrammen. Hier sind es endlich einmal ernstzunehmende Anwenderprogramme und nicht lächerliche Spielchen oder Standard-Adreßdateien. Es werden mathematische Routinen vorgestellt, die jeder in seinen eigenen Programmen unterbringen kann. Mit Werterfassung, Statistikteilen (etwa Standardabweichung und Varianz) kann sich jeder ein mathematisches Hilfsprogramm zusammensetzen. Bei den Utilities handelt es sich um Menüformen, allgemeine Hilfsroutinen wie Warteschleifen und Grafikteile sowie Speicher- und Ladehilfen.

Wie ernst man das Thema nimmt, wird mit den kompletten Programmen bewiesen. Restwertberechnung, Barwertberechnungen, Rabatter-

MUSTERPROGRAMME, UTILITIES UND MODULBIBLIOTHEKEN FÜR RAFFINIERTERE LÖSUNGEN IN TURBO PASCAL

Turbo Pascal hat sich mittlerweile auf allen Rechnern durchgesetzt und auch beim Schneider CPC wird die Hochsprache immer beliebter. Es sei Zeit, so dachte zumindest der Interest-Verlag, für alle Programmierer eine Musterprogramm-Sammlung herauszubringen, die nach Belieben ergänzt und vervollständigt werden könnte. Aber alleine dabei ist es nicht geblieben. Leider, muß man sagen!

Den Programmen hat man einen einführenden Teil vorangestellt, der einen Intensiv-Kurs und einige Erklärungen zur Programmhandhabung von Editor und Compiler enthält. Für die Besitzer von Homecomputern ist es sicherlich enttäuschend, daß Fritz Jobst, der Autor, nicht auf CP/M einging, als es sich um den Zusammenhang zwi-



schen Turbo Pascal und Betriebssystem handelte. Da die Ausführungen für MS-DOS relativ knapp gehalten wurden, wäre es wohl möglich gewesen, gleiches auch mit dem CP/M-System zu versuchen.

Aber lassen Sie sich davon nicht täuschen. Spätestens bei der eigentlichen Programmsammlung ist auch der CPC-Besitzer wieder Nutznießer. Turbo Pascal unterliegt einer gewissen Normung, und die Utilities und Programme dürften im Zweifelsfall leicht anzupassen sein.

Dem Anspruch ernsthafter Programme wird man schon bei der Auswahl der Hilfsroutinen gerecht. Es geht um Rechnerkopplung, Datei-Anpassungen und digitale Bildverarbeitung. Wer es genau nimmt, der kann diese Utilities bereits als „Entwicklungsprogramme“ bezeichnen. Hierfür ist jedoch ein eigener Buchabschnitt vorgesehen, der bislang nur das Programm eines konfigurierbaren Assemblers enthält. Weitere sollen folgen.

Komplexe Listings zu ernsthaften Anwendungsbereichen verspre-

chen die folgenden beiden Kapitel „Technisch-mathematische Programme“ und „Kommerzielle Programme“. Dabei handelt es sich um Grundaufgaben aus dem mathematischen Bereich sowie um ein Listing zur Linearen Algebra und Statistik. Als bislang einziges kommerzielles Programm findet sich die Restschuldenberechnung für Darlehen und Bereich. Eben nicht vervollständigt wird die Sammlung durch das Vokabel-Training „SO-WORT“.

Ausgefuchsten Pascal-Programmierern erscheint die Programm-bibliothek vielleicht zu knapp. Wichtiger als vollständige Listings sind jedoch Modulbibliotheken. Turbo Pascal erlaubt ja die Nutzung solcher Programmbausteine. Im entsprechenden Kapitel dieses Buches sieht es jedoch (noch) etwa leer aus.

Es gibt lediglich Softwarebausteine für ein Eingabe-Ausgabe-System und einen Grafiktreiber. Daß unter bestimmten Umständen auf allgemeine Funktionen des MS-DOS zugegriffen werden kann, ist nichts Neues.

Fazit: Von den drei Interess-Büchern ist „Musterprogramme, Utilities ...“ jenes, welches am dringendsten der Ergänzung bedarf. Statt eine möglichst ausführliche Programmbibliothek zu liefern, ver-gedeutet man zu viele Seiten für grundsätzliche Einführungen in Turbo Pascal. Das Wissen, das dort vermittelt wird, darf man aber bei jedem voraussetzen, der nach diesem Wälzer greift.

Interest Verlag
Best.-Nr.: 2500 / 92,- DM

Biete Profi-Software f. Schneider PC (Turbo Pascal, Turbo Basic, dBase III +, Lotus 1,2,3 u.a.). Alles nu u. orig.-verpackt, 50 % unter Neupreis. Info kostenlos bei: M. Lehmann, Mittelweg 11, 6000 Frankfurt

Mathe auf dem Schneider PC. Biete 31 Programme f. Sek. I, teilweise Grafik, über 200K, ideal f. Lehrer u. Schüler, menügesteuert, Druckoptionen Preis nur 50,- DM. Kostenloses Info anfordern! M. Schäfer, PF 7222, 48 Bielefeld 1

Achtung! Hobbyaufgabe. Ich verkaufe meine Orig.-Software f. Schneider, Kass. nur 5k,- DM, Disk. nur 10,- DM. Vollst. Orig., Liste anfordern bei: H.A. Kretzschmar, Postf. 210405, 41 Duisburg 1!

Gerdes-Maus m. Software, Zenith Grün-Mon. m. spez. Adapter, Wordstar (6128), Amdrum, Headline, sämtliche Schneider-Hefte, Sonderhefte, teils m. Softw., alles f. den CpC günstig zu verk. Tel. 07129/2376 abends od. Wo-ends

Verkaufe CPC 6128, DMP 2000 m. Zubehör, Lit., 34 CPC-Magazinen u. Software f. DM 1.900,- (NP 2.400,-) Uwe Winter, Goethestr. 20, 6313 Homberg 1, Tel. 06633/1788

Verkaufe CPC 6128 m. Farbmon., Drucker NLQ401 m. 30 Disk., Wordstar, Textomat, Datamat, Tex-Pack u.v.m. Nur kompl. abzugeben. Preis 2.200,- DM. Werner Rösch, Mühlweg 5, 8460 Schwandorf

Suche Tauschpartner v. Software u. Anwendung f. CPC 464. Schickt Eure Listen an: Markus Klesen, Mozartstr. 14, 6686 Eppelborn-Dirmingen. Verk. außerdem f. ca. 6,- DM „NONTERRAQUEOUS“, orig.,-verp.

Verk.: Orig.-Software: Thundercats, Tramton, Jackal u.a., Preis 25,- DM auf Disk. Suche Vortex-Modul zum Anschl. an TV od. Farbmon. f. 6128. Andreas Schmidt, Weinberg 8, 5600 Wuppertal 1

*** PUBLIC-DOMAIN ***
Wer macht mit? Sendet uns Eure Programme! Jeder Einsender erhält volle PD-Disk. zurück! K. Schäfer, Amselweg 6, 8056 Neufahrn

Top-Angebot! Tausche u. verk. Top-Games z.B. They Stole a Million 5,- DM; Elite 5,- DM; Tarzan 3,- DM usw. Schreibt an Volker Lorenz, Blücherstr. 58, 6200 Wiesbaden, Tel. 06121/44487. Liste gg. Briefm. Ich suche Sportspiele z.B. Summer Games!

Tausche u. verk. Spiele u. Anwend.-Prog. z.B. Prohibit, Elevator, Werner mach hin, Desert Fox, Die Erbschaft usw. Liste gg. Rückporto bei: Heike Lorenz, Blücherstr. 58, 6200 Wiesbaden, Tel. 06121/444487 ab 20 Uhr, nur 3“-Disk.

ORIGINALE FÜR CPC 464: 5 Sonderhefte zus. nur 15,- DM; Handyman, Discovery je 20,- DM; HIT-PAK 6 Disk.) 15,- DM. Bücher: CPC Intern 25,-; Tips u. Tricks 15,- DM. Köneke, Westendstr. 26, 6072 Dreieich. Plus 5,- DM Porto.

TYPE TEACHER Lernprog. f. Schreibmasch.-Schreiben auf IBM-Kompatiblen, CGA/Hercules-Karte, 48,- DM. Scharf, Waldspielplatz 12, 8130 Starnberg, Tel. 08151/16410

Verk. Apple Iie, Floppy, Disks (Flipper, Musik-Editor etc.), Joystick, Dokumentation, Grün-Mon., Keyboard Casio MT-52 (Rhythmen, Drums, Instrumente etc.) VB 900,- DM. Ingo Schwarz, Möllkoven 19, 4300 Essen 11, Tel. 0201/672903

Hilfe !! Möchte meinen 464 in der Elektronik einsetzen, z.B. Meßtechnik u.a. Wer hat Bau- u. Schaltpläne m. Hard- u. Software. Dringend an: R. Kosfeld, Rabenhof 41, 4800 Beilefeld 1

Sprechen Sie „C“

Die Sprache „C“

auf CPC

und Joyce

Die Sprache C konnte in der letzten Zeit mehr und mehr Freunde gewinnen. Sie hat sich als sehr populäre Programmiersprache für Mikrocomputer etabliert. Flexibilität und Effizienz sind nur zwei Schlagwörter, aufgrund derer sie sich bei der neuen Generation der Home-Computer mit dem 68000er-Prozessor durchgesetzt hat. Wie leistungsstark C ist, zeigt sich daran, daß ganze Betriebssysteme mit ihr geschrieben wurden und werden! Besitzer der Schneider-8-Bit-Systeme brauchen auf diese – nun in Mode gekommene – Sprache nicht zu verzichten. In diesem Artikel können Sie – nach den einführenden Informationen – unseren Testbericht über Arnor-C lesen, welche sowohl für den CPC6128 als auch für Joyce angeboten wird. Unsere Rubrik C-Splitter werden Sie auch in den nächsten Heften wiederfinden.

Damit keine falschen Hoffnungen geweckt werden, zu Beginn gleich der Hinweis, daß Arnor's C nur unter CP/M-Plus arbeitet! Dies bedeutet, daß im Normalfall nur JOYCE und CPC6128-Besitzer in den Genuß von C-Programmen kommen. Inwieweit „aufgemotzte“ 464er oder 664er mit diesem Programmpaket arbeiten können, falls sie über CP/M+ verfügen, haben wir nicht getestet! Ebenfalls gleich vorweg, um keinen falschen Eindruck entstehen zu lassen: Das Programmpaket ARNOR-C und das zugehörige Handbuch allein eignen sich nicht dazu, die Sprache C zu lernen! Es ist kein Lehrprogramm, sondern Anwendungssoftware. Die C-Kenntnisse muß man sich mit entsprechendem Lehrmaterial aneignen.

WAS IST C EIGENTLICH?

C ist eine – für allgemeine Anwendungen – sehr universelle Programmiersprache, die ursprünglich für das UNIX-Betriebssystem entworfen und entwickelt wurde. Sie ist aber nicht an eine bestimmte Hardware oder an ein bestimmtes Betriebssystem gebunden, sondern für

viele Maschinen verfügbar.

Wie leistungsstark C ist, zeigt sich am eindruckvollsten daran, daß beispielsweise das Unix-Betriebssystem in C geschrieben wurde.

Die C-Sprache ist sehr maschinen-nah und deshalb schwieriger zu erlernen als z.B. die Hochsprache Basic. Viele grundlegenden Ideen von C stammen von der Programmiersprache BCPL (Richards, 1969) bzw. von B (Johnson und Kernighan, 1973). Man kann C deshalb als verbesserten Nachfolger der Sprache B betrachten. Der Standard der Sprache C wurde im Buch „The C Programming Language“ von Brian W. Kernighan und Dennis M. Ritchie im Jahre 1977 festgelegt. 1978 wurden nachträglich noch einige Sprachelemente hinzugefügt. Verschiedene Anbieter von C-Compilern haben außerdem noch eigene Elemente eingebaut, so daß manchmal bereits – etwas abweichend vom Standard – verschiedene Dialekte entstanden sind (leider!).

Die wichtigste Frage bei einem C-Compiler ist also immer, ob er sich an den Standard hält und inwieweit der komplette Sprachumfang nach Kernighan und Ritchie implementiert wurde. Kleine Unter-

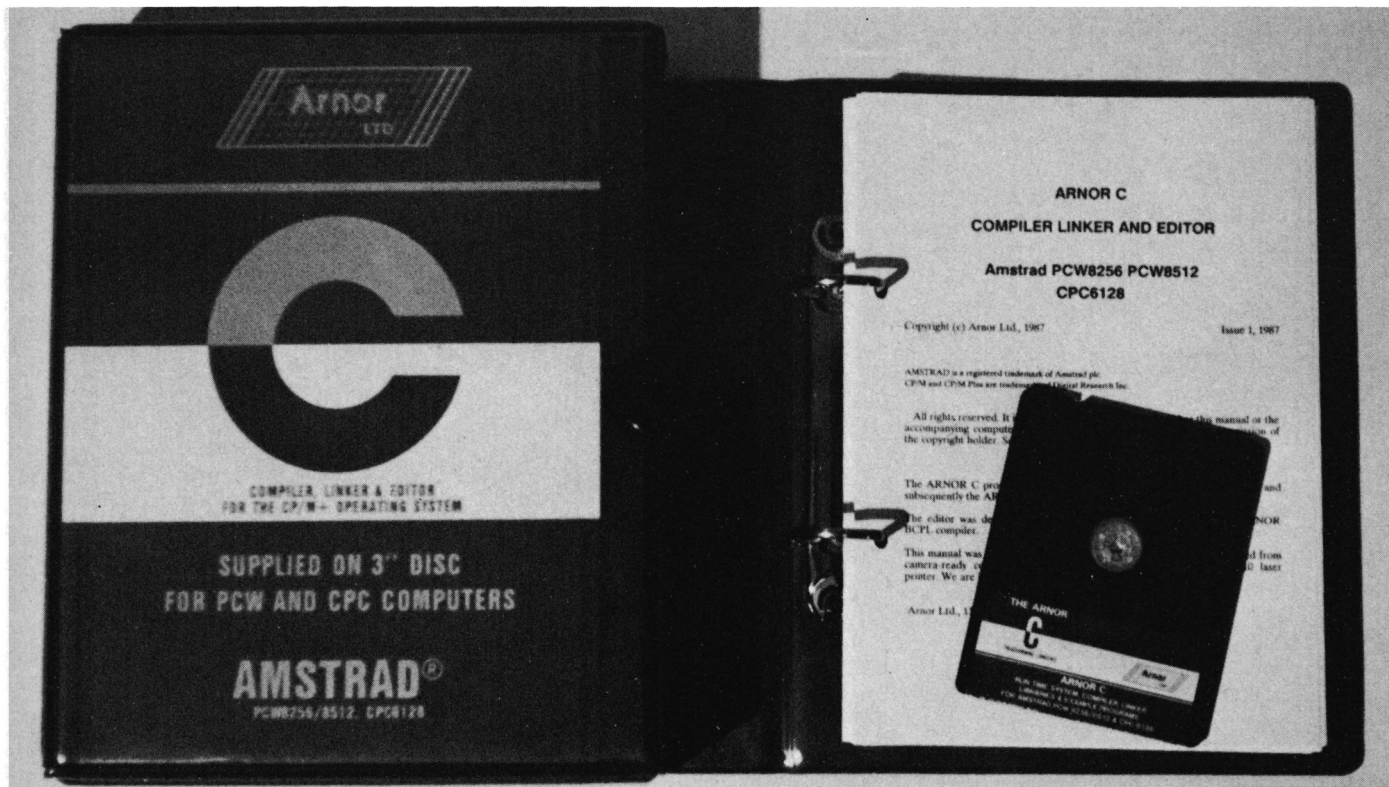
schiede und Ergänzungen sind dabei aber nicht immer von Nachteil, sondern manchmal sogar sehr sinnvoll.

WAS IST ARNOR-C?

Der Arnor-C-Compiler ist eine volle Implementation des C-Standards, wie er von Kernighan und Ritchie definiert wurde, allerdings mit kleinen Ergänzungen und einer ganz wesentlichen Besonderheit; doch von dieser später! Zusätzlich zum Standard ist das Präprozessor-Kommando #assert vorhanden, welches einen Variablenwert abfragt und dann, wenn dieser z.B. 0 ist, die Kompilation abbricht. Das Kommando #include weicht ebenfalls leicht vom Standard ab. Mit diesem Befehl kann festgelegt werden, ob das zu kompilierende File auf dem aktuellen Laufwerk oder aber auf allen Laufwerken gesucht werden soll; eine positive Erweiterung also!

WAS LÄUFT BEI C AB?

Um zu einem lauffähigen C-Programm zu kommen, müssen ganz bestimmte Stufen eines festgelegten Ablaufes erfolgen. Im Gegensatz zur Basic-Programmierung auf den CPCs mit dem eingebauten Interpreter wird ein C-Programm mit einem Text-Editor erstellt. Das erstellte „Text-File“ muß dann kompiliert und später gelinkt werden. Damit C-Programme auf unterschiedlichen Computern ablaufen können, müssen beispielsweise verschiedene Hardware-Eigenschaften berücksichtigt werden. Diese sind aber von Computer zu Computer verschieden, sie sind rechner-spezifisch. Quell-Files, also die mit einem Textprogramm erstellten ASCII-Files, können zwar von einem Computersystem übernommen, um dann auf einem anderen kompiliert zu werden, fertige Kompilate aber können meist nicht von einem Computertyp zu einem anderen übertragen und zur Ausführung gebracht werden. Rechner-spezifische Eigenheiten werden bei der Kompilation zugefügt! Auch bei Arnor's C ist dies ähnlich, denn Arnor bietet sogar zwei Möglichkeiten! Dies hängt vor allem damit zusammen, daß Speicherplatz knapp ist, und



das gilt insbesondere für die CPC-Laufwerke! Mit Platz muß vor allem auf den Disketten geizt werden.

ARNOR-C – EIN TEIL EINES ZWITTERS

Üblicherweise werden C-Programme mit einem sog. „Run-Time-Modul“ versehen und sind deshalb eigenständig lauffähig! Beim Arnorprodukt aber wird im Regelfall dieser Teil nicht hinzugefügt und deshalb wird – bei mehreren C-Programmen auf einer Diskette – sehr viel Speicherplatz gespart. Da ein Programm in dieser Form aber nicht laufen kann, muß dieser benötigte Teil im Fall des Aufrufes zugefügt werden. Man kann ihn gewissermaßen als einen Interpreter betrachten, der zusätzlich zu den Programmen in der „Kurzform“ geladen wird. Man kann diese Methode als einen Mittelweg zwischen Compiler und Interpreter sehen und spricht in solchen Fällen auch von Semicompilern.

Arnor wendet zur Speicherplatz-Einsparung also einen kleinen Trick an. Andererseits aber kann jemand, der diesen „Run“-Teil, der bei Arnor RUNC.COM heißt, nicht hat,

auch nichts mit diesen Rumpf-C-Programmen anfangen. Die Weitergabe dieser Files ist deshalb meist sinn- und zwecklos. Weshalb wir in der Zwischenüberschrift von einem Zwitter geschrieben haben, erfahren Sie etwas später.

ERST HANDBUCH LESEN!

Das Handbuch ist das erste, was bei kommerzieller Software in Augenschein genommen werden sollte (sonst kann man mit einem derartigen Programmpaket ja nicht arbeiten), und deshalb soll es auch gleich beurteilt werden. Es ist in gut verständlichem Englisch geschrieben, die Einteilung fand allerdings nicht unsere begeisterte Zustimmung. Hierfür ausschlaggebend war die Mischung von Vorgehensweisen und Erklärungen zwischen den beiden Rechnerarten CPC6128 und Joyce. Andererseits aber wäre es wahrscheinlich zu kostenträchtig, wenn für jeden der beiden Computertypen ein eigenes Handbuch geschrieben worden wäre, denn ein Programmpaket wie C wird ganz bestimmt nicht in solchen Mengen verkauft wie beispielsweise ein Textverarbeitungsprogramm.

DER LIEFERUMFANG VON ARNOR-C

Neben dem schon erwähnten Handbuch erhält der Käufer noch eine Diskette im Drei-Zoll-Format, die ziemlich „vollgepackt“ ist. C besteht also nicht nur aus einem Compilerfile, sondern aus verschiedenen Programmen mit unterschiedlichen Aufgaben. Wichtig, um C-Programme zu schreiben, ist ein Text-Editor, denn wie bei anderen Programmiersprachen auch, muß ja erst ein Quellfile geschrieben werden. Und hier staunten wir nicht schlecht, denn der mitgelieferte Text-Editor APED erinnerte uns sehr stark an das ebenfalls von Arnor stammende Textverarbeitungsprogramm PROWORT bzw. dessen englische Version PROTEXT. Kein Wunder, denn APED (Arnor Programm Editor) ist die volle Implementation des PROTEXT-Programm-Modus!

Ein äußerst leistungsfähiger Text-Editor also, mit dem es möglich ist, zwei Files gleichzeitig zu bearbeiten. Damit ist es gewährleistet, daß man auf äußerst einfache Weise Programmteile aus einem Quellfile in ein anderes umkopieren kann. Auch Funktionen wie „Suchen“ und „Ersetzen“ sind damit verfügbar.

DAS LESEN VON README-FILES IST AUSSERST WICHTIG!

Jeder Käufer sollte sich von den ersten Arbeiten zunächst das auf der Diskette enthaltene README-File ansehen, denn in diesem stehen wichtige Informationen: Korrekturen für das Handbuch, Hinweise für die ebenfalls mitgelieferten Quellfiles u.ä. Auch für uns sind diese Infos wichtig, denn wir können damit ganz genau aussagen, welche Version bzw. welchen Stand der Software wir im Test hatten (Arnor C Latest Informations 20, May 1987). So fanden wir in diesem File auch die Hinweise, daß die Library-Funktionen ESCOFF und ESCON (ESCAPE OFF und ESCAPE ON) neu aufgenommen wurden; außerdem auch Korrekturen zu BUSYPR und MATHERR sowie Hinweise für das Linken von Maschinencode und C.

README-Files sind eine einfache und praktische Methode, dem Endkäufer die neuesten Ergänzungen und Fehlerberichtigungen mitzuteilen, ohne neue Handbücher drucken zu müssen und sind neuerdings bei fast allen kommerziellen Programmpaketen üblich.

JETZT GEHT'S LOS!

Wie heute meist für kommerzielle Software üblich, sollte von der Arnor-C-Diskette zuerst eine Arbeitsversion erstellt werden, damit man im Falle eines Falles auf das Originalprogramm wieder zurückgreifen kann. Wie das durchzuführen ist, steht im Handbuch. Aber nicht in der Klarheit, wie mancher es sich wünschen würde. Da Arnor nicht weiß, welche Gerätekonfiguration beim Anwender letztendlich vorliegt, ist nur eine allgemeine Beschreibung enthalten. Das bedeutet für denjenigen, der mit C arbeiten will, daß CP/M-Grundkenntnisse vorhanden sein müssen.

TROTZ TRICK: DISKETTEN-SPEICHER TUT NOT!

Unsere ersten Tests erfolgten mit einem CPC6128 ohne externes Lauf-

werk. Da uns damit die Handhabung zu umständlich war, haben wir nach sehr kurzer Zeit dann eine Vortex-Station mit X-Modul angeschossen. Damit ließ sich weit komfortabler arbeiten, und wir mußten nicht dauernd Disk-Jockey spielen. Über die Handhabung und den Ablauf bei der C-Programmierung zu schreiben, würde hier zu weit führen. C-Einsteiger dürfen sich aber nicht darüber wundern, daß die Vorgänge bis zur Erstellung des ablauffähigen Programmes viel Zeit in Anspruch nehmen können. Dies hat aber nichts mit Arnor oder dessen C-Paket zu tun, sondern ist nun einmal so, auch auf schnelleren und teureren Maschinen. Pascal- und auch Maschinensprache-Programmierer kennen dies wahrscheinlich schon.

DIE HAUPTTEILE VON ARNOR-C

Der Editor wurde schon erwähnt, er ist sehr leistungsfähig. Aber das Schreiben von Quell-Codes ist nur eine Seite; letztendlich will man ja ein lauffähiges C-Programm erstellen. Dazu ist ein Compiler erforderlich, der den Quellcode kompiliert. Danach ist der Linker und ggf. der Joiner an der Reihe. Diese Files sind selbstverständlich im Lieferumfang enthalten.

DER ZWEITE TEIL DES ZWITTERS

Wie schon erwähnt, handelt es sich beim „Prüfling“ um einen Zwitter. Denn im Regelfall werden keine eigenständig lauffähigen Programme erzeugt, sondern „semikompilierte“ Files, die dadurch ablaufen können, daß das Runtime-Programm „RUN.COM“ aufgerufen wird.“

Nachdem wir uns am Komfort des getesteten Programmes erfreut hatten, wollten wir auch eigenständige Programme erzeugen. Wir hatten es schon im Handbuch gelesen: Dazu wird ein zusätzlicher Teil, nämlich MAKECOM.COM, benötigt, der zugekauft werden muß! Ein wesentlicher Teil fehlte also. So etwas empfanden wir aber als Zumutung und machten deshalb unse-

rer Verärgerung hierüber bei Arnor Luft.

Drei Tage später hatten wir eine neue Diskette, auf welcher dieser „fehlende Teil“ enthalten war! Vielen Dank im Namen der zukünftigen Käufer, denn durch unseren Einspruch wird das Programmpaket in Deutschland nun inklusive MAKECOM-Generator verkauft. Danke vor allem auch deshalb, weil sich am Preis dadurch nichts geändert hat.

ZUSAMMENFASSUNG ZU ARNOR-C:

Die Sprache C von Arnor ist eine wirkliche Bereicherung für die CPC-6128- und JOYCE-Welt. Es ist ein leistungsfähiges, gutes Programmpaket. Der Preis von DM 249,- ist – inklusive Programm-Generator MAKECOM.COM – im Rahmen des Üblichen und aufgrund des Komforts, der geboten wird, auch gerechtfertigt. C-Compiler für andere Computer kosten meist mehr.

Für „Profis“, die ein leistungsfähiges und komfortabel zu bedienendes C-Entwicklungspaket für den CPC6128 oder den JOYCE suchen, kennen wir derzeit nichts Besseres!

Ein zweites Laufwerk (möglichst mit höherer Diskettenkapazität als die normalen Drei-Zoll-Laufwerke) ist bei einem CPC6128 unbedingt anzuraten, da die Handhabung ansonsten aufgrund der geringen Diskettenkapazität doch sehr umständlich ist.

Ach so, fast hätten wir es vergessen, auch das sind wichtige Punkte: Der Compiler erlaubt bedingtes Kompilieren und unterstützt auch Gleitkomma-Arithmetik, die Standard-Libraries sind in verschiedenen Umfängen enthalten. Ihre richtige Auswahl kann ebenfalls Speicherplatz sparen. CPC-Eigenarten, wie beispielsweise einfache Window-Handhabungen, können berücksichtigt werden. LM



Bezugsquellen-Hinweis:

Arnor (Deutschland) Ltd.
Hans-Henny-Jahn-Weg 21
2000 Hamburg 76

HiSoft-Pascal erweitert

Lassen Sie sich nicht durch Behauptungen, HiSoft-Pascal sei nicht sehr grafikfähig, verunsichern. Gerade im Zusammenhang mit rechenintensiven Aufgaben kann Pascal seine Stärke beweisen. Ein sehr gutes Beispiel hierfür ist die automatische Erstellung eines Kreisdiagrammes.

In diesem Artikel soll Ihnen ein ganz besonderes Leckerbissen dargeboten werden. Eine Prozedur, die vollautomatisch ein Kreisdiagramm auf den Bildschirm bringt. Sie können dabei sogar ein-

zelne Segmente hervorheben. Dafür müssen zunächst zwei neue Variablentypen deklariert werden. Einmal benötigen wir ein Array, welches die Werte für die einzelnen Kreissegmente enthält. Nennen wir es „valuearray“. Ein zweites Feld mit Elementen vom

Typ BOOLEAN soll angegeben, ob ein Segment hervorgehoben werden soll (TRUE) oder nicht (FALSE). Dieser Typ heißt „explarray“, nach dem englischen Wort „explode“, das in diesem Fall als „hervorheben/-rücken“ zu verstehen ist. (Das ist kein Witz – der Fachausdruck heißt tatsächlich so!)

Die Prozeduren move und draw sind Ihnen in der Zwischenzeit wohl sattsam bekannt. Da sie nicht nur für das Kreisdiagramm benutzt werden sollen, stehen sie hier auf der ersten Ebene. Sie können damit zum Beispiel Ihr Diagramm umrahmen, beschriften oder auch durchstreichen, wenn es Ihnen nicht gefällt.

GLIEDERUNG IN HAUPT- UND UNTERPROGRAMME

Bevor wir zur Prozedur „pie“ kommen, noch eine kurze Erläuterung zu dem Ausdruck „erste Ebene“: die Experten streiten sich hier nämlich. Man kann natürlich auch das Hauptprogramm als erste Ebene bezeichnen, womit jede Prozedur automatisch auf der zweiten Ebene wäre. Ebenso gut ist es aber, wie hier geschehen, das Hauptprogramm quasi losgelöst von den Procedures und Functions als „Ebene Null“ anzunehmen. Diese Begriffsverwirrung läßt sich bei den hier vorgestellten, wenig komplexen, Programmen durch die Verwendung der Begriffe „Hauptprogramm“, „Prozedur“ und „Unterprozedur“ weitgehend vermeiden. Wird noch weiter untergliedert, können Sie sich mit „UP1“, „UP2“ und so weiter behelfen.

```

100 {*****}
110 {*   HISOFT PIECHART           Version 1.0           02.01.1986   *}
120 {*                               by Alwin Ertl           *}
130 {*****}
140
150 PROGRAM piechart;
160
170 CONST
180   pi = 3.14159;
190   maxvalues = 20;
200
210 TYPE
220   valuearray = PACKED ARRAY [1..maxvalues] OF real;
230   explarray = PACKED ARRAY [1..maxvalues] OF boolean;
240
250 VAR
260   wert : valuearray;
270   hervorhebung : explarray;
280   anzahl : integer; {alle Variablen nur fuer Demo!}
290
300 PROCEDURE move (x, y : integer);
310 BEGIN
320   rde := x;
330   rhl := y;
340   user (#bbc0) {gra move absolute}
350 END;
360
370 PROCEDURE draw (x, y : integer);
380 BEGIN
390   rde := x;
400   rhl := y;
410   user (#bbf6) {gra line absolute}
420 END;
430
440 PROCEDURE pie (n : integer; v : valuearray; e : explarray;
450               x,y, r : integer);
460 VAR
470   s, schritt, grad, stop : real;
480   xx, yy, i : integer;
490
500 FUNCTION sum (z : integer; a : valuearray) : real;
510 VAR
520   s : real;
530   i : integer;
540 BEGIN
550   s := 0;
560   FOR i := 1 TO z DO
570     s := s + a[i];
580   sum := s
590 END;
600
610 FUNCTION degree (ges : real) : real;
620 BEGIN
630   degree := (2*pi) / ges
640 END;
650
660 FUNCTION mittelwert (a, b : real) : real;
670 BEGIN
680   mittelwert := (a+b) / 2
690 END;

```

Nun aber zur Prozedur „pie“, abgeleitet von „piechart“, also Pasteten-, Kuchen- oder, ebenso deutsch wie fantasielos: Kreis-Diagramm. Übergeben wird hier zuerst die Anzahl der Segmente, die natürlich nicht größer als maxvalues sein darf. Diese Konstante können Sie selbstverständlich nach Belieben anpassen. Daran schließen sich die beiden Arrays mit den Flags für die Hervorhebung der Segmente an. Zu guter Letzt müssen Sie den Mittelpunkt und den Radius des Kreises übergeben.

BERECHNUNG DER KREISFORMEL

Drei kurze Funktionen nehmen uns umständliche Berechnungen im Prozedurkörper ab. Da ist zuerst einmal die Funktion `sum`. Sie dient dazu, alle Elemente des Werte-Arrays zu summieren. `degree4` berechnet, wieviel Grad (oder auch Grad-Bruchteile) dem Wert entsprechen. Und der Mittelwert errechnet genau das, was der Name aussagt. Wozu wir diese drei Funktionen benötigen, wird später erklärt. Die Unterprozedur `drawsegment` zeichnet den Kreisabschnitt zwischen „start“ und „ende“ an der Position (x/y) mit dem Radius r. Dazu wird die bekannte Kreisformel

$$\begin{aligned} x &:= \cos(n) \\ y &:= \sin(n) \end{aligned}$$

angewendet. Natürlich muß der Nullpunkt verschoben und mit r multipliziert werden, um den Kreis mit dem richtigen Radius an die richtige Stelle zu bekommen. Und damit es auch ein Segment wird (nicht nur

```

700
710 PROCEDURE drawsegment (start, ende : real; x,y, r : integer);
720   VAR
730     i : real;
740   BEGIN
750     move (x,y);
760     draw (round(x+cos(start)*r), round(y+sin(start)*r));
770     i := start;
780     WHILE i < ende DO
790       BEGIN
800         i := i + 0.002;
810         draw (round(x+cos(i)*r), round(y+sin(i)*r))
820       END;
830     draw (x,y)
840   END;
850
860 BEGIN {pie}
870   s := sum (n,v);
880   schritt := degree (s);
890   grad := 0;
900   FOR i := 1 TO n DO
910     BEGIN
920       xx := x;
930       yy := y;
940       stop := grad + v[i]*schritt;
950       IF e [i]
960         THEN
970           BEGIN
980             xx := round (xx + (r/5)*cos(mittelwert(grad, stop)));
990             yy := round (yy + (r/5)*sin(mittelwert(grad, stop)))
1000          END;
1010       drawsegment (grad, stop, xx, yy, r);
1020       grad := stop
1030     END
1040   END;
1050
1060 BEGIN
1070   page;
1080   anzahl := 5;
1090   wert [1] := 5.2; hervorhebung [1] := false;
1100   wert [2] := 3.8; hervorhebung [2] := true;
1110   wert [3] := 2.75; hervorhebung [3] := false;
1120   wert [4] := 2.12; hervorhebung [4] := false;
1130   wert [5] := 1.15; hervorhebung [5] := true;
1140   pie (anzahl, wert, hervorhebung, 100, 100, 40);
1150   pie (anzahl, wert, hervorhebung, 400, 200, 100)
1160 END.

```

ein einfacher Kreisbogen), werden Start- und Endpunkte noch mit dem Mittelpunkt verbunden. Eine entsprechend kleine Schrittweite bei der Berechnung der einzelnen Koordinaten vermeidet Ecken und Kanten.

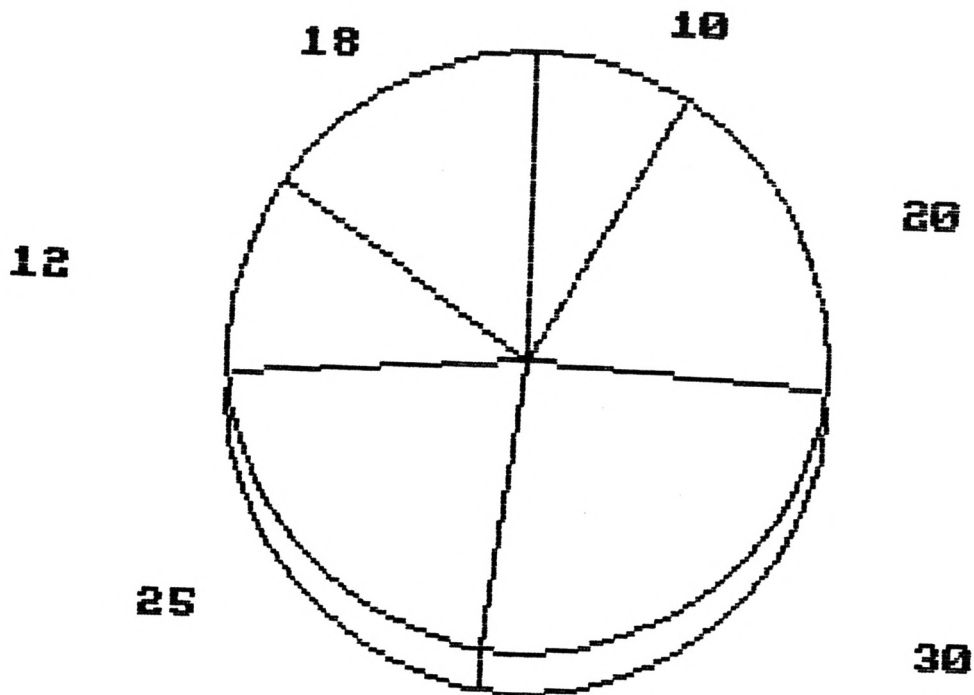
Wie aber weiß der Rechner, von wo bis wo ein Segment gezeichnet werden muß? Das ist die Arbeit, die von der Prozedur „pie“ geleistet wird. Erst einmal werden die einzelnen Werte summiert (Variable s). Danach wird dieser Wert

durch $2 * \pi$, den Vollkreis also, dividiert. Das ergibt die Anzahl der Grade, die dem Wert 1 entsprechen. Er steht jetzt in der Variablen `schnitt`. Der Startpunkt des ersten Segments liegt bei null Grad und ist in der gleichnamigen Variablen festgelegt. Für alle Werte wird nun zunächst der Mittelpunkt auf die angegebenen Parameter übertragen. Anschließend erfolgt die Berechnung des Endpunktes für diesen Abschnitt, indem zum Startpunkt der aktuelle

Wert, multipliziert mit der Schrittweite, addiert wird.

HERVORHEBUNG EINES SEGMENTES

Jetzt muß noch der Fall in Betracht gezogen werden, daß das Segment hervorgehoben werden soll. Das bedeutet: Es muß ein Abschnitt gezeichnet werden, der zu einem Kreis mit demselben Radius, jedoch mit radial nach außen ver-



schobenem Mittelpunkt, gehört. Eine Verschiebung um ein Fünftel des gewählten Halbmessers hat sich als praktikabel erwiesen. Die radiale Verschiebung haben Sie sich so vorzustellen, daß das Segment an einer Linie, die vom Zentrum des Kreises nach außen führt und dabei den aktuellen Abschnitt halbiert, verschoben wird. Deshalb muß hier der Mittelwert aus Start- und Endpunkt gefunden werden. Auf diesen wird die Kreisformel angewendet. Eine einfache Addition oder Subtraktion reicht nicht, da in diesem Fall ein Segment auf einer Seite des Kreises zwar nach außen, auf der gegenüberliegen-

den Seite jedoch nach innen rücken würde. Beachten Sie den Vorzeichenwechsel bei \sin und \cos .

Mit den so ermittelten Werten kann der Kreisabschnitt gezeichnet werden. Da das nächste Segment an der Stelle beginnen muß, an der das erste endete, wird der neue Startpunkt grad mit dem Wert des Endpunktes stop versehen.

PROBLEME MIT DER HARDCOPY? EINE EINFACHE LÖSUNG

So einfach ist es also, ein komfortables Kreisdiagramm zu entwerfen.

Das Hauptprogramm demonstriert, was unsere kurze Routine leisten kann. Sie können das Ergebnis auch auf einer Hardcopy bewundern. Diese Hardcopy wurde mit der Routine aus Teil drei angefertigt. Dort wurde erwähnt, daß sie nicht ganz verzerrungsfrei arbeitet. Den schlagenden Beweis sehen Sie hier: Was auf dem Bildschirm kreisförmig aussieht, wird auf dem Drucker zum Ei. Aber nicht zu jenem sagenhaften des Christopher Kolumbus, das kommt nämlich erst jetzt: Bauen Sie die Routine so um, daß die Kreise auf dem Bildschirm etwas flachgedrückt werden. Dann stimmt wenigstens die

Hardcopy. Eine gewisse Ähnlichkeit mit Kolumbus' Methode ist hier durchaus zu bemerken – auch er bevorzugte die einfache, aber radikale Lösung.

Bevor diese Folge nun jedoch ausartet in den alten Streit, was zuerst da war (Henne oder Ei), bereiten wir der Sache lieber ein Ende. Nicht jedoch ohne den obligaten Hinweis auf die nächste Ausgabe. Dort beschäftigen wir uns ausführlich mit Figuren, die auf dem Bildschirm dargestellt werden sollen. „Blockgrafik“ heißt das Thema. AE

Basic — Programmierung

Vom Problem zur Lösung

Unsere Leserumfragen ergeben immer wieder, daß viele von Ihnen Interesse an Anwendungsprogrammen haben. Ein ebenfalls großer Teil von Ihnen gibt zu erkennen, daß die Basic-Kenntnisse noch nicht sehr umfangreich sind. Wir haben uns nun gedacht, in Zukunft diesem Teil unserer Leser etwas zu bieten. Hier unser erster Versuch.

Die Auswertungen unserer Fragebogen ergab, daß viele von Ihnen sehr starkes Interesse an Anwendungsprogrammen haben. Anwendungsprogramme können sowohl professioneller als auch „eigen-gestrickter“ Natur sein. Professionelle Programme können Sie kaufen und einsetzen. Sie können aber auch versuchen, selbst welche zu schreiben. Da aber „hakt“ es noch bei manchem, das Wissen reicht noch nicht aus! Programmieren lernen Sie aber nicht durch Lesen von Büchern und Zeitschriften, sondern durch Übung!

WAS SOLL DENN PROGRAMMIERT WERDEN?

Dabei tritt anscheinend beim einen oder anderen ein Problem auf, nämlich die Frage: Was soll ich denn programmieren? Die Antwort darauf ist eigentlich sehr einfach: Ein Programm, welches Ihr Anwendungsproblem löst! Um Ihnen aufzuzeigen, wie Sie ein Problem lösen können, werden wir eines darstellen und Ihnen dann einen Lösungsweg und selbstverständlich ein Programm hierfür bieten und durchsprechen.

Sicher haben auch Sie schon einmal darüber nachgedacht, wie denn chiffrierte Nachrichten

decodiert werden können. In unserem Zeitalter, dem Zeitalter der Computer, sind für solche Arbeiten zwar auch noch Spezialisten erforderlich, aber ein Großteil der — oft sehr mühseligen — Arbeit kann durch Computer in sehr kurzer Zeit erledigt werden.

Bestimmt haben Sie auch schon von verschiedenen Methoden der Verschlüsselung gehört. Möglichkeiten gibt es viele. Eine vielleicht allseits bekannte Methode ist das Verschlüsselungsquadrat.

Das heißt, der zu codierende Text wird beispielsweise senkrecht in ein Rasterfeld mit gleicher Spalten- und Zeilenzahl eingegeben und waagrecht ausgelesen.

Selbst wenn Sie aber bei einem Text wissen, daß er nach dieser Methode verschlüsselt wurde, ist es „zu Fuß“ trotzdem schwer, die Decodierung durchzuführen. Lassen Sie doch Ihren Computer ran!

VERSUCHEN SIE ES ZUERST OHNE PROGRAMM!

In Zeile 120 des zu diesem Artikel gehörenden Listings sehen Sie eine „verschlüsselte“ Mitteilung.

Versuchen Sie doch einmal, ohne Computerunterstützung herauszufinden, wie die Nachricht lautet! Wenn Sie es so

```

100 MODE 2:PRINT"Loesung fuer Vers
chluesselungsquadrat:";PRINT"-----
-----"
                                                    <5B64>
110 PRINT"Waagerecht schreiben,":P
RINT"Senkrecht auslesen!":PRINT <3635>
120 'INPUT "Bitte Text eingeben";c
ode$ <28B3>
130 code$="SRLZI - C-TEFDW HC,ITIE
NP T EL ECDSF T I-ICUC! DWEHEP E
E RRC" <4DEA>
140 laenge=LEN(code$):wurzel=INT(S
QR(laenge)) <360E>
150 IF wurzel^2<laenge THEN code$=
code$+CHR$(32):GOTO 140 <3792>
160 FOR j=1 TO wurzel <16D3>
170 FOR k=0 TO laenge STEP wurzel <20EB>
180 element$=MID$(code$,j+k,1) <2602>
190 klar$=klar$+element$ <1F45>
200 PRINT element$;" " <1615>
210 NEXT k:PRINT:NEXT j <1297>
220 PRINT:PRINT"Klartext: ";klar$ <1C7E>
230 ' <07D6>
240 FOR j=1 TO wurzel:FOR k=0 TO l
aenge STEP wurzel <3299>
250 element$=MID$(klar$,j+k,1):cod
e1$=code1$+element$:NEXT k:NEXT j <4F07>
260 PRINT:PRINT"Codierter Text: "c
ode1$ <22CB>

```

schnell schaffen, wie Ihr CPC mit dem abgedruckten Programm, dann sind Sie absolute Spitze!

VOM PROBLEM ZUR LÖSUNG

Sie wissen nun also, um was es geht und haben die Aufgabe, ein Programm zu schreiben, welches diesen verschlüsselten Text als Klartext ausgibt! Bevor Sie sich nun auf Ihren Computer stürzen und versuchen, wie ein Weltmeister zu programmieren, sollten Sie vielleicht erst einmal überlegen, wie ein Lösungsweg aussehen könnte.

Bei einem Quadrat ha-

ben wir es mit einer Fläche zu tun, deren vier Seitenlängen gleich sind. Denken Sie zum Beispiel an ein Schachbrett. Die Buchstaben des Klartextes wurden so auf die Felder geschrieben, daß der Text — links oben beginnend — nach rechts „läuft“, so, wie eben normalerweise geschrieben und gelesen wird. Ausgegeben wurde der Text aber derart, daß von oben nach unten „gelesen“ wurde.

Obwohl Sie den codierten Text kennen, wissen Sie nicht, welche Seitenlänge das Quadrat hat — oder doch? Wenn, wie in diesem Beispiel, eine sehr einfache Verschlüsselung

erfolgt, kann Ihnen die Länge des Textes diese Information liefern! Bei einer quadratischen Anordnung muß der Text im Quadrat komplett enthalten sein. In diesem simplen Beispiel muß die Quadratzahl der Seitenlänge also gleich oder größer der Zeichenkettenlänge sein! Das heißt, die nächst-mögliche, ganzzahlige „Wurzel“ aus der Länge der verschlüsselten Nachricht ergibt die Seitenlänge des Quadrates.

ES GEHT NICHT OHNE MATHEMATIK

Kurz zur Erinnerung und Auffrischung: Die Wurzel einer Zahl ist die Zahl, die mit sich selbst multipliziert (malgenommen) die Quadratzahl ergibt. Für diese Erklärung bitte ich Mathematikprofessoren um Nachsicht! Die Wurzel der Zahl 9 zum Beispiel ist 3, denn $3 \cdot 3 = 9$.

Wenn Sie nun beispielsweise einen Text mit 30 Zeichen Länge haben, dann paßt dieser in ein minimales Feld mit einer Seitenlänge von sechs, denn eine Seitenlänge von fünf ergäbe als maximale Textlänge nur 25 Zeichen.

Damit haben wir bereits den Lösungsweg dieses Problemes. Wir brauchen nun nur – beim ersten Buchstaben beginnend – den codierten Text z.B. vertikal (senkrecht) in ein gezeichnetes Quadrat einzutragen und horizontal (waagrecht) zu lesen. Aber wir wollen diese Aufgabe ja nicht selbst erledigen, sondern ein Programm schreiben, das uns diese Arbeit automatisch abnimmt.

DIE PROGRAMMSTRUKTUR

Zunächst einmal soll das

Programm beim Programmstart den Bildschirm löschen. Außerdem wollen wir auch festlegen, in welchem Darstellungsmodus wir arbeiten wollen. Wir entscheiden uns für die 80-Zeichen-Darstellung. Es wäre aber Unsinn, wenn wir erst den Bildschirm löschen (Befehl: CLS) und dann den Modus auswählen würden, denn der Befehl für die gewünschte Darstellungsart (MODE) führt automatisch auch ein Löschen des Bildschirms durch. Da wir die 80-Zeichen-Darstellung wünschen, lautet der komplette Befehl: MODE 2.

Damit wir aber nach dem Programmstart sehen, welchen Zweck unser Programm erfüllen soll, geben wir eine entsprechende Mitteilung durch den PRINT-Befehl aus. PRINT ist in seiner einfachsten Form die Anweisung an den Computer, etwas auszugeben. Wenn wir Text ausgegeben haben wollen, dann kann dies beispielsweise so geschehen, daß nach dem PRINT-Befehl der gewünschte Text in „Gänsefüßchen“ eingeschlossen steht. Hierfür ist im Programm die Zeile 100 zuständig. Die Zeile 110 bewirkt die Ausgabe des Textes: „Waagrecht schreiben“, und in der folgenden Zeile: „Senkrecht auslesen!“.

Als nächstes wollen wir dem Computer mitteilen, welcher Text decodiert werden soll. Diese Mitteilung kann beispielsweise über eine Zuweisung oder auch durch den Input-Befehl erfolgen. Der INPUT-Befehl in seiner einfachsten Form benötigt aber einen Zusatz.

Der Computer muß sich unsere Eingabe vorübergehend merken können, deshalb gehört zum INPUT-Befehl noch die

Zuweisung eines Variablen-Namens. Aber diese Form der Input-Anweisung ist für den Bediener des Programmes nicht sehr aufschlußreich. Die Aufforderung zur Eingabe erfolgt – seitens des Computers – durch Ausgabe eines Fragezeichens und des Cursor-Symboles.

Bei Programmen, deren Erstellung schon längere Zeit zurückliegt, wissen wir bestimmt nicht immer, was als Eingabe zu folgen hat. Eine kurze Mitteilung ist deshalb bestimmt sinnvoll. Wenn Sie Zeile 120 des Programmlistings betrachten, dann sehen Sie, wie dies erfolgen kann. Allerdings haben wir diese Zeile durch das vorgesetzte Apostroph „außer Gefecht“ gesetzt. Dieses Zeichen ist zu vergleichen mit dem Befehl REM. Alles, was hinter diesem Befehl kommt, wird vom CPC nicht ausgeführt, sondern als Bemerkung betrachtet. Um uns die Arbeit zu erleichtern, haben wir in der folgenden Zeile der Variablen code\$ die codierte Zeichenfolge direkt zugewiesen. Falls nämlich noch Programmfehler enthalten sind, dann braucht nach deren Korrektur diese Zeichenfolge nicht jedesmal mühsam von Hand eingegeben zu werden. Für spätere Programmläufe kann dann diese Zeile entfernt und die vorhergehende aktiviert werden.

DER CPC WEISS DURCH DIE ZEICHENKETTE ALLES „NÖTIGE“!

Alles, was wir dem Computer an Informationen geben konnten, haben wir getan. Die Zeichenfolge ist ihm bekannt, und die Länge dieser Zeichenkette kann er sich ausrechnen. Der Befehl

hierzu lautet LEN(string). LEN können Sie sich leicht merken (dies gilt übrigens für die meisten Basic-Befehle), wenn Sie der englischen Sprache mächtig sind. Er wurde aus dem Wort „length“ abgeleitet.

Die Folge von „laenge=LEN(code\$)“ läßt sich mit: die Länge der Zeichenkette codes in der Variablen „länge“ merken, übersetzen.

Nachdem der Computer die Länge berechnet hat, soll er nun die Wurzel ziehen, damit er die Seitenlänge des Quadrates weiß. Die Wurzel (in diesem Falle die Quadratwurzel) wird durch den Befehl SQR(länge) gezogen. Da es aber keine halben Felder usw. gibt, ist die Lösung nur dann richtig, wenn es sich bei der „Wurzel“ um eine echte Integer-Zahl handelt. Deshalb wird in Zeile 150 gleich festgelegt, daß für die Variable „wurzel“ der Integerwert berechnet wird. Wenn die Mitteilung aber nicht so lang ist, daß sie ein Quadrat ganz ausfüllt, dann sind noch „Leerstellen“ im Text. Dadurch würde sich über die Integerberechnung ein Fehler einschleichen. Wir fangen dies in Zeile 160 ab, indem immer dann, wenn das „Quadrat der Seitenlänge“ kleiner ist als die Anzahl der Zeichen, ein Leerzeichen angefügt und die Berechnung neu durchgeführt wird. CHR\$(32) ist der Code für ein Leerzeichen.

DIE „WURZEL“ DARF NUR GANZZAHLIG SEIN!

Zur Verdeutlichung des eben Geschriebenen: Die Länge der Mitteilung beträgt 62 Zeichen. Berechnen wir daraus die

Quadratwurzel, dann ergibt sich ein Wert von ca. 7,87. Dieser Wert aber kann in unserem Falle nicht die Seitenlänge des Quadrates sein, denn wir arbeiten ja nicht mit Bruchteilen von Buchstaben. Deshalb die künstliche Verlängerung der Zeichenkette, eine neuerliche Berechnung der Wurzel und dann wieder der Vergleich usw. Dies erfolgt so lange, bis der Wert für „Wurzel“ – in unserem Beispiel – der Zahl acht entspricht.

Nun ist die Seitenlänge bekannt. Was aber noch zu tun ist, ist, die einzelnen Buchstaben in korrekter Weise zusammenzufügen bzw. auszugeben. Zu diesem Zweck programmieren wir zwei Schleifen, die uns den jeweils aktuellen Buchstaben, im Programm `element$` genannt, herausfiltern. In unserem Programm geschieht dies in den Zeilen 160 bis 210. Dabei erledigen wir in diesen Zeilen gleich zweierlei: Einmal wollen wir den Klartext so ausgeben, wie er im ursprünglichen Quadrat stand, und zum anderen wollen wir ihn gleich in einen neuen String schreiben, der anschließend ausgegeben werden soll. Wir wissen es und der CPC berechnet es aus der Länge; unser Beispieltext stand in einem 8*8-Quadrat. Deshalb brauchen wir nur, beim ersten Zeichen beginnend, jeweils immer das neunte (acht plus eins!) Zeichen zu lesen, bis wir am Ende angelangt sind; dann ist der gleiche Vorgang zu wiederholen, nur diesmal beim zweiten Zeichen beginnend usw! Aber das soll ja unser Programm machen. Die Lösung ist ganz einfach: Eine Schleife (Zeile 160) zählt immer von 1 bis „wurzel“ und die andere von 0 bis

„länge“, mit der Schrittweite von „wurzel“. Die Schrittweite wird durch den Befehl STEP und den anschließend stehenden Ausdruck definiert. Die Zeile 180 isoliert nun den jeweilig korrekten Buchstaben aus der Gesamtzeichenkette. Denken wir uns das einmal theoretisch durch.

Beim ersten Durchlauf der Zeilen 160 bis 210 ist der Wert in der Variablen „j“ = 1, der Wert in „k“ = 0. Das heißt, in der Zeile 180 wird der Stringvariablen „element\$“ der erste Buchstabe (= „S“) zugewiesen. Dies können wir leicht feststellen, wenn wir die Variablennamen in Zeile 180 durch ihre Werte ersetzen. Es ergibt sich folgender Befehl:

```
element$ = MID$(
cod$, 1+0, 1).
```

MID\$, in dieser Befehlsform angewandt, ordnet `element$` die Zeichenkette aus `code$` zu, die an der Position `j+k (=1)` beginnt und ein Zeichen lang ist.

Durch Zeile 190 wird die Zeichenkette für „klar\$“ zusammgebaut. Beim ersten Durchlauf enthält „klar\$“ deshalb das Zeichen „S“.

Durch Zeile 200 wird das jeweils aktuelle Zeichen und anschließend ein Leerzeichen ausgegeben. Das Semikolon am Ende dieser Zeile weist den CPC an, nun keinen Zeilenvorschub auszuführen, sondern in der gleichen Zeile zu bleiben! Der Befehl „NEXT k“ in Zeile 210 bewirkt, daß der Schleifenwert für „k“ erhöht und dort weitergemacht wird, wo die entsprechende FOR-NEXT-Schleife begann. Es wird deshalb bei Zeile 170 (!) weitergemacht. Diese bewirkt nun eine Erhöhung des Wertes für „k“ um den Wert „wurzel“. Das heißt, „k“ erhält eine Er-

höhung um acht und hat nun den Wert 8! Wenn Sie Zeile 180 nun wieder „auflösen“, dann steht dort:

```
element$=MID$(
code$,8+1,1).
```

Das zweite Zeichen steht also an neunter Stelle und entspricht dem Zeichen „C“. Beim nächsten „NEXT k“ wird wieder um 8 erhöht usw.

Wenn Sie sich die Mühe machen wollten, können Sie nun – Schritt für Schritt – den Programmablauf vollziehen. Der Befehl PRINT in Zeile 210 bewirkt, daß bei der Ausgabe, nach jeweils acht Zeichen, ein Zeilenvorschub durchgeführt wird. Nach diesem wird dann der Wert für j erhöht. Da in der Schleife für „j“ keine Schrittweite angegeben ist, wird jeweils um eins erhöht. Nach der ersten PRINT-Ausführung in Zeile 210 wird bei Zeile 160 weitergemacht. Der Wert für „j“ ist dann zwei. Gleichzeitig wird durch Zeile 170 der Wert für k wieder auf 0 gesetzt. Dadurch ergibt sich für Zeile 180 nun folgende „Wertstellung“:

```
element$=MID$(
code$,0+2,1).
```

Das heißt, nun wird beim zweiten Buchstaben begonnen und die innere Schleife (wieder neu) abgearbeitet. Dadurch stehen die aktuellen Zeichen nun an den Positionen: 2, 10, 18, 26 usw.

Diese Vorgänge wiederholen sich so lange, bis der komplette Text entschlüsselt, d.h., sowohl in der 8*8-Matrix ausgegeben als auch in `klar$` zusammgebaut ist. In Zeile 220 wird dann der decodierte String ausgegeben.

UND ALLES AUCH NOCH UMGEKEHRT

Die Zeile 240 bewirkt

das Gegenteil, nämlich die Rück-Verschlüsselung des Klartextes in die ursprüngliche Form. Deshalb kann dieses Programm auch zur Verschlüsselung eingesetzt werden. Es ändert sich dann lediglich immer der Sinn der Ausgaben. Aus Klartext wird codierter Text und umgekehrt.

Ein kurzer Hinweis bezüglich des Programmes sei uns noch gestattet: Bei der Ausgabe der Zeichenkette in der quadratischen Darstellung fehlt selbstverständlich eine Zeile, denn der ursprüngliche Klartext war ja nur 54 Zeichen lang, der Rest waren Leerzeichen!

HAT ES IHNEN GEFALLEN?

Wir hoffen, daß wir vor allem Einsteigern zeigen konnten, wie sie von einem gestellten Problem zur Programmlösung gelangen. Was wir erreichen wollten ist, daß Sie Anregungen bekommen, wie Probleme überdacht, Lösungswege gesucht und programmtechnisch umgesetzt werden können.

Wenn Ihnen diese Anleitungen zusagen, dann teilen Sie uns dies bitte mit, denn wir machen die Schneider-CPC-Welt nicht für uns, sondern für Sie und deshalb sollten Sie auch mitbestimmen, was (und in welcher Art) Sie in Zukunft bei uns lesen wollen. Diese Aufforderung gilt für Themen, die Sie gelöst haben wollen. Bedenken Sie dabei aber, daß es unmöglich ist, ein Programm, welches aufgrund der Komplexität des Problems 20 Seiten Listing umfassen muß, nicht in dieser Rubrik abgehandelt werden kann. Mal sehen, welche Aufgaben Sie uns stellen. LM

Wir programmieren in Maschinensprache

So arbeitet unser Eingabe-Prüfprogramm

Checker

Nachdem wir im vorigen Heft bereits einige Routinen zur Hexzeichen-Ausgabe besprochen haben, wenden wir uns diesmal einem etwas komplexeren Thema zu: Das Eingabe-Prüfprogramm CHECKER, welches Ihnen Programmeingaben erleichtern soll, wird besprochen. Daß wir bei diesem zwar noch kurzem, aber gegenüber unseren bisherigen Assemblerlistings dennoch schon relativ umfangreichen, Programm nicht alle Details bis aufs Kleinste durchsprechen können, verstehen Sie sicher. Dennoch werden wir uns bemühen, Ihnen möglichst viele Hintergrund-Informationen zu geben, denn diese erscheinen uns sehr wichtig, weil sie nicht nur bei einem Checksummer berücksichtigt werden müssen.

Immer wieder wurden wir gebeten, doch einen Checksummer zu veröffentlichen und bei allen unseren Listings, die Zeilen, mit einer Prüfsumme versehen, abzudrucken. Lange haben wir uns geweigert. Der Grund für unsere Weigerung war bestimmt nicht, daß wir hierzu nicht in der Lage gewesen wären, nein, wir standen nur auf dem Standpunkt, daß der Lerneffekt durch die Checksummer gewaltig verringert werde. Viele Fehler werden durch den Vergleich mit den Prüfsummen zwar eliminiert, aber der „Abtipper“ lernt weniger, er muß nichts mehr im Handbuch nachschlagen, sondern nur noch darauf achten, daß er richtig abtippt.

Wir diskutierten in der Redaktion immer wieder darüber, ob wir nun unsere Listings mit Prüfsummen versehen sollten oder nicht. Vieles sprach dafür,

vielen aber auch dagegen.

Wie Sie sicherlich bemerkt haben, ist, seit unserer letzten Ausgabe, nun doch die Entscheidung zugunsten eines Eingabe-Prüfprogrammes gefallen. Doch wir wollen Ihnen diese Eingabe-Hilfe nicht nur einfach geben, sondern auch erklären, wie sie arbeitet. Dabei erfahren Sie wieder einiges – zum Teil sehr spezifisches – über Ihren CPC, und deshalb erfolgt die Besprechung im Rahmen der Artikelserie über die Maschinenspracheprogrammierung.

PROBLEME, GEDANKENGÄNGE UND BACKGROUNDWISSEN

Bevor wir uns an die Checksummer-Entwicklung machen, müssen wir einige grundsätzliche Dinge beachten und berücksichtigen!

Das Problem, daß ein

abgetipptes Programm nicht läuft, obwohl kein offensichtlicher Fehler erkennbar ist, kann daran liegen, daß der Speicherplatz für das Programm nicht mehr reicht! Dieser Effekt ist ganz gewiß noch nicht sehr oft bei Ihnen aufgetreten, es kann aber geschehen. Wie aber kann es überhaupt dazu kommen, wenn beim Programmator und auch beim Test in der Redaktion genügend Speicherplatz vorhanden war!?

Wir führten Telefongespräche mit Lesern, bei denen dieser Effekt auftrat, also muß es einen Grund geben, der dies bewirken kann. Zeilenweise Vergleiche mit dem Abdruck zeigten keine Unterschiede. Sehr mysteriös, das alles! Dieses Problem ist aber gar nicht so schwer zu verstehen, wenn man eine spezielle Eigenart der CPCs betrachtet:

Alle CPCs übernehmen im Normalfall auch Leerzeichen in den Basiczeilen. Diese können sowohl am Anfang und Ende einer Zeile als auch zwischen den einzelnen Statements stehen. Mittels einer kleinen Programmzeile können Sie sich sehr leicht davon überzeugen. Geben Sie bitte ein

```
10 PRINT „Test“
```

und am Ende des Textes „Test“ betätigen Sie nicht gleich die ENTER-Taste, sondern geben noch Leerzeichen (Spaces) ein. Wenn Sie sich dieses „Programm“ mittels eines Monitorprogrammes ansehen, werden Sie sich davon überzeugen können, daß die Leerzeichen tatsächlich im Speicher des CPCs stehen.

Die Tatsache der Leer-

zeichen-Übernahme können Sie aber auch anders feststellen: Geben Sie EDIT 10 ein und fahren Sie mit der Cursor-Rechts-Taste an das Ende der Basiczeile. Sie werden bemerken, daß der Cursor nun nicht nach dem zweiten „Anführungszeichen“ stehen bleibt, sondern erst einige Stellen danach, nämlich um so viele Stellen nach rechts versetzt, wie Sie vorher Leerzeichen eingegeben haben. Auch das ist ein Beweis dafür, daß diese Leerzeichen in den Speicher übernommen wurden.

Sicher ahnen Sie nun bereits, daß es durchaus vorkommen kann, daß bei einem Vergleich eines abgetippten Programmes mit dem gedruckten Basiclisting keine Fehler festgestellt werden und trotzdem das Programm mit „OUT of memory“ beim Lauf abbricht. Nämlich dann, wenn beim Abtippen zuviele Leerzeichen eingegeben wurden.

Es wäre kein Problem, einen Checksummer so zu programmieren, daß dieser bei der Prüfsummenberechnung Leerzeichen nicht berücksichtigt. Aber dann wäre ein Teil des Sinnes eines Checksummers schon wieder hinfällig. Denn so kann es ja vorkommen, daß das abgetippte Programm doch nicht dem Original entspricht! Vor allem würden mit Leerzeichen versehene Ausgaben nicht unbedingt stimmen.

Eine andere Möglichkeit wäre, daß der Checksummer nur die überflüssigen Spaces am Ende einer Basiczeile unberücksichtigt läßt. Eine kleine Unterroutine des Programmes bräuchte – be-

```

100 '***** <1AB6>
110 '** FCLEAN ** <1A7A>
120 '** (c) by LM ** <1A92>
130 '***** <1AF2>
140 ' <0722>
150 POKE &AC00,&FF:MODE 2:r$=CHR$( <1CC5>
24)
160 PRINT r$;"FCLEAN = File-Clean <3B53>
r von Lothar Miedel";r$:PRINT
170 PRINT r$;"Achtung: Das ASCII-F <42BC>
ile muss ohne .EXT vorliegen";r$
180 PRINT:INPUT"Wie heisst das zu <3C95>
behandelnde ASCII-File";filnam$
190 OPENOUT"druckf.zwi":OPENIN fil <1D10>
nam$
200 IF NOT EOF THEN LINE INPUT#9,a <1855>
$ELSE 250
210 laenge=LEN(a$) <1785>
220 IF RIGHT$(a$,1)=" "THEN a$=LEF <345C>
T$(a$,laenge-1)ELSE GOTO 240
230 GOTO 210 <0950>
240 PRINT#9,a$:GOTO 200 <12F9>
250 CLOSEIN:CLOSEOUT <08AA>
260 PRINT:PRINT:PRINT"Moment bitte <3765>
noch, gleich steht das File:":PRI
NT
270 PRINT TAB(30);CHR$(24);filnam$ <2313>
;CHR$(24)
280 PRINT:PRINT"'geCLEANT'' im Sp <271C>
eicher!":PRINT:PRINT
290 PRINT"Dann muss es als bereini <44EF>
gtes Basicfile abgespeichert werde
n!"
300 LOAD"druckf.zwi" <12C5>

```

ginnend am Ende einer jeden Zeile – nur die hexadezimalen Werte für Space (&20) unberücksichtigt zu lassen und nicht in die Prüfsumme aufzunehmen! Doch weit gefehlt, denn dann spielt uns eine CPC-Eigenheit einen Streich: Steht als letztes Statement einer Basiczeile ein hexadezimaler Wert, dessen High-Byte ebenfalls &20 ist (z.B. &20FF), kann es bei einer Prüfung auf „Zeilen-Endspaces“ bereits zu Komplikationen kommen. Woher sollte ein einfaches Maschinenprogramm auch wissen, ob das das letzte Zeichen einer Zeile nun ein Teil einer Hexzahl oder ein Leerzeichen ist? Deshalb

müßten weitere Abfragen erfolgen und dieser Teil würde umfangreicher werden. Bei der Entwicklung unseres Checksummers stießen wir tatsächlich auf dieses Problem, denn wir waren anfangs auch davon ausgegangen, die letzten Spaces einfach zu übergehen. Wir stießen plötzlich aber darauf, daß die Prüfsummen bei unserem Druckprogramm und beim Eingabe-Prüfprogramm unterschiedlich waren, da wir im Checksummer eine Unteroutine hatten, die überflüssige Spaces einfach nicht berücksichtigte. Gleichzeitig wurde dann natürlich auch ein High-Byte mit dem Wert &20xx nicht berücksich-

tigt. Deshalb mußte also ein anderer Weg gegangen werden. Doch davon später.

VERNICHTUNG VON FÜHRENDEN SPACES

Eine weitere Eigenart der CPCs gegenüber anderen Computern ist, daß nach den Zeilennummern auch führende Spaces stehen können. Diese Leerzeichen können zwar dazu verwendet werden, um ein Programm-Listing übersichtlicher zu machen, andererseits aber können sie bei der Eingabe leicht übersehen werden und führen dann zu Prüfsummenfehlern.

Können Sie sich vorstellen, wie wir einem Leser weiterhelfen sollen, wenn er uns sagt, die Zeile sei genauso eingegeben worden, wie wir sie gedruckt haben und trotzdem wird eine andere Prüfsumme ausgegeben? Sagen Sie bitte nun nicht, dies käme nicht vor: Es hat sich bis heute noch nicht herumgesprochen, daß der „Hochpfeil“ bei der Ausgabe über einen Drucker nur als „kleines Dach“ erscheint, und wir werden immer wieder danach gefragt, wo sich denn dieses Zeichen auf der Tastatur befinde!

Die eben angeführten Leerzeichen müssen deshalb ebenfalls unterdrückt bzw. entfernt werden. Das bedeutet andererseits aber auch, daß bei uns in der Programm-Redaktion alle Leserprogramme vor dem Ausdruck erst einmal bearbeitet werden müssen, denn wir können und dürfen nicht erwarten, daß alle Programm-Einsender diese Zeilen lesen und alle „führenden“ oder „endenden“ Spaces entfernt haben. Sie müssen aber weg! Da-

mit wir dies nicht von Hand erledigen müssen, haben wir uns hierzu ein kleines Hilfsprogramm geschrieben. Den Namen des Programmes haben wir mit FCLEAN festgelegt, um über den Namen gleich auf den Sinn hinzuweisen: FCLEAN stammt von FileCLEANer, also einem „File-Reiniger“.

Der Sinn dieses Programmes ist der, die abzudruckenden Programm-Files von den unerwünschten Leerzeichen am Anfang und Ende von Basiczeilen zu befreien. Obwohl dieses Programm vor allem wegen der Checksummer-Angelegenheit geschrieben wurde, kann es bei Ihnen auch zu anderen Zwecken sinnvoll eingesetzt werden. Haben Sie nämlich Programme mit überflüssigen Leerzeichen, so können Sie diese damit behandeln und werden plötzlich feststellen, daß Sie mehr Platz für Strings haben (die Garbage-Collect-Routine wird weniger aufgerufen). Evtl. haben Sie dann plötzlich auch mehr Platz auf einer Programmdiskette.

Daß wir dieses reine Basic-Programm in der Rubrik über die Maschinensprache abhandeln, ist darin begründet, daß wir sonst alles aus dem Zusammenhang reißen würden und vieles dann noch doppelt schreiben müßten. Wir hoffen, Sie haben dafür Verständnis.

PROGRAMMBESPRECHUNG: FCLEAN

In den ersten Zeilen (100 bis 140) sehen Sie den Programmkopf, der nur aus REMarks besteht. In Zeile 150 wird in die Speicherstelle &AC00 der Wert &FF eingeschrieben.

```

;*****
;* CSUMMER Version 1.0 *
;* fuer CPC6128 *
;* (c) by Lothar Miedel *
;*****

org ka300

znstrw equ keecf ;Zeilen-Nummern-String wandeln 464=ee04 664=eed4
zips equ ke864 ;Zeile im Programm suchen 464=e7a3 664=e869
zipe equ ke7a5 ;Zeile im Programm einfuegen 464=e6d2 664=e7aa
edrst equ kac02 ;Editor-Restart 464=aa98 664=ac02
znaw equ ke254 ;Zeile nach ASCII wandeln 464=e163 664=e259
edit equ kbd5e ;Editor in Sprungtabelle 464=bd3a 664=bd5b
spflag equ kac00 ;Spaceflag (00 = Spaces werden uebernommen)
log equ kbcd1 ;RSXen einbinden
zeiout equ kbb5a ;Zeichenausgabe
null equ k0000

;RSXen einbinden *****
A300 01 0A A3 init ld bc,jptab ;Zeiger auf Sprungtabelle
A303 21 19 A3 ld hl,kram ;Zeiger auf Hilfsspeicher
A306 CD D1 BC call log ;RSX-Erweiterung einbinden
A309 C9 ret ;Ruecksprung nach Basic

A30A 12 A3 jptab defw rsxbef
A30C C3 C3 A3 jp ein
A30F C3 D4 A3 jp aus

A312 45 49 rsxbef defb "EI"
A314 CE defb "N"+&00
A315 41 55 defb "AU"
A317 D3 defb "S"+&00
A318 00 defb &00

A319 (0004) kram defs 4

;*****
A31D CD BF A3 start call rstrt ;Eigentlicher Beginn

A320 F5 push af ;Registerinhalte
A321 C5 push bc ;retten
A322 D5 push de
A323 E5 push hl

A324 2A EF A3 ld hl,(buf) ;Buffer-Adresse holen
A327 28 6A jr z,ferti ;keine Eingabe

A329 E5 push hl ;hl retten
A32A CD 98 A3 call zeile ;Zeile suchen
A32D E1 pop hl ;HL wieder holen
A32E 30 63 jr nc,ferti ;Fehler oder Befehl

A330 CD CF EE call znstrw ;in DE steht nun die Zeilen-Nummer
A333 ED 53 EB A3 ld (znr),de ;Zeilennummer merken
A337 CD 64 E8 call zips ;Zeilenlaenge steht danach in C
;in HL steht der Zeilenbeginn
A33A 22 E5 A3 ld (zeisad),hl ;Zeilenbeginn (SA)
A33D ED 43 E7 A3 ld (zeilen),bc ;Laenge in C
A341 09 add hl,bc ;in HL nun das Zeilenende
;d. h. der naechste Zeilenbeginn

A346 0F rrca
A347 E6 0F and &0f ;Bit 7-4 ausblenden
A349 CD B3 A3 call umsetz
A34C 7B ld a,e
A34D E6 0F and &0f ;Bit 7-4 ausblenden
A34F CD B3 A3 call umsetz
A3B2 C9 ret

A3B3 FE 0A umsetz cp &0a
A3B5 38 02 jr c,ziffer
A3B7 C6 07 add a,7
A3B9 C6 30 ziffer add a,48
A3BB CD 5A BB call zeiout
A3BE C9 ret

A3BF CF 02 AC rstrt rst 1,edrst

```

```

A3C2 F7          brk          ;Nur ein Trennzeichen

A3C3 3E FF      ein  ld  a,&ff  ;keine Spaces
A3C5 32 00 AC   ld  (spflag),a ;mehr zulassen
A3C8 01 03 00  ld  bc,&0003 ;drei Bytes
A3CB 21 F4 A3   ld  hl,neu ;von Tabelle NEU
A3CE 11 5E BD   ld  de,edit ;nach edit
A3D1 ED 00      ldir         ;kopieren
A3D3 C9         ret          ;fertig

A3D4 3E 00      aus  ld  a,&00  ;Spaces
A3D6 32 00 AC   ld  (spflag),a ;wieder zulassen
A3D9 01 03 00  ld  bc,&0003 ;drei Bytes
A3DC 21 F1 A3   ld  hl,alt ;von Tabelle ALT
A3DF 11 5E BD   ld  de,edit ;nach edit
A3E2 ED 00      ldir         ;kopieren
A3E4 C9         ret          ;fertig

A3E5 (0002)     zeisad defs 2      ;Zeilenstartadresse
A3E7 (0002)     zeilen defs 2     ;Zeilen-Laenge
A3E9 (0002)     zeadr  defs 2     ;Zeilen-Endadresse
A3EB (0002)     znr     defs 2     ;Zeilen-Nummer
A3ED (0002)     psun   defs 2     ;Pruefsumme

A3EF 8A AC      buf  defb  &8a,&ac

A3F1 CF 02 AC   alt  rst  1,drst
A3F4 C3 1D A3   neu  jp   start
A3F7 (A3F7)     end

Errors: 00000 Warnings: 00000

```

Dadurch werden vom CPC keine führenden Spaces mehr angenommen. Leider werden in Programmen enthaltene Leerzeichen dadurch aber nicht nachträglich entfernt. Deshalb müssen wir ein Programm „neu eingeben“. Dies wollen wir aber nicht von Hand durchführen, sondern das soll vom Programm gleich miterledigt werden.

Wir haben uns dazu einen kleinen Trick ausgedacht: Wenn wir ein Programm als ASCII-FILE abspeichern (SAVE „Programm-Name“,A), dann werden führende Spaces zwar abgespeichert, beim Einlesen durch unser Programm aber nicht mehr berücksichtigt, da in &AC00 keine „0“ mehr steht! Damit verschwinden also überflüssige Leerzeichen. Die Zeilen bis 180 sollten eigentlich keine Verständnisschwierigkeiten bereiten, deshalb übergehen wir diese einfach. In Zeile 190 werden dann

das „Schreib-“ und das „Lesefile“ geöffnet.

Ab Zeile 200 geht es richtig los: Falls das File-Ende noch nicht erreicht wurde, wird eine Basic-Zeile durch LINE-INPUT #9,a\$ in die Variable a\$ eingelesen. In Zeile 210 wird die Länge dieser Basiczeile festgestellt. In 220 wird abgefragt, ob das letzte Zeichen der Zeile ein Space ist (durch das vorherige Abspeichern als ASCII-File ist nun eine leichte Unterscheidung zwischen dem Wert &20xx und echten Spaces möglich!).

Deshalb wird in dieser Zeile dann, wenn der eingelesene String eines der unerwünschten Zeichen am Ende enthält, die Stringlänge um eins verringert. Da es aber sein kann, daß mehrere Leerzeichen am Ende vorhanden sind, wird, wenn ein „Endspace“ gefunden wurde, durch Zeile 230 gleich wieder zu Zeile 210 gesprungen.

Wurde das Programm

in der eingelesenen Zeile nicht (oder nicht mehr) fündig, dann geht es in Zeile 240 weiter. Durch diese wird bewirkt, daß die eingelesene und evtl. geänderte Zeile in das in Zeile 190 geöffnete File geschrieben wird. Nach diesem Vorgang wird mit Zeile 200 weitergemacht. Ergibt die Abfrage in 200, daß das File-Ende erreicht wurde, dann geht es mit Zeile 250 weiter. Dort werden die beiden – noch offenen – Files geschlossen. Danach erfolgen kurze Mitteilungen auf dem Bildschirm (Zeilen 260 bis 290), und durch Zeile 300 wird das bereinigte File komplett in den Speicher des CPC eingelesen. Nun kann das Programm ganz normal abgespeichert werden, überflüssige Leerzeichen sind nicht mehr vorhanden. Erst jetzt ist es sinnvoll, das Programm, mit Prüfsummen versehen, auszudrucken!

EINE REINE ADDITION GENÜGT NICHT!

Zurück zu den grundlegenden Überlegungen: Ganz sicher ist es nicht zweckmäßig, eine Prüfsumme nur aus der Additionsfolge der einzelnen Bytes einer Zeile zu bilden, denn sonst würden beispielsweise

10 a=1234

und

10 a=4123

die gleiche Prüfsumme ergeben! Da diese beiden Wertzuweisungen aber vollkommen unterschiedlich sind, könnte ein Programm – bei einer Wertvertauschung – trotz der als korrekt gemeldeten Prüfsummen niemals richtig laufen!

Aus diesem Grunde

muß eine Verschlüsselung der jeweiligen Zeilenprüfsumme erfolgen, doch davon später mehr.

Nun zu den Überlegungen, wie denn ein Checksummer überhaupt aufgebaut werden kann. Dabei müssen zwei Dinge gleichzeitig berücksichtigt werden: Erstes müssen beim Ausdruck die Prüfsummen ausgegeben werden und zweitens müssen diese auch bei Ihnen (also dem Leser) auf dem Bildschirm erscheinen, wenn eine Zeile eingegeben wurde.

Da Programmzeilen nach der Eingabe aber „tokenisiert“ im Speicher stehen, dürfen die Checksummen nicht aus einer ASCII-Zeichenkette berechnet werden, denn sonst ergäbe sich z.B. für den Befehl PRINT eine andere Prüfsumme, wenn dieser Befehl in der Kurzform (=Fragezeichen) eingegeben und nicht ausgeschrieben wird!

WOHER HOLEN WIR DIE INFORMATIONEN FÜR DIE PRÜFSUMME?

Alle CPCs haben mehrere Speicherbereiche, in denen bei Eingaben Informationen abgeholt werden können. Einmal gibt es den sog. Tastaturbuffer-Bereich, in welchem die Anzahl der Zeichen, der Beginn der eingegebenen Zeichen und weitere Informationen stehen. Dieser befindet sich bei den verschiedenen CPC-Typen aber an unterschiedlichen Stellen, und deshalb müßte ein Checksummer für die drei CPCs in dieser Beziehung angepaßt werden. Da dieser Buffer aber sowieso nicht die Abholung aller Informationen zuläßt und wir auch so noch genügend

Anpassungen für die verschiedenen CPCs durchführen müssen, kommt dieser nicht in Frage!

Als nächstes käme der Buffer für Eingabe und List in Betracht, aber auch dieser befindet sich bei den verschiedenen CPCs an unterschiedlichen Adressen, und außerdem liegen die Eingaben dort nicht in tokenisierter Form vor.

Doch es gibt einen Speicherbereich, der bei einer Eingabe alles so enthält, wie wir es eigentlich wünschen. Er beginnt ab Speicheradresse &40, liegt also unterhalb eines Basic-Programmes. Alle mit ENTER abgeschlossenen Eingaben stehen dort zur Verfügung und zwar bereits in tokenisierter Form; also so, wie sie in ein Programm eingefügt werden. Eine Information fehlt allerdings: die Zeilennummer! Diese bei der Prüfsummenberechnung einzubeziehen, ist eigentlich nicht sehr kompliziert und deshalb wundern wir uns, daß sie bei den uns bekannten Checksummen nicht berücksichtigt wird. Auch falsche Zeilen-Nummern können zu einem Fehlverhalten des Programms führen!

JETZT WIRD'S ERNST!

Wie können wir erreichen, daß nach jeder Eingabe einer Programm-Zeile eine Prüfsumme auf dem Bildschirm ausgegeben wird? Hierzu muß programmtechnisch in den normalen Betriebssystem-Ablauf des CPCs eingegriffen werden. Dies könnte interrupt-gesteuert geschehen, aber es geht viel einfacher:

Bei den CPCs sind sogenannte „Sprung-Tabel-

len“ vorhanden, über welche die wichtigsten Betriebssystem-Routinen aufgerufen werden können.

Eine wichtige Sprungadresse steht bei &BD5E. Der CPC6128 enthält dort den Befehl: RST 1,&AC02.

Durch diesen Befehl wird eine Routine im unteren ROM angesprungen und zwar die Editor-Routine bei &2C02. Beim CPC464 steht dort allerdings ein anderer Restartbefehl, denn die CPCs sind ja nicht gleich! Wir müssen aber gar nicht wissen, was ab &BD5E steht und brauchen uns an dieser Stelle deshalb auch gar nicht weiter darum zu kümmern. Was wir aber tun wollen, ist, die Werte an dieser Stelle so umzuschreiben, daß immer dann, wenn der Editorteil über diesen Einsprung ab &BD5E aufgerufen wird, auf eine von uns geschriebene Routine verzweigt und erst dann der ursprüngliche Restart ausgeführt wird.

ERKLÄRUNGEN AM BEISPIEL CPC6128

Damit wir nicht zu theoretisch werden, wollen wir nach diesen Vorinformationen nun gleich auf das Programm des für Sie entwickelten Checksummers eingehen. Betrachten Sie hierzu bitte das Assemblerlisting.

Nach dem ORG-Befehl erfolgt eine Zuweisung verschiedenster Adressen zu den zugehörigen Labels und kurzen Mitteilungen, für welche Funktionen sie zuständig sind. Die Zuweisungen erfolgen im Beispiel für den CPC6128. Die Werte für die anderen CPC-Typen sind ebenfalls angegeben.

Nach den Definitionen der Labels kommt der Abschnitt für die RSX-Einbindung von zusätzlichen Befehlen. Dieser Teil wurde schon mehrfach von uns beschrieben und deswegen diesmal übergangen. Betrachten Sie bitte gleich den Teil, der als Label die Bezeichnung „ein“ besitzt (Adresse &A3C3!). Als erstes wird dort in den Akku der Wert &FF geladen und dann in die Speicherstelle „spflag“ gebracht. Durch diese Manipulation werden die nun schon häufig erwähnten überflüssigen Spaces unterdrückt.

Als nächstes wird das BC-Doppelregister mit dem Wert „003“ geladen. Diese Zahl entspricht auch dem dezimalen Wert drei. Dann wird das HL-Doppelregister mit der Adresse einer „Tabelle“ geladen. Diese Tabelle (neu) enthält den Drei-Byte-Befehl JP Start. Der Wert von „start“ ist die Adresse &A31D und liegt in unserem eigenen Programm. Anschließend erhält das DE-Register die Mitteilung der Adresse, an welcher der bisherige Editor-Einsprung liegt. Dieser ist beim CPC6128/664 &AC02, beim 464 &AA98. Dort befindet sich normalerweise der schon erwähnte RST-Befehl. Durch den Befehl LDIR wird ein Speicherbereich umkopiert, dessen Herkunft (Quelladresse) in HL und dessen Zieladresse in DE steht.

Die Übertragung wird byteweise so oft durchgeführt, wie der Wert in BC angibt! Nach der Befehlsausführung steht — anstelle des ehemaligen RST-Aufrufes — nun ein Sprung auf unser eigenes Programm.

Betrachten wir gleich noch die gegenteilige

Routine, die bewirkt, daß der ursprüngliche Zustand wieder hergestellt wird. Zu finden ist dieser Programmteil beim Label „aus“. Als erstes wird durch die ersten beiden Befehlsfolgen erreicht, daß „überflüssige“ Leerzeichen wieder erlaubt sind. Dann erfolgt — ebenso wie vorher — das Umkopieren von drei Bytes. Diese Programmteil dürfte eigentlich nun nicht mehr schwierig zu verstehen sein; deswegen gleich wieder weiter.

DIE VORHANDENE EDITOR-ROUTINE WIRD AUCH VON UNS GENUTZT!

Das eigentliche Checker-Programm beginnt beim Label „start“. Dort steht CALL RSTRT, also der Befehl, die Routine bei Adresse &A3BF aufzurufen. Dort finden Sie den ursprünglichen Restart-Befehl, der in die Editor-ROM-Routine (im Falle des CPC6128 und 664 bei &2C02 beim 464 bei &2A98) einspringt. Diese Firmware-Routine nun näher zu beschreiben, wäre in diesem Artikel zu umfangreich. Für Sie wichtig ist aber, daß beim Verlassen dieser Routine an die Adresse &A320 (nach dem Befehl CALL RSTRT) zurückgekehrt wird! Dadurch werden dann die dort stehenden Befehlsfolgen abgearbeitet.

Durch die vier PUSH-Befehle werden alle Register-Inhalte zunächst einmal „gerettet“, denn durch den Programmablauf verändern sich diese, und manche Werte benötigt das Programm später nochmal deshalb: „Sicher ist sicher“. Die erste Änderung ist nun schon, daß das HL-Register mit den

beiden Werten, die ab „buf“ stehen, geladen wird. Dies hätten wir auch nach dem nun folgenden Befehl tun können! Der Befehl jr z, ferti stellt fest, ob eine Programmzeileneingabe erfolgte. Ist dies nicht der Fall (die Zeroflagge ist gesetzt), dann muß noch der Stack geleert (erfolgt ab der Routine „ferti“) und aus dem Maschinenprogramm ausgelesen werden.

Anders ist es, wenn eine Basiczeile eingegeben wurde, dann geht es bei &A329 weiter. Der Wert aus „buf“, nämlich &AC8A, steht noch im HL-Register. &AC8A ist der Eingabebuffer, in welchem die gemachten Eingaben stehen. Nun wird es etwas kompliziert, denn &A32A bis &A345 dienen dazu, die Parameter für die eingegebene Zeile zu holen, bei einem Eingabefehler den Programmablauf zu beenden usw. Dabei werden verschiedene Firmware-Routinen benutzt, die aber recht umfangreich sind und – ohne kommentiertes ROM-Listing – von Ihnen kaum nachvollzogen werden können. Wie Sie aber dem Assemblerlisting entnehmen können, erhalten wir über diese Routinen die Zeilennummer, den Zeilenbeginn und die Zeilenlänge.

Falls Sie im Assemblerlisting feststellen sollten, daß manche Befehle für den Checksummer eigentlich überflüssig sind oder auch besser programmiert werden könnten, dann denken Sie bitte daran, daß der Checksummer nur ein Teil der Prüfungsangelegenheit ist. Der andere Teil ist der Drucksummer, also der Teil, den wir zur Erzeugung der mit Prüfsummen versehenen Listings ein-

setzen. Damit wir bei eventuellen, zukünftigen, Verbesserungen nicht zuviel Arbeitsaufwand haben, sind wichtige Programmteile in beiden Programmen identisch, auch wenn sie in einem der beiden gar nicht benötigt werden. Der Aufwand bei uns ist dann wesentlich geringer und die paar Bytes, die beim Checksummer gespart werden könnten, rechtfertigen es nicht, daß wir den Checker optimieren!

Machen wir bei &A345 weiter: Dort wird die Prüfsumme auf Null gesetzt, damit die Prüfsumme einer vorher eingegebenen Zeile keinen Fehler bewirkt. Ab &A34B wird die Zeilenlänge zuerst in den Akku geladen und dann in der Adresse psum+1 „gemerkt“.

Der nachfolgende Befehl „sub &05“ subtrahiert von der Zeilenlänge den Wert fünf. Wie Sie wissen, wird eine Basiczeile folgendermaßen im Speicher abgelegt: zwei Bytes für die Zeilenlänge, zwei Bytes für die Zeilennummer und dann folgen die Statements. Der Offset zu den Statements wäre also in Wirklichkeit der Wert 4 und nicht 5! „Programmbedingt“ aber, weil die Speicherstelle &40 für den Zähler im B-Register den Wert 9 hat, müssen wir fünf abziehen! „ld ix,&40“ bewirkt, daß das IX-Register als Zeiger auf den Beginn des Tokenbuffers gestellt wird.

NUN BEGINNT DIE PRÜFSUMMENBERECHNUNG

Bei der Routine „addi“ wird als erstes durch den Befehl „or a“ die Carry-Flag gelöscht. Danach wird durch den Befehl „ld a,(ix+00)“ der erste

Wert aus dem Tokenbuffer geholt und durch „ld e,a“ in das E-Register übergeben. Danach wird in den Akku ein Teil der bisherigen Prüfsumme geladen „ld a (psum)“ und der im C-Register stehende Wert addiert (add a,e). „or a“ löscht wieder die Carry-Flag und dann folgt die Codierung mittels „rlca“. Nun müssen wir auf unser letztes Heft verweisen, dort haben wir sehr ausführlich gezeigt, was es mit den Rotier- und Schiebebefehlen auf sich hat. Hier nur noch der Hinweis, daß durch diesen Befehl gewährleistet wird, daß „Verdreher“, wie z.B. a=1234 anstelle von a=4321, erkannt werden, denn durch „ld (psum),a“ wird der codierte Wert als Prüfsumme zwischengespeichert!

Anschließend wird das IX-Register um eins erhöht, und der Befehl „djnz addi“ erniedrigt das B-Register und verzweigt immer dann wieder nach „addi“, solange der Inhalt des B-Registers größer als 0 ist, d.h., bis die gesamte eingegebene Zeile „abgearbeitet“ ist.

Die Unterroutine „znadd“ (Zeilen-Nummer-Addition) addiert zur bisherigen Prüfsumme, die noch im Akku steht, den Low- und Highbyte-Wert der Zeilennummer, und danach steht die komplette Prüfsumme in den Speicherstellen „psum“ und „psum+1“. Genauer gesagt, steht in „psum+1“ die Länge der Basiczeile und in „psum“ die eigentliche Prüfsumme.

DIE PRÜFSUMME WIRD AUSGEGEBEN

Ab &A376 erfolgt nun die Ausgabe der Prüfsumme. Diesen Teil dürften

Sie nun, wenn Sie unsere letzten Artikel gelesen haben, ohne Probleme verstehen. Das gleiche gilt auch für die Wandlungsroutinen zur Ausgabe der Hexwerte und für die Unterroutinen „ein“ und „aus“.

HANDHABUNG DES PROGRAMMES

Den Lader für den Checksummer finden Sie in allen Heften. Nach dem Lauf des Laders stehen die RSX-Befehle EIN und AUS zur Verfügung. Logischerweise bewirkt der Befehl EIN die Aktivierung und AUS die Abschaltung. Den Ladeteil selbst können Sie dann löschen und mit der Eingabe des Programmes beginnen. Nach jeder eingegebenen Zeile wird Ihnen eine Prüfsumme ausgegeben, wobei die beiden ersten Zeichen der Länge der Zeile entsprechen. Aufgrund dieser Zahl können Sie also feststellen, ob bei einem Fehler zu viel oder zu wenig eingegeben wurde! Die beiden anderen Zeichen sind der „verschlüsselte“ Wert der Zeilennummer und des Zeileninhaltes.

Nach der Eingabe eines Programmes mittels Checksummer-Überwachung sollte diese abgespeichert und dann der CPC zurückgesetzt werden. Der Grund ist: Die Ein- und Abschaltung des Checksummers erfolgt mit sog. RSX-Befehlen. Diese RSX-Befehle und das Programm des Checksummers selbst aber liegen irgendwo im RAM-Speicher des CPCs. Falls nun aber ein Programm ebenfalls diesen Speicherplatz beansprucht oder gleichlautende RSX-Befehle beinhaltet, muß es zu einem Programmab-

Jetzt schlägt's Dreizehn!

CPC-WELT
kommt direkt zu
Ihnen ins Haus!
13x pro Jahr!
Für nur 60 DM!

Sie sparen fast 30 Mark!

Nr. 2/88-Februar

CPC-WELT

SCHNEIDER

CPC-WELT

DM6/ÖS/48/SFR6

SCHNEIDER

BTX-Modul
Neue Zukunft für den CPC

Im Test:
Drucker LQ 3500

Neu dabei:
C-Splitter

Neue Spiele
im Test:
*Basil
Firetrap
Trantor
und viele andere*

Neue Anwendung für den CPC



SCHNEIDER AKTIV-SOFTWAREJAHRBUCH 88

schneider
aktiv
special

Nr. 1/88
DM 14,80
ÖS 124
SFR 14,80

CPC
SOFT-
WARE
JAHR-
BUCH
1988

Die schönsten
Spiele- Die
besten
Anwender-
Programme
Hilfreiche



Finden Sie Ihre CPC-WELT nicht immer am Kiosk? Vielleicht, weil schon ausverkauft? Möchten Sie die CPC-WELT schon vor der Kioskbelieferung in Händen haben? Dann gibt es jetzt die Möglichkeit! Wir beliefern Sie im Abonnement mit zwölf plus einer Ausgabe für ganze 60 DM (Inland) oder 80 DM (Ausland). Sie erhalten dann das jeweils druckfrische Heft, in der Regel sogar früher, als es am Kiosk hängt (so die Bundespost will). Zwölfmal. Und außerdem gehört zum Abo noch unser jährlicher CPC-SPECIAL Sammelband im Wert von DM 14,80. Einzige Bedingung: Das Abo muß zum Zeitpunkt des Erscheinens dieses Bandes noch bestehen und bezahlt sein. Ist das ein Angebot?

WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Zugang der Abo-Bestätigung durch den Verlag an Sie widerrufen. Postkarte genügt! Bitte, bestätigen Sie durch Ihre zweite Unterschrift, daß Sie von diesem Widerspruchsrecht Kenntnis genommen haben.

ACHTUNG: WICHTIGER HINWEIS!

Sie können dieses Abonnement jeweils mit einer Frist von einem Monat zum Ende des Bezugszeitraumes (zwölf Hefte) kündigen. Unterlassen Sie diese Kündigung, wird die Belieferung mit weiteren zwölf Heften zu den gleichen Bedingungen fortgesetzt! Die Lieferung beginnt nach Eingang der Abo-Gebühr.

ABO-SERVICE-KARTE

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist!

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten zwölf

Ausgaben an untenstehende Anschrift. Wenn ich nicht vier Wochen vor Ablauf kündige, läuft diese Abmachung automatisch weiter.

Name _____

Vorname _____

Straße/Hausnr. _____

PLZ/Ort _____

Ich bezahle:

per beiliegendem Verrechnungsscheck

gegen Rechnung

bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort _____

Kontonummer _____

Bankleitzahl _____

(steht auf jedem Kontoauszug)

**SCHNEIDER CPC-WELT
ABO-SERVICE 2/88
POSTFACH 1161
D-8044
UNTERSCHLEISSHEIM**

Unterschrift _____

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____ 2/88



PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen die Listings dieses Heftes auf

Kassette 30 DM Diskette incl. Bonus-Programm DM 30

Ich zahle: (Zutreffendes bitte ankreuzen!)
per beigefügten Scheck Schein

Gegen Bankabbuchung am Versandtag

Meine Bank (mit Ortsname) _____

Meine Kontonummer _____

Meine Bankleitzahl _____

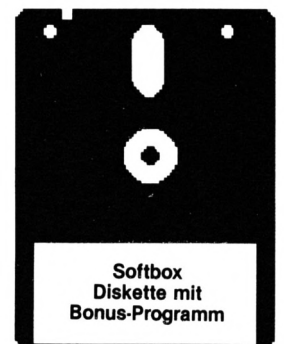
Vorname _____

Str./Nr. _____

(steht auf jedem Bankauszug)

Nachname _____

PLZ/Ort _____ 2/88

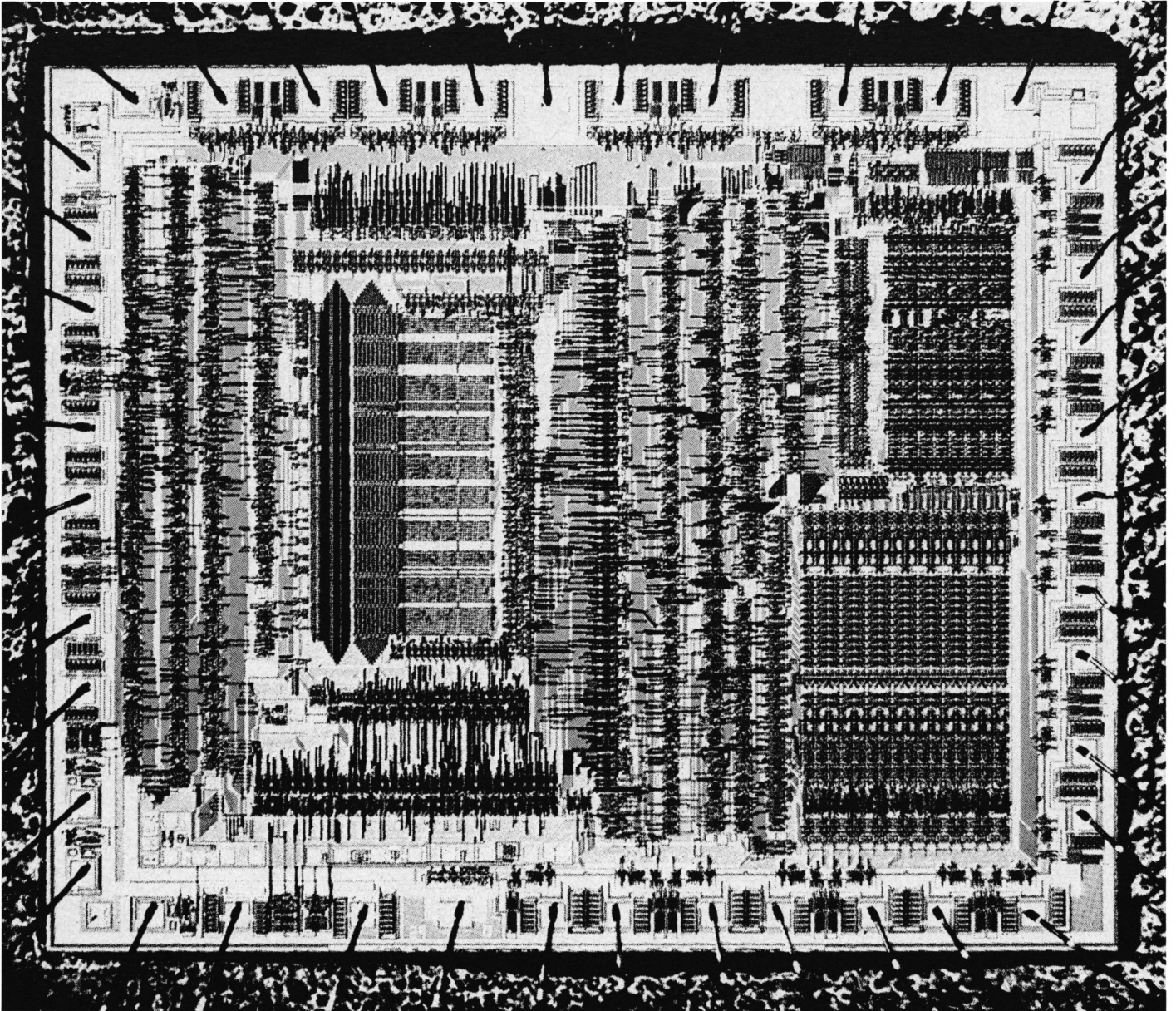


Verkaufsbedingungen: Lieferung nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme. Umtausch bei Nichtfunktionieren gegen gleiches Programm. Keine Rücknahme, da offene Software.

Unterschrift _____

Bitte ausschneiden und einsenden an

**SCHNEIDER CPC WELT
SOFTWARE-SERVICE 2/88
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM**



sturz kommen, deshalb nach der Eingabe also erst einmal abspeichern.

Benutzt das Programm keine gleichlautenden RSX-Befehle und sind auch keine Maschinen-Spracheteile für das Programm vorgesehen o.ä., dann können Sie das Programm aber auch ohne Rücksetzung des CPC ablaufen lassen.

WENN SCHON, DENN SCHON!

Konsequenterweise müssen alle Zeilen so einge-

geben werden, wie sie abgedruckt sind. Bei Zeilen mit REM am Ende dürfen keine Leerzeichen zusätzlich eingetippt werden. Gleiches gilt für Zeilen, deren letztes Statement der Befehl

Print"text

ohne schließendes Anführungszeichen ist! Der Befehl PRINT darf auch als Fragezeichen eingegeben werden. Wurde der am Ende einer Zeile stehende Text-PRINT-Befehl mit einem Anführungszeichen korrekt abgeschlossen, dann werden nachfolgen-

de Leerzeichen nicht mehr berücksichtigt!

Die Editor-Routinen vom CPC464 und vom CPC664/6128 sind sehr unterschiedlich. Die automatische Zeilenvorgabe durch den Befehl AUTO gibt für bestehende Zeilen beim 464 nur ein Sternchen aus, bei den beiden anderen Typen hingegen den Zeileninhalt. Leider haben wir dieses Problem noch nicht lösen können, d.h. für CPC464-Besitzer: Keine Programmeingabe unter Checksummer-Überwachung mittels AUTO-

Befehl! Ansonsten viel Erfolg und fehlerfreie Programmeingaben mittels unseres CHECKER.

VORSCHAU

Beim nächsten Mal wenden wir uns der Diskettenstation zu. Dabei geht es dann wieder um kleinere Routinen, die sehr detailliert besprochen werden, und sobald wir wieder die „Grundlagen“ besprochen bzw. erarbeitet haben, kommt wieder ein „dicker Brocken“ in Form eines umfangreichen Programmes. *LM*

Mit fremden Federn

Seit einiger Zeit ist es schon abzusehen, daß die 24-Nadel-Drucker die magische Grenze von 1000,- DM unterschreiten. Schneider war nicht einmal der erste, der sich diese Hürde vornahm, konnte aber die Konkurrenz mit einem besonders preisgünstigen Modell erschrecken. Mit nur 898,- DM ist der Kunde dabei. Ob und welche Abstriche bei dieser Kalkulation gemacht werden müssen, haben wir getestet.

Zuallererst muß gesagt werden, daß dieser Drucker aus dem Hause Amstrad, dem Partner von Schneider, stammt. Wie offensichtlich dies zum Testzeitpunkt gehandhabt wurde, war eine der Überraschungen: Der Pappkarton, in dem das Gerät angeliefert wurde, trug das Amstrad-Emblem. Was fehlte, war das Handbuch. Dies kann man sich durchaus erklären, denn warum sollte man in England eine englische Bedienungsanleitung dazupacken, wenn es doch Aufgabe des Distributoren ist, für eine deutsche Übersetzung zu sorgen? Lediglich der Aufkleber wies auf den Vertriebspartner aus Türkheim hin. Wie man aus der Lieferung schließen mußte, wurde aber auch das Etikett in Großbritannien angebracht. Im Inneren herrschte der Name „Amstrad“ vor. Die Platine bewies, daß es sich um eine englische Entwicklung handelt, bei der allerdings auf fernöstliche Bauteile nicht verzichtet wurde. So unwesentlich, wie es auf den ersten Blick scheint, ist diese Tatsache nicht. Haushalte, bzw. kleine Büros, die bereits einige Schneider-Drucker benutzen und sich nun den 24-Nadel-Drucker zusätzlich kaufen, werden sicherlich herb enttäuscht sein: wieder ein anderes Farbband gegenüber den DMP-Modellen, eben eine Amstrad-Norm. Den Normalverbraucher (er hat nur einen Drucker) mag es weniger stören. Vielleicht ist er sogar froh, daß er der japanischen Übermacht bei den Druckern endlich einmal ein (wenigstens) europäisches Modell entgegengesetzt werden kann.

KOMPAKTES GERÄT MIT EINFACHER KONSTRUKTION DES ENDLOSTRAKTORS

Die Größe des Gerätes bleibt im üblichen Rahmen. Trotz Traktor ist der Printer nur zwölf cm hoch und

paßt auch dann noch in ein Regal, wenn die Papierhalterung auf Einzelblätter gestellt ist. Weniger wichtig, aber auch noch kompakt zu nennen, sind die 42 cm Breite und 32 cm Tiefe. Allerdings müssen bei der Platzvergabe ohnehin der Stecker des Parallelanschlusses und das Stromkabel berücksichtigt werden. Wieder einmal wurde der alten Unsitte gefrönt, sämtliche Anschlüsse nach hinten zu verlegen. Wie sehr dies den Papiertransport behindern kann, weiß anscheinend nur ein Anwender, für Konstrukteure bleibt es ein Geheimnis.

Ebenso unsinnig ist die Bauart des Traktors. Er wird beim Kauf gleich mitgeliefert und einfach nur bei Bedarf aufgesteckt. So weit, so gut. Warum es aber unbedingt ein Zugtraktor sein muß, verrät niemand. Auf die Dauer kommt diese Version teurer, da ein neues Blatt immer erst ab der siebten oder achten Zeile bedruckt werden kann. Um einen sauberen Zeilenvorschub zu erreichen, können Sie erst mit Blatt zwei beginnen. Überhaupt macht der Traktor nicht den solidesten Eindruck. Die Verbindung zum Antrieb wird über kleine Plastikzahnäder hergestellt, denen man keine lange Lebensdauer zutraut. Unser Test rief zwar keinen Schaden hervor, es bleibt jedoch fraglich, wie oft sich der Traktor unbesorgt abnehmen und wieder aufstecken läßt. Überhaupt erfordert der Wechsel zwischen Endlos- und Einzelblatt Zeit und Mühe; mal ganz abgesehen davon, daß ständig irgendwelche Einzelteile herumliegen, die bei Papierwechsel benötigt werden. Sicherlich sind dies Kleinigkeiten, doch Tester, die solches bemerken, sollten nicht gleich als „notorische Nörgler“ abgetan werden. Wie schwer wäre es denn wirklich, derartige Mängel von vornherein zu beseitigen? Diese Frage muß man sich bei Amstrad durchaus einmal gefallen lassen.

Die Papierführung selbst kann leichtgängig verschoben werden. Zu leicht, mag der eine oder andere glauben, doch beim Test ergaben sich keine Schwierigkeiten. Ein mittleres Transportrad stützt das Endlospapier ab, so daß es nicht durchhängt. Es ist also bei Amstrad durchaus machbar, auf kleine Konstruktionsdetails zu achten. Diese machen den Drucker nicht unbedingt teurer, fallen aber positiv ins

PAPIERVERSCHWENDUNG IST VORPROGRAMMIERT

Gewicht. Unter diesem Gesichtspunkt ist auch die (halb)automatische Papiereinführung zu sehen. Zieht man den Friktionshebel ganz nach vorn, dann wird das Blatt bis zur ersten Zeile gezogen. Vielleicht muß man ein wenig nachrichten, die größte Arbeit bleibt jedoch erspart. Dies gilt allerdings nur für die Benutzung eines Einzelblattes. Wie schon erwähnt, muß bei einem Traktor das Papier weiter vorgezogen werden und der Druckkopf liegt an Zeile acht.

Der Schneider LQ 3500 im Matrimodus bei 24 Nadeln

```
ABC abc [\] {}
ABC abc [\] {}
ABC abc [\] {}
ABC abc [\] {}
```

Der Schneider LQ 3500 im LQ-Modus bei 24 Nadeln

```
ABC abc [\] {}
ABC abc [\] {}
ABC abc [\] {}
ABC abc [\] {}
```

Wohl jeder wird den LQ 3500 zuerst einen Selbsttest durchführen lassen. Gleich zwei Funktionen stehen zur Auswahl. Die Matrixschrift wird mit Line Feed, die 24-Nadel-LQ mit Form Feed ausgelöst. Diese Tasten sind jeweils zu drücken, während man den Drucker am Hauptschalter einschaltet. Das Ergebnis ist auf den ersten Blick zufriedenstellend. Der LQ verdient seinen Namen (Letter Quality) durchaus. Von der Druckqualität jedoch später mehr, zunächst soll

die Geräuschentwicklung bemängelt werden. Der Druck ist nicht nur sehr laut, sondern vollzieht sich auch in einer unangenehmen Tonfrequenz, die sensible Anwender an den Zahnarzt erinnert. Eine Filzunterlage – sonst immer die letzte Hilfe – nutzt hier wenig. Der Lärm wird vom Druckkopf erzeugt und nicht durch die Resonanz des Gehäuses.

Kommen wir zu den – mit Recht – unbeliebten DIP-Schaltern. Das Mäuseklavier des LQ 3500 verfügt gleich über zwei Anordnungen zu je acht Schaltern. Sie sind leicht erreichbar an der Rückseite untergebracht und das Gerät muß nicht auseinandergenommen werden. Trotzdem ist eine Pinzette oder eine Büroklammer notwendig, um Veränderungen vorzunehmen. Die Voreinstellungen sind IBM-kompatibel, es muß allerdings noch der Druckerpuffer mit 7 KByte initialisiert werden, sofern man in der Praxis damit arbeiten will. Unverzichtbar ist der deutsche Zeichensatz, den man sich ebenfalls erst einmal per DIP-Schalter auswählen muß. Dies wäre etwas, was man im selben Arbeitsgang mit den Aufklebern erledigen könnte.

BEDIENERFELD MIT VIELEN OPTIONEN

Das Bedienerfeld an der Vorderseite des Druckers erlaubt es übrigens, per Tastendruck den LQ-Modus auszuwählen. Die Tastenbeschriftung dient dabei allerdings eher der Verwirrung. Durch einen Modus-Schalter lassen sich mit den vier Tasten mehrere Optionen herausholen. Die Beschriftung, die der Erklärung dienen soll, ist jedoch nicht ganz eindeutig. Normal ist, daß sich der Drucker ON LINE und OFF LINE stellen läßt, ebenso verhält es sich mit Line Feed und Form Feed. Im täglichen Gebrauch ist es jedoch sehr angenehm, wenn man den Matrix- oder LQ-Modus einstellen, das Papier millimeterweise vorbegeben kann oder auch zeilenweise zurück. Über das Bedienerfeld hat man bei Amstrad also nachgedacht, wengleich man dem Käufer zumutet, sich an die ungewöhnliche Handhabung zu gewöhnen.

24-Nadel-Drucker bringen ein



Im Test: Der Schneider LQ 3500

A
B
C

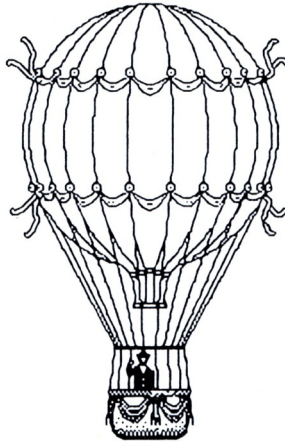


E

f
G

H

I



großes Problem mit sich: Die meiste Software verfügt noch nicht über einen entsprechenden Druckertreiber und die Ausgaberroutine muß mühselig angepaßt werden. Entsprechende Software würde eine Lieferung angenehm vervollkommen. Von Schneider ist allerdings nur ein

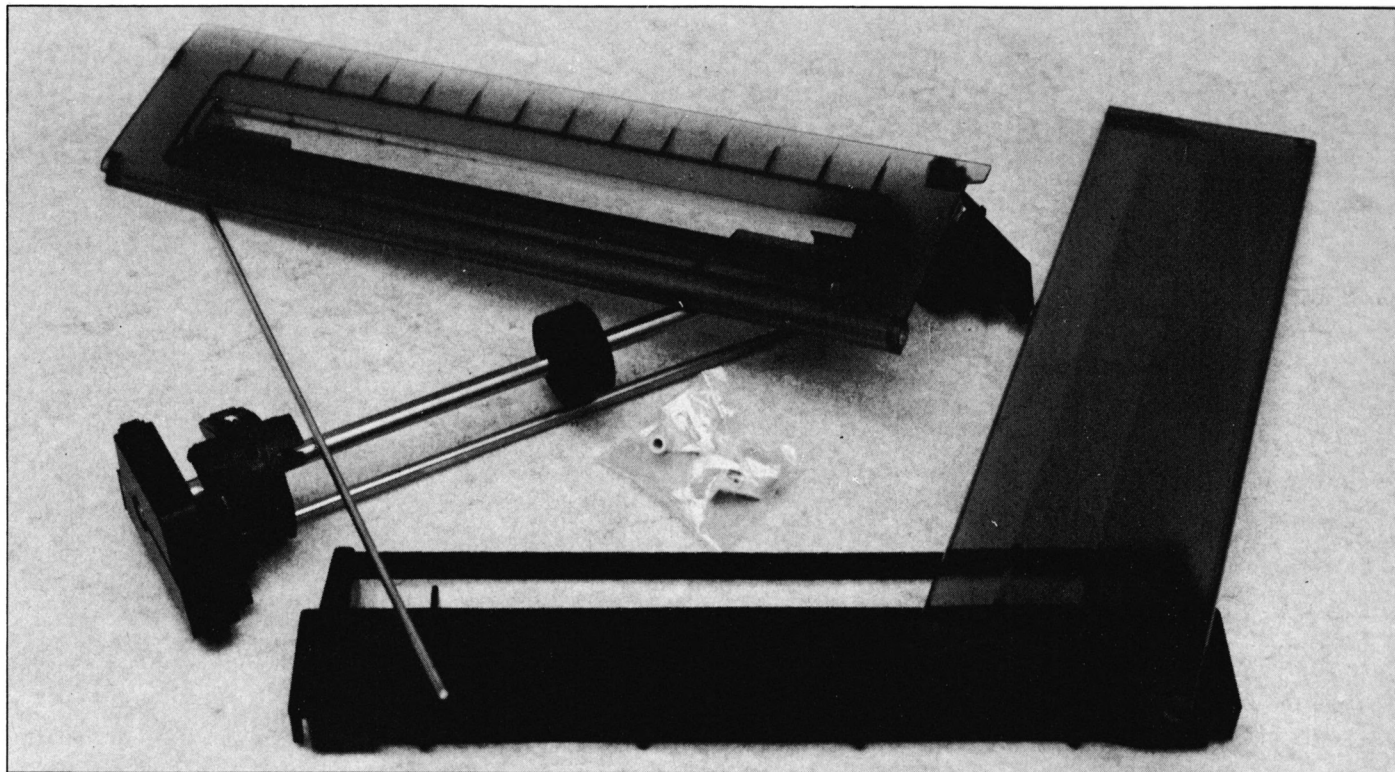
Letter Quality Modus funktioniert ausgezeichnet und das Schriftbild ist exzellent. Die Buchstaben „fransen“ auch bei schlechterem Papier nicht aus, was für den Druckkopf spricht.



Handbuch geplant. Software muß vorhanden sein (im Anwenderprogramm) oder bei anderen Marken (Epson LQ) entliehen werden. Aber es geht auch anders. Bei der Textausgabe sind ohnehin keine großen Probleme zu erwarten, der

FEHLENDER DRUCKERTREIBER

Dabei muß allerdings darauf geachtet werden, daß die Nadeln nicht zu fest andrücken. Je nach Anzahl der Durchschläge (oder verbrauchtem



Wichtige Einzelteile sind dem Drucker als Bastelarbeit beigelegt.

Farbband) kann dies nämlich eingestellt werden. Schwierig wird es mit Grafikausgaben. Wer seine bisherige Hardcopy (etwa unter MS-DOS) benutzen will, der hat nichts von den 24 Nadeln. Eine echte Verbesserung der Druckqualität läßt sich erst mit einem Druckertreiber erreichen. Die Epson LQ Software hilft hier über die ersten Runden. Wenn dem LQ 3500 schon keine Diskette beiliegt, so ist es wenigstens erfreulich, daß man sich bei Amstrad an bestehende Verbreitungsnormen hält.

Das Drucktempo ist abhängig von der Druckqualität. Angegeben sind 160 Zeichen je Sekunde. Wie immer stimmt dieser Wert nicht ganz. Bei einem echten Testbrief wurden lediglich 138 Zeichen gezählt. Es kam allerdings hinzu, daß Teile des Textes kursiv bzw. unterstrichen waren. Die Geschwindigkeit ließe sich also noch geringfügig steigern, sofern man auf Auszeichnungen im Text verzichtet. Der Schneider LQ 3500 muß sich mit diesen Werten sicher nicht verstecken; vielleicht einigt man sich in der Druckerindustrie bald einmal auf einen Normbrief, um mit realen Zahlen werben zu können. Beim Schönschrift-Ausdruck sieht das Ganze schon wieder anders aus. Hier sind es laut Firmenprospekt

„immerhin noch 64“ Zeichen je Sekunde, mit dem Testbrief wurden es gerade noch 45. Da hier der Vergleich zu herkömmlichen 9-Nadel-Druckern schlecht ausfällt, sollte man nicht das einwandfreie Schriftbild vergessen. Zeitmessungen bei der Grafikausgabe wollten wir im Test nicht vornehmen. Erstens sind diese Daten sehr stark vom Druckertreiber abhängig, zweitens konnte der Schneider Drucker mit anderen, um das Vierfache teurere, 24-Nadel-Geräten mithalten.

Der LQ 3500 wird zumindest bei Amstrad als PC-kompatibel gepriesen. Was immer dies bedeuten soll, es braucht die Besitzer eines Acht-Bit-Rechners nicht zu stören. Sämtliche Funktionen sind über Steuerzeichen zu schalten, so daß es möglich ist, den Luxusdrucker auch vom einfachsten Homecomputer aus zu schalten. Das Ergebnis rechtfertigt die Mühe, die man mit der Anpassung vorhandener Druckertreiber hat.

FAZIT: WENIG GELD, EINIGE MÄNGEL, VIEL LEISTUNG

Das erste Kriterium ist der Preis von 898,- DM: Man sollte bedenken, daß dafür selbst im Neun-Nadel-Bereich

kein Drucker für die professionelle Arbeit angeboten wird. Zielgruppe von Amstrad und Schneider – hier dürfte man sich einig sein – sind die privaten Anwender. Diese sind es schon längst gewohnt, für den Niedrigpreis einige Mängel hinzunehmen. Die Lautstärke im unangenehmen Frequenzbereich gehört ebenso dazu wie das umständliche Umrüsten bei Papierwechsel. Wo jedoch Konstruktionsmängel der Grund für Ungemach sind, fängt der Ärger an. Eine sinnvollere Technik beim Endlostraktor und eine bessere Platzierung der parallelen Schnittstelle wären ohne Mehrkosten durch ein bißchen Nachdenken zu realisieren gewesen. Dazu gehört auch ein Druckertreiber, denn es gibt noch zu viel Anwender-Software, die mit diesen Utilities schwach bestückt sind.

Beurteilt man jedoch Qualität und Geschwindigkeit des Ausdrucks, so darf man zufrieden sein. Auch wesentlich teurere Konkurrenzfabrikate sind nicht unbedingt besser. Solange man bei der Nadeltechnik bleibt, ist das Schriftbild nicht mehr zu verbessern. Auch aus dieser Sicht lohnt der Kauf, und der LQ 3500 von Schneider ist deshalb trotz der kleinen Mängel sehr zu empfehlen.

Schneider Experten für Sie

1000 Berlin

SEI NICHT DUMM... KOMM RUM...
DIE GANZE WELT DES SCHNEIDER CPC




Herbert Köcher GbR
**A + C Vertrieb, 1 Berlin 44,
Emser Str. 18**
Tel. 625 36 05, Mailbox 6252098


1000 Berlin

Ihre **COMPUTEREI** C

Wolfgang Hiller



Hardware
Software
Zubehör
Beratung
Literatur



Tempelholer Damm 120
1000 Berlin 42
Am U-Bhf. Tempelhof
Tel.: 7 52 20 91

6090 Rüsselsheim

Computer u. Bürotechnik
Dipl.-Ing. Neuderth

Frankfurter Str. 23/Ecke Friedenspl.

6090 Rüsselsheim
Tel. 06142-68455

DISKETTEN m. Gar.
3" Markendiskette 2 CF
DM 6,19
5 1/4" 2 D 48 tpi DM 0,79
All. Austro-Ag., Schleiß-
heimer Str. 16, 8057 Eching,
Tel. 089/319 54 56

**UMSATZSTEUERVORAN-
MELDUNG.** Schneider CPC
464, 664, 6128, Kassen- und
Girokontobuchführung mit
UST-Voranmeldung. Einfache
Anwendung für Haushalt, Vere-
ine und Einzelunternehmen für DM
70,- + Porto u. Vers. Siegfr.
Bastian, Zusenhofener 15,
7602 Oberkirch 2, Tel. 07805-
39 47

ADRESSENVERWALTUNG
mit Etikettendruck, Kartei-
verwaltung mit Einzeldruck od.
Gesamtdruck aller Adressen.
(z.B. Rundschreiben) Geeignet
für Haushalt, Vereine und Fir-
men.
Preis DM 40,- + Porto u. Vers.
Siegfr. Bastian, Zusenhofener 15
7602 Oberkirch 2
Tel. 07805 / 39 47

VERKAUFE: CPC464, Vor-
tex Sp.-Erw. 256K, DDI1,
Joystick, orig. Wordstar 3.0
m. Handbuch f. 1.100,-
DM. Thomas Sommerfeld,
Bergach 4, 7614 Gengenbach,
Tel. 07803/2839

Tausche Spiele auf Tape.
Habe gute Software. 100 %
Rückschreibgarantie. Listen
an: Michael Schäfer, Rhein-
mass-Str. 3, 5100 Aachen

Verk. Bücher zum halben
Preis: Mein Schneider CPC
(Sybex) 20,- DM; Graphik
u. Sound CPC 464 (DB)
20,- DM; Progr. in Maschi-
nespr. (M+T) 23,- DM; Der
Schneider CPC 6128 (M+T)
23,- DM. Alles zus. 79,-
DM. D. Hamm, Eschenreid-
str. 19, 7032 Sindelfingen

* Public-Domain-User-Grup-
pe * CP/M-Software auf 3"-
Disketten f. Joyce u. alle
CPCs zu einem geringen Un-
kostenbeitrag. Etwa 2000
dokumentierte Programme
stehen zur freien Auswahl.
Katalogdisk 12,- DM. Info
1,60 DM Porto. PDUG, Post-
fach 1118, 6464 Linsenge-
richt

Verk. IBM/PC-kompatible
Spiele!! Es gibt Winter Gam-
es, Summer Games II u.
Leaderboard. Interessenten
schreiben an: Harald Strecker,
Goethestr. 98, 7182 Gera-
bronn. Beeilung, denn es sind
nur 3 Spiele zu je 50,- DM.

THE VERY BEST FOR CPC
Auf 3" u. 5.25" bietet Euch
Michael Deni - Oberthal 4 -
8944 Grönenbach - Service:
08334/1513. Neu im Sorti-
ment: Anwend. spez. f. d. ge-
werbl. Bereich - jede Menge
Public Domain - neueste
Spiele - fordert Listen an!

AMX-Pagemaker, dBase II f.
CPC 6128 8orig.) zu verkauf-
en. Preis VS. Auch möglich
ist Tausch v. Green-Mon. u.
Druckerständer gg. Farbmon.
od. andere Hardware. Info
unter Tel. 0631/49526 ab
19 Uhr

Tausche u. verkaufe Spiele u.
Anwend.-Prog. f. CPC 6128,
A/D. Listen od. bespielte
C/D an: Peter Neumann, Gei-
selbachstr. 23, 8300 Esslin-
gen. 100 % Antwort! C/D neu
bespielt zurück!

LICHTGRIFFEL
mit Programm für CPC
464 oder CPC 664 + CPC
6128 nur 49,- DM. Ver-
sand gegen Scheck/Nach-
nahme. Info gratis!
Fa. Schießbauer, Postfach
1171S, 8458 Sulzbach,
Tel. 09661/6592 bis 21h

+++ VERKAUFE +++ Games
auf Kass.: Helicopter, Soul of
a Robot, Ghosthunters,
Bombscare, Chiller, Macadam,
Bumper u. 10 weitere Games.
Tel. 089/616389. Bitte nur
Mo. u. Die. zw. 17 u. 18.30
Uhr anrufen. Nur Barbezahl-
ung.

Tausche Programme auf 3"-
Dis., nur in Gießen u. Wies-
eck. Jeder Brief wird beant-
wortet (bitte m. 80-Pf.-Rück-
umschlag). Liten bitte an:
Marc Pankow, Weimarer Str.
6, 6300 Gießen-Wieseck.
PS: Bitte im Alter v. 11-16 J.

Verkaufe od. tausche 1 CPC-
Floppy DDI-1 gg. Commodore-
Floppy. VHP 500,- DM.
Tel. 07822/7056. 3 Mon. alt.

7054 Korb



Computer + Software
Winnender Str. 25, 7054 Korb
Tel. 07151/325 13

7700 Singen




Ihr Fachhändler
Marianne Tröndle
Lindenstr. 3, 7700 Singen
(Hohentwiel), Tel. 077 31/644 33

7700 Singen

Ihr kompetenter, autorisierter
SCHNEIDER-Fachhändler



Hard- u. Software, Peripherie,
Literatur
Fachkundige Beratung - kompletter
Service
7700 Singen, Freibühlstr. 21-25
Tel. 07731/82020

Erste Hilfe gegen Spielfieber

Um Ihnen ein wenig mehr zu bieten, als nur Testberichte über Spiele, möchten wir hier ein paar Tricks verraten, damit Sie möglichst viel von Ihrem Software-Kauf haben. Denken Sie daran, daß wir dabei auf Ihre Mithilfe angewiesen sind und daß sich das Mitmachen für Sie lohnen kann. Viel Spaß also bei unserer „Ersten Hilfe“.

Hallo, Spielefreaks und Joystick-Artisten! Willkommen in den Payer's Pages, unseren neuen Spieleseiten! Bis jetzt sind zwar noch nicht allzu viele Tips, Pokes, Leistungen und Fragen bei uns eingegangen, aber dies liegt natürlich an unserem redaktionellen Vorlauf. Also nicht gleich böse sein, wenn Ihr Tip noch nicht veröffentlicht wurde. Aber wir möchten hier noch einmal kurz zur Mithilfe aufrufen: Also, wir suchen Hinweise aller Art zu Spielen und Adventures für den CPC und natürlich auch für den PC! Auch Fragen zu bestimmten Spielen sind uns herzlich willkommen. Und ganz nebenbei nimmt ja jeder Einsender eines Pokes, Tips, Tricks, einer Lösung oder eines Problems an unserer Verlosung teil. Wir verschleudern nämlich jeden Monat ein Originalspiel für Ihren Computer (Typ angeben). Also, mitmachen und gewinnen!

1942

Und nun zu den neuen Spieltips: Der erste betrifft das Arcade-Adventure „1942“ von Elite. Das folgende Programm lädt das Hauptprogramm „1942.BIN“ von der Kassette. Anschließend können Sie die gewünschte Anzahl von Flugzeugen eingeben, welche in den Speicher gepoked wird. Danach wird das Programm ganz normal gestartet.
10 REM 1942-LIFEHELPER
20 REM für die Kassettenversion!!!
30 openout“d“:memory &257:load
“1942.bin“

```
40 input“Wieviel Flugzeuge (0-255)
:“,a
50 if a<0 or a>255 then 40
60 Poke &251C, a:call &89B3:
call &51C7
```

ASTRO ATTACK

Unser nächster Poke betrifft den beliebten Oldie „Astro Attack“ von Amsoft. Hier stehen dem Spieler bekanntlich nur drei Schiffe zur Verfügung. Mit folgendem Poke können Sie sich aber 255 Schiffe



Nemesis: Linksflug täuscht den Gegner

beschaffen: POKE &7AS0,&FF. Dieser Poke muß als Zeile 545 in den Lader eingebaut werden, der anschließend ganz normal mit RUN gestartet wird.

GREEN BERET

Daß sich brutale Kampfspiele großer Beliebtheit erfreuen, zeigt die Nachfrage im Laden. Aber viele, ansonsten recht erfolgreiche, Spieler kommen einfach nicht recht voran. Das größte Problem ist für viele, den ersten Level zu verlassen,

was folgendermaßen leicht zu bewerkstelligen ist: Bevor der Bildschirm das letzte Mal scrollt, ertönt die Sirene. Jetzt sollten Sie warten, bis ein Gegner mit einem Flammenwerfer vorbeikommt, den Sie ihm freundlich aber energisch abnehmen. Nun betreten Sie das letzte Bild und halten sich am linken Rand auf. Da die Gegner von nun an von rechts gelaufen kommen, müssen Sie nur warten, bis genug im Bild sind. Dann können Sie den Flammenwerfer benutzen. Nach den drei Schuß kommen noch etwa drei bis vier Feinde, die Sie sich mit dem Messer vom Leib halten.

Die Hunde am Ende des zweiten Levels lassen sich nur liegend mit dem Messer vernichten. Hierbei müssen Sie aber schnell sein! Die Hubschrauber am Ende des dritten Levels schießen Sie mit dem Granatwerfer ab. Für alle, die es immer noch nicht schaffen, hier noch ein kleiner Poke: POKE &34C,&B7.

ANIMATED STRIP POKER

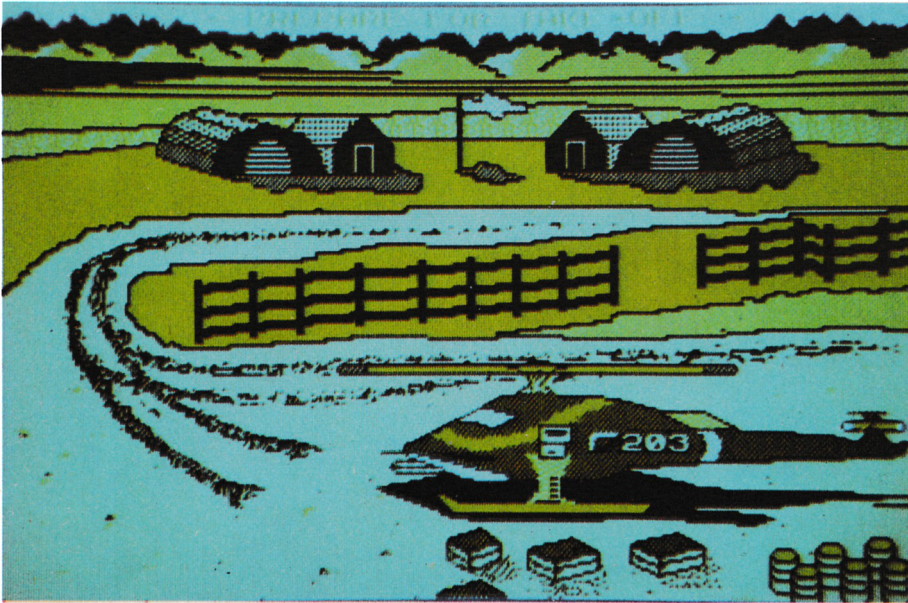
In diversen Zeitschriften wurden schon die CALLs zum Aufrufen der einzelnen Bilder abgedruckt. Wer jedoch ein bißchen sichergehen will, daß er beim Pokern mit Mindy auch gewinnt, dem sei folgende Methode ans Herz gelegt:

1. <Poker Basic> laden.
2. Zeile 2500 folgendermaßen ändern: 2500 CL=0:PCL=0:PM=500:CM=6:POT=0
3. Programm starten oder abspeichern.

Nach Programmstart hat Mindy nur sechs Pfund, Sie aber 500! Wenn Sie nun einmal gewinnen, sind Sie bereits eine Stufe weiter. Das geht so bis zum Schluß. Was Sie aber tun müssen, wenn die wackere Lady trotz des geringen Kapitals gewinnt, wissen wir auch nicht.

NEMESIS

Wer bei „Nemesis“ sein Flugzeug nur am linken Bildschirmrand bewegt, dem können die Gegner nichts anhaben, da sie kurz vorher abdrehen. Vorsicht ist allerdings ge-



Infiltrator: Welcher Weg führt zum zweiten Level?

boten, wenn ein Rudel Feinde vernichtet ist und das Objekt erscheint, durch dessen Abschluß der Spieler in die nächste Stufe kommt, denn mit diesem Objekt kann er kollidieren. Also Nerven behalten!

EDEN/DOOMSDAY BLUES

Von Stefan Hochmuth aus München stammt das folgende kleine Programm, welches einen Binärlader für das Hauptprogramm „EDEN“ von Doomsday Blues erzeugt. Der Lader kann dann künftig mit RUN geladen werden. Allerdings funktioniert er nur bei der Kassettenversion!

```
5 ' *** EDPO - (C) Stefan Hochmuth 1987
10 a$ = "0604212ABE11008CD77BCD213BD214000CD83BCD213BDCD7ABCD213BD21867136FF3E01CDOEBCC3007000"
20 MODE 1:call &BC02
30 adr=&BE00:for i=1 to 84 step 2:a=VAL("&"+MID$(a$,i,2)):poke adr,a:adr=adr+1
40 sum=sum+a:next:if sum<>4688 then?"Bitte Zeile 10 überprüfen!":end
50 input"Name des Hautprogramms ":"n$
60 for i=0 to len(n$)-1:poke &BE2A+i,ASC(mid$(n$,i+1,1)):next:poke&BE01,LEN(n$)
70 print"Taste zum Saven drücken!":call&bb18:save"EDEN.LDR",b,&BE00,&39,&BE00
```



Dogfight 2187: Unsterblichkeitspoke gesucht

ROLAND IN THE CAVES

Dieter Bachmayr aus München-Trudering hat uns einen netten kleinen Trick für „Roland In The Caves“ anvertraut: Dieses Spiel aus der Roland-Reihe von Amsoft erfreut sich immer noch großer Beliebtheit. Da es aber relativ schwierig ist, alle Levels durchzustehen, hier ein kleiner Schummeltrick: Durch gleichzeitiges Drücken der Tasten SHIFT und A gelangen Sie in den nächsten Level. So einfach kann es gehen, wenn sich der Programmierer ein Hintertürchen offen läßt.

VIKINGS

Bei Vikings von Kena Line Ltd steht der arme Held, der ein ganzes Dorf vernichten soll, am Anfang ja recht waffenlos da. Deshalb sollte er sich zuerst einmal auf die Suche nach einem Schwert machen: Geht er von Spielbeginn an geradeaus, so kommt er in das Dorf. Hier findet er etwas weiter rechts ein Schwert. Nimmt er dieses mit dem Feuerknopf auf, hat er ab sofort eine unbegrenzte Zahl von Schwertern zur Verfügung.

WER WEISS MEHR?

Auch diesmal sind wieder viele Fragen zu beantworten. Also, wer kann helfen? Gibt es einen Poke für „Highway Encounter“ oder „Dogfight 2187“? Kennt jemand die genaue Lösung zu „Saboteur“? Gibt es bei „Bobby Bearing“ einen Cheatmode? Hat jemand schon einen Lageplan zu „Fantastic Voyage“ entwickelt? Wer kennt einen Poke für „Arcanoid“? Wer weiß einen Poke für „Clever & Smart“? Kann man „Dan Darach“ eigentlich auch mit dem Joystick spielen? Gesucht werden auch Pokes zu Lightforce, Batty, Shockway Rider, Dandy und Rogue Trooper!

So, das war's wieder für dieses Mal. Also, Leute, nichts wie ran an die Joysticks und Pokes: Tips und Lösungen gesucht! Denn nicht vergessen: Das Mitmachen kann sich lohnen, denn nur wer mitmacht, kann auch gewinnen! TB

Jobkiller Computer? Ja, aber!

Menschenleere Fabrikhallen. Computer steuern Roboter an den automatisierten Fertigungsstraßen. Um die immer größer werdende Anzahl von Arbeitslosen bezahlen zu können, erhebt die Regierung eine Abgabe auf jeden Roboter. Die Maschinen zahlen „Lohnsteuer“. Ein Science-fiction-Roman? Sicher nicht. Japan lebt mit dieser Geschichte und ein annehmbares Ende ist noch gar nicht abzusehen. Für jeden weiteren Roboter müssen die Firmen eine monatliche Abgabe entrichten, um die Staatskasse aufzufüllen. Die – auch in Japan – zunehmende Anzahl der Arbeitslosen muß schließlich bezahlt werden. Wer noch Arbeit hat, dem sitzt die Angst im Nacken. In Europa, oder speziell in Deutschland, ist man noch nicht soweit. Hier war ein wenig Glück im Spiel, daß wir mit unse-

LERNEN AUS JAPANS FEHLERN

ren Fabrikationsmethoden nicht immer auf dem neuesten Stand waren. Japan ging allzu enthusiastisch an Rationalisierungsaufgaben, machte dabei Fehler und wir können daraus lernen. Zudem schützt uns ein soziales Netz vor Entlassungen, wenn Roboter das Fließband erobert. Firmenleitungen müssen einen Sozialplan erstellen und die Beschäftigten in anderen Abteilungen unterbringen. Der schlimmste Fall – dies passiert leider häufig – ist, daß keine Neueinstellungen erfolgen, obwohl die Produktion gestiegen ist. Der Nachwuchs der jeweiligen Berufe hat es immer schwerer, einen Arbeitsplatz zu finden. Sorgen müssen wir uns

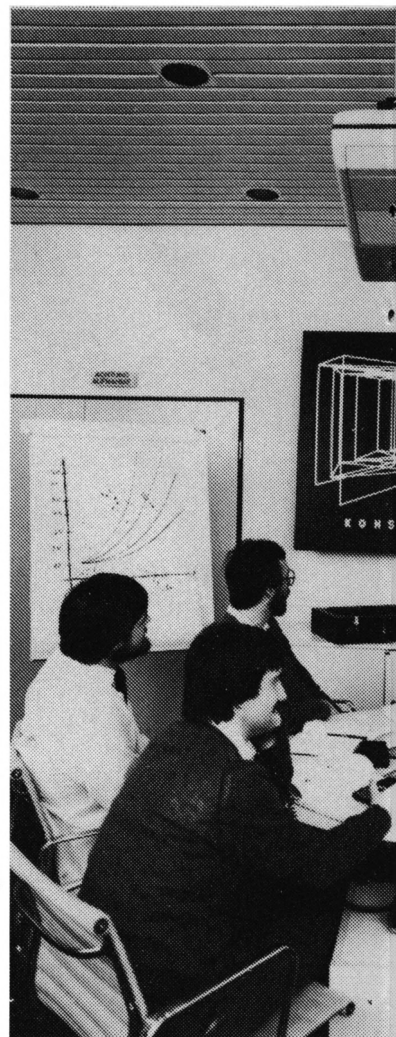
aber vorerst auch um die Konkurrenzfähigkeit deutscher Firmen machen, also den Erhalt des jeweiligen Betriebes. Gerade ein Betrieb ohne jegliche Datenverarbeitung ist in seiner Existenz gefährdet und viele Stellen stehen auf dem Spiel.

So ist die EDV mittlerweile Gesprächsstoff bei jeder Betriebsversammlung, in den Familien oder auch beim Stammtisch. Manchmal fehlt es jedoch am nötigen Hintergrundwissen. Was kann der Computer, wen kann er überhaupt ersetzen? Dieser Text soll dazu mit einigen Fallbeispielen und Theorien zur weiteren Diskussion verleiten. Dabei wurde in voller Absicht das weite Feld der Roboter-Steuerung vergessen. Wie es in den Fabriken aussieht, weiß jeder. Niemand kann damit rechnen, daß ein Fließband-Job bis zur Rente reichen wird. Die Technik ist noch längst nicht am Ende, aber gesellschaftspolitisch ist der Umbruch fast vollzogen.

DER PC RÜCKT AN DIE STELLE DER ROBOTER

Die allgegenwärtige Angst vor der Elektronik verlagerte sich dann auch auf den PC. Während ein Facharbeiter seiner Arbeit noch relativ sicher sein kann, beobachten Angestellte, die „Büroarbeiter“, argwöhnisch die EDV-Einführung im Büro. Hier tut sich tatsächlich einiges und die Frage taucht auf, wessen Job entbehrlich wird. Bei allem Für und Wider ist diese Frage nur individuell zu beantworten, weshalb wir hier einige Fallbeispiele bringen, aus denen Einzelheiten herausgezogen werden. Betrachten wir aber zu

allererst einmal ein kleines Büro, in dem zwei oder drei Angestellte eine Buchhaltung, eine Versandorganisation oder ähnliches zu erledigen haben. Jeder kann sich ausmalen, daß hier der PC keinen Lohnempfänger ersetzen wird, vorausgesetzt, diese finden sich mit der neuen Technik ab und sind bereit, damit zu arbeiten. Ein Abteilungsleiter wird keineswegs seine Sekretärin entlassen, weil ihm nun ein Computer zur Verfügung steht. Mit einem Textprogramm lassen sich die Briefe zwar leichter und auch schneller schreiben, immer noch ist es aber die „Vorzimmerdame“ die den manuellen Vorgang beherrscht. Ihr bringt der Computer dann auch die eigentliche Arbeitsentlastung. Andererseits ist es natürlich denkbar, daß hier ein Rechner einen von mehreren gleichartigen Angestellten ersetzt. Das Beispiel einer größeren Arztpraxis – mit vier Sprechstundenhilfen – soll dies verdeutlichen. Die Quartalsabrechnungen für die Krankenkassen sorgte in derartig großen „Betrieben“ immer für Überstunden und beanspruchte sämtliche Kräfte. Die Belege mußten pünktlich abgeliefert werden, sollen aber auch Eintragungen bis zur letzten Minute beinhalten. Für ein bis zwei Wochen ging es hektisch zu. Für diese „Hochkonjunktur“ mußte aber der Arzt das Personal berechnen, auch wenn danach eine ruhigere Zeit von ca. 9 Wochen kam, in der er mit 3 Hilfen ausgekommen wäre. Wird hier nun ein PC eingesetzt, der mit einem entsprechendem Programm die Arbeit alleine deshalb erleichtert, weil über das gesamte Quartal eingege-

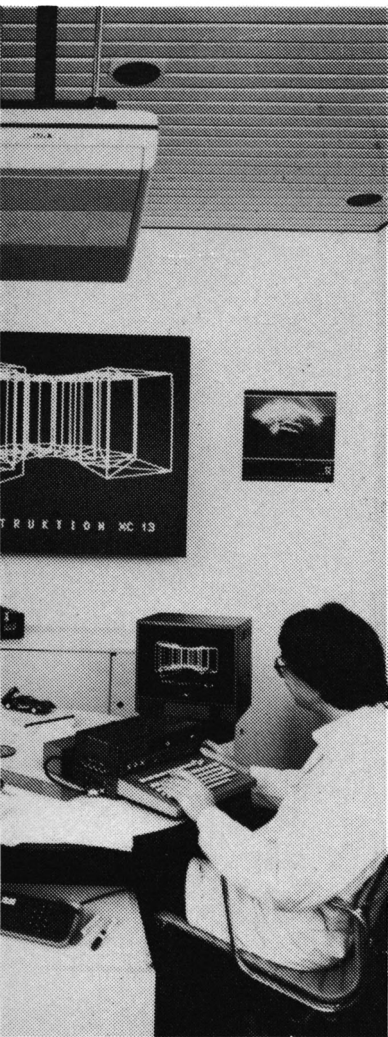


Nicht nur als Jobkiller, sondern auch als hilfreiches Werkzeug läßt sich ein Computer einsetzen.

ben werden kann und die Abrechnung erst am letzten Tag „per Knopfdruck“ erfolgt, dann wäre tatsächlich weniger Personal nötig.

AUCH POSITIVE SEITEN BEI DER EDV-EINFÜHRUNG

Computer können einen Herstellungsbetrieb auch konkurrenzfähiger machen, ohne daß es deswegen zu Entlassungen kommen muß. Ein immer wieder auftretendes Beispiel sei hier zitiert.



Weil früher ein Angebot kalkuliert wurde, indem man durch die einzelnen Abteilungen ging und den Meister befragte, wie lange er für seinen Teil der Arbeit brauchte, kam es immer wieder zu Fehlberechnungen. Der Kostenvoranschlag war zu hoch, man verlor den Auftrag an eine preiswertere – meist ausländische – Firma. Heute erfolgt die Kalkulation im Computer. Er hat Richtwerte, wie lange man beim Fräsen braucht und welche Zeiten bei der Lackierung zu veranschlagen sind. Zusammen mit einigen anderen exakten Werten kann alles auf das neue Werkstück umgerechnet werden. Anschließend ist ein genauer Endpreis ermittelt, mit dem man konkurrenzfähig ist. Er liegt fast immer unter den Kalkulationen,

die ohne EDV durchgeführt werden. Woran liegt es? Der Mensch schreckt nun einmal vor Neuerungen im Arbeitsablauf zurück. Der Meister, der alte Hase, der immer weiß was läuft, war naturgemäß recht vorsichtig, wenn er eine Arbeitszeitschätzung machte. Er baute eine Schutzmauer um seine Abteilung und veranschlagte lieber einen halben Tag mehr, um nicht in Druck zu geraten. Daß er damit der Wettbewerbsfähigkeit seines Arbeitgebers schadete, erkannte der Fachmann nicht. Betriebe, die auf eine Computerkalkulation umrüsten, decken damit auch diese „Wespennester“ auf und es gibt kurzfristig Ärger. Der Computer ist da realistischer. Seine Werte für die einzelnen Arbeitsvorgänge basiert auf Werkstücken, die bereits routiniert und schnell gefertigt werden. So soll es ja auch einmal bei der eigentlichen Fertigung mit dem Auftrag geschehen und der gebremste Elan bei neuen Arbeitsvorgängen ist dem Elektronenhirn nicht einprogrammiert worden.

BEISPIEL FÜR DIE GELUNGENE EINFÜHRUNG DER RECHNER

Ein letztes Beispiel soll zeigen, wie geschickt sich Computer einsetzen lassen, ohne Arbeitslose zu erzeugen. Daß es sich dabei nicht um einen PC im üblichen Sinne (sondern um eine Großrechenanlage mit Erfassungsterminals) handelt, ist eigentlich unwesentlich. Ein großes Automobilwerk lieferte an alle Händler und Werkstätten der Firma PCs aus. Diese können zwar auch als Einzelrechner arbeiten, ihre Aufgabe ist aber der Verbund mit dem Großrechner der Firma. Beim Verkauf der Pkws ergeben sich großartige Möglichkeiten. Der Kunde weiß sofort, ob ein Sondermodell ver-

fügbare ist, oder wann sein Auto produziert und geliefert wird. Wenn der Käufer ein rosafarbenes Cabrio mit lila Punkten bestellt und das Modell in Hamburg auf Halde steht, dann hat es der Händler innerhalb einer Woche. Viel wichtiger wurde der Computer jedoch für die Werkstätten. In den Vertragsbetrieben wurden aufgrund der vorhandenen Zahlen die „Durchläufe“, also die Reparaturen je Tag, ermittelt. Daraus konnte eine Statistik abgeleitet werden, nach der das Ersatzteillager aufgefüllt wurde. Ein Auspuff, zwei Kotflügel, vier Batterien und so weiter. Nach einer gewissen Zeit werden diese Erstwerte aktualisiert und werden realistisch. Zum Schluß kann sogar auf saisonale Einflüsse (mehr Blechschaden im Winter) Rücksicht genommen werden. Mit diesem Material im Lager kann der Meister arbeiten. Schreibt er dann abends die Rechnung in den Computer, so wird in der Zentrale automatisch das Ersatzteil als Bestellung ausgeführt. Am nächsten Tag ist das Lager der Werkstatt wieder aufgefüllt, ohne daß eine aufwendige Verwaltung notwendig wird. Und weil nun einmal alles kontrolliert wird, kann auch gleich eine Bilanz und Buchhaltung ausgeführt werden. Da auch bei größeren Werkstätten die Lagerhaltung „nebenher“ erledigt wurde, fällt kein Arbeitsplatz weg. Nun hat der Meister Zeit, sich um das Auto zu kümmern. Welche Formulare er bei der Bestellung eines Auspuffes benötigt, kann ihm egal sein. Und daß sämtliches Material für eine Inspektion im Lager ist, davon kann er ausgehen. Die Nachteile dieses Systems sollen nicht verschwiegen werden. Die Werkstatt hat keinen großen Handlungsspielraum. Sie ist auf das Ersatzteillager des Werkes angewie-

sen und kann nicht mehr durch preiswerten Einkauf kalkulieren. Und weil „Big Brother“ alles weiß, und sogar über den Einsatz der Lehrlinge bestimmt, besteht nicht einmal die Möglichkeit, diesen einzusetzen und dafür Meisterlohn zu kassieren.

GEWÖHNEN WIR UNS AN DIE EDV, OHNE SIE GEHT ES NICHT

All die Beispiele die hier vorgestellt wurden, repräsentieren natürlich nur den gegenwärtigen Stand der Dinge. Es ist überhaupt noch nicht absehbar, welchen Einfluß der Computer letztlich auf Dienstleistungs- oder Herstellungsbetriebe nimmt. Die Entwicklung der EDV ist noch nicht abgeschlossen, bestenfalls stehen wir am Anfang.

Ob der Computer Arbeitsplätze wegrationalisiert oder gar neue schafft, ist nicht zu klären. Die Beispiele, die genannt wurden, sollten nur beweisen, daß diese Frage individuell zu sehen ist. Wer sich darüber Gedanken macht, der sollte aber auch wissen, daß die Ablehnung neuer Technik mit Sicherheit dazu führt, daß man künftig um den Job bangen muß. Sei es, weil man an dem persönlichen Arbeitsplatz mit der Technik konfrontiert wird und diese nicht bewältigt, oder weil der Arbeitgeber ohne Computer wettbewerbsunfähig auf den Konkurs zusteuert.

Die elektronische Lawine kam einmal ins Rollen und heute kann sie niemand mehr aufhalten. Es gilt, mit der Konkurrenz gleichzuziehen oder „den Laden zu schließen“. Es gilt, sich an den PC auf dem Schreibtisch zu gewöhnen oder sich auf dem Arbeitsplatz um die wenigen EDV-losen Arbeitsplätze anzustellen. Niemand fragt, ob wir Elektronik befürworten – wir haben uns daran zu gewöhnen.

(GS)

GEM contra GEM

Atari gegen PC 1512 — gibt es einen Fortschritt?

Der erste GEM wurde mit dem Atari ST ausgeliefert. Allerdings ist dies schon eine Weile her und bei neueren Geräten, etwa dem PC 1512, erwartet man eigentlich auch eine verbesserte Version des Desktop-Programmes. Wo liegen denn nun die Unterschiede? Oder gibt es gar Verbesserungen?

Die Idee zu diesem Artikel kam beim Durchlesen eines Vergleichstests zwischen der Amiga Workbench und dem GEM des Atari ST. Wie vielleicht nicht jeder weiß, war der Atari ST der erste Rechner, der mit dem Desktop GEM (für Graphic Environment Manager — Manager der grafischen Umgebung) arbeitete. Verantwortlich für diese Schreibsimulation war das Softwarehaus Digital Research. Die Programmierer orientierten sich weitgehend am bereits bestehenden Maus-System des McIntosh von der Firma Apple. Die Übereinstimmung geriet jedoch etwas zu perfekt, und es mußten nach einer Klage, die einen Prozeß nach sich gezogen hätte, ein paar Abstriche gemacht werden, damit man sich außerhalb des Gerichtssaales einigen konnte. Spätestens dann geriet der Atari in den Verdacht, in einer Magerversion ausgeliefert zu werden. Mittlerweile sind jedoch zwei Jahre ins Land gegangen und mit dem Auftauchen neuer Computer gab es auch neue

Software. Es wäre nun an der Zeit, einmal zu testen, ob der Schneider PC als jüngstes Mitglied der GEM-Familie Vorteile bietet, oder ob er alte Mängel mit sich herumschleppt. Es geht dabei auch um die Frage, ob ein MS-DOS Rechner mit 8086 Prozessor dem Motorola 68000 — der intern immerhin mit 32 Bit arbeitet — gewachsen ist. Da der Atari ST sonst über keinerlei Zusatzchips verfügt, wie etwa der Amiga, dürfte der Wettbewerb einigermaßen fair werden.

DAS PROBLEM DES EINLESENS

Beide Konkurrenten können beim Einlesevorgang des Desktops aus unterschiedlichen Gründen nicht gut abschneiden. Die Schwierigkeiten beim Schneider PC sind allgemein bekannt: GEM ist auf zwei Disketten verteilt und der leidgeprüfte Anwender wird zum Diskjockey. Da niemand weiß, wie lange der Wechsel dauert, kann nur eine ungefähre Zahl ermittelt werden.

43 Sekunden, so der Wert ohne den Diskettenwechsel.

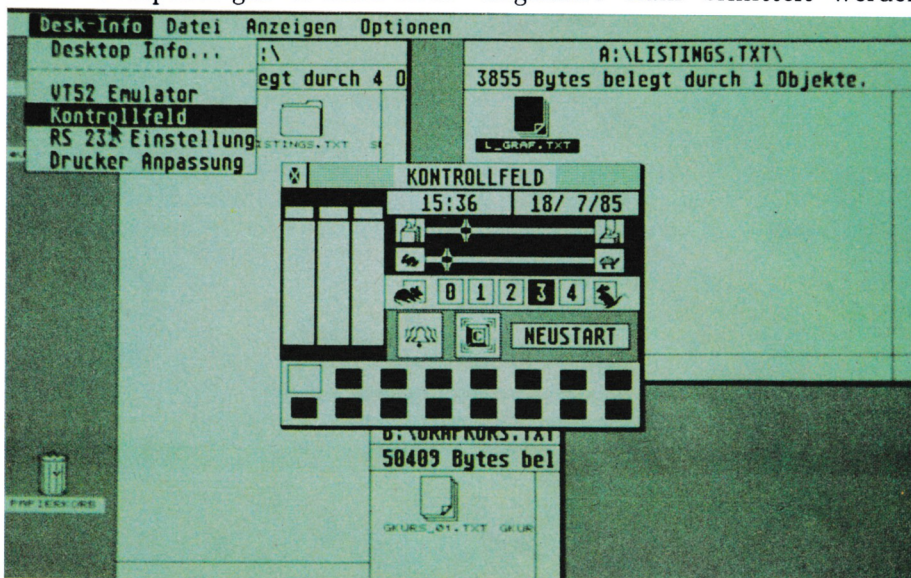
Ähnlich schlecht sieht es beim Atari ST aus. 59 Sekunden benötigt diese GEM-Version, um sich auf dem Bildschirm zu präsentieren. Zwar bleibt der Diskettenwechsel erspart, doch trotzdem wäre es sinnvoller, beim Einschalten des Rechners eine Diskette mit einer individuellen Arbeitsversion im Laufwerk zu haben. Nur dann kommt der Atari ST mit 39 Sekunden aus. Die Bedienungsirrtümer, die dadurch zustande kommen, lassen diesen Zeitvorteil nichtig werden. Fassen wir einmal zusammen: Während der Schneider PC von zwei Disketten aus liest, bietet der Atari ST eine ROM-Version, die merkwürdigerweise 20 Sekunden benötigt, um festzustellen, ob eine Diskette eingelegt wurde. Wenn dies der Fall ist, dann wird in 29 Sekunden eingelesen.

Nur der Vollständigkeit halber seien hier die Werte für die Schneider Festplatte genannt: Mit nur 15 Sekunden ist der PC 1512 unschlagbar schnell. Aber — wie bereits erwähnt — dies ist der Wert bei einer Festplatte und natürlich ohne Diskettenwechsel.

GRUNDAUSSTATTUNG

Digital Research tat einiges innerhalb der letzten zwei Jahre. Der „Schnellschuß Atari“ mußte korrigiert und verbessert werden. Viele Möglichkeiten gab es dazu nicht. Im wesentlichen stand das System fest, lediglich mit den Utilities konnte etwas Luxus erzeugt werden.

Beginnen wir bei den Voreinstellungen: Auch hier liefert der Atari ST aus dem ROM heraus nur eine äußerst knappe Version. Lediglich die Löschestätigungen und der „Piepton“ können eingestellt werden, um die Arbeit mit dem Betriebssystem zu erleichtern. Erst bei einem „Disketten-GEM“ erhöht sich der Komfort. Neben den Farbmischungen gibt es einen simulierten Schieberegler für die Übersetzung der Mausbewegung und einen Wahlschalter, mit dem man das Tempo des Doppelklicks einstellen kann. Eine Uhr (mit Datum) ist zwar vorhanden, muß aber bei jedem Neustart eingestellt werden.



Das Kontrollfeld des ST erlaubt nur die einfachsten Veränderungen für das Desktop

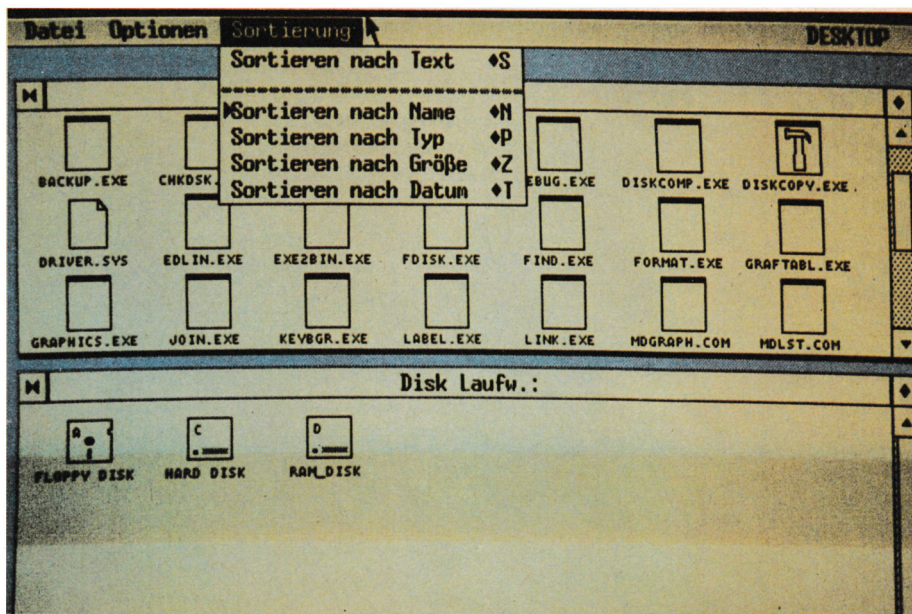
Im Wesentlichen war dies alles, was das Kontrollfeld – so heißen die Voreinstellungen beim Atari ST – zu bieten hat. Es soll allerdings nicht verschwiegen werden, daß unter den Pull-Down Menüs des Desktops noch andere Möglichkeiten zu finden sind. Positiv ist die Einstellung der seriellen Schnittstelle, die einfacher nicht mehr zu handhaben ist. Das gleiche Ziel wollte man anscheinend auch beim Druckertreiber verfolgen und tat damit einiges zuviel. Es genügt nicht immer, die Grafikauflösung des Peripheriegerätes und andere Kleinigkeiten einzustellen. Wie kompliziert eine perfekte Druckeransteuerung ist, weiß wohl jeder Leser selbst.

preiswerter, aber auch praktischer Clou gelungen: das NVR-Programm mit seiner zusätzlichen Hardware-Einrichtung, dem batteriegepufferten RAM. Zuerst mutet diese Lösung natürlich sehr billig an, für den alltäglichen Gebrauch läßt sich jedoch nichts Besseres denken (mit Ausnahme eines Akkus, der natürlich auch teurer wäre). Sämtliche Wünsche, die man ansonsten an die Funktionsweise des GEM hat, können eingestellt und abgespeichert werden. Darunter auch einige Parameter, die der Atari ST nicht berücksichtigt. Die ENTER-DEL, Joystick- und Maus-Übersetzungswerte helfen bei der Überwindung einiger Softwareprobleme. Maus-Bel-

Neben den Standardüberlegungen gibt es noch optionelle Parameter und eine erweiterte Ablaufkontrolle. Dies mag zuerst einmal die DFÜ-Freaks freuen, doch dahinter steckt noch mehr. Spätestens hier können Druckerprobleme beseitigt werden, vorausgesetzt, das Gerät verfügt über einen seriellen Eingang.

Der allergrößte Vorteil dieser Voreinstellungen wurde schon im Zusammenhang mit der RAM-Disk erwähnt: Das Individual-Desktop kann in einem batteriegepufferten Speicher abgelegt werden, wo eine Batterie die Parameter über ein halbes Jahr sichert.

Beim Atari-GEM muß das Ergebnis des Kontrollfeldes auf eine Diskette abgelegt werden. Dabei gehen unter anderem auch Uhrzeit und Datum verloren, abgesehen davon, daß diese Diskette beim Einschalten des Gerätes im Laufwerk sein muß. Da nützt es wenig, daß es auch ein Desktop im ROM gibt. Dort ist nur eine Sparversion untergebracht, die nicht (Read Only Memory) verändert werden kann.



Mit nur zwei Windows übersichtlich und ordentlich – Das Schneider GEM

Bei Schneider liegt die Sache etwas anders. Man verläßt sich völlig darauf, daß der Benutzer einen IBM-kompatiblen Drucker hat. Darauf ist der GEM eingerichtet und es gibt keinerlei Probleme. Sollte der Anwender allerdings auf eine exotischere Marke zurückgreifen, dann beginnen seine Sorgen. In einigen Fällen vermag die Software, also das Textverarbeitungsprogramm oder ähnliche Anwendungen, weiterzuhelfen, im GEM steht der User allerdings recht hilflos da. Die Druckereinstellungen entfallen also, sowohl bei den Voreinstellungen als auch im eigentlichen Desktop.

Doch zurück zur individuellen Einrichtung eines Schreibtisches. Schneider ist hier ein besonders

wertungsmaßstab und Farbeinstellungen finden sich auch im Atari-GEM, aber schon bei der RAM-Disk ist der Schneider wieder um eine Nasenlänge voraus. Zwar ist der virtuelle Speicher auch im ST zu realisieren, aber nicht ohne zusätzliche Software, zumeist aus dem Public-Domain Bereich. Außerdem verschwindet das „Pseudo-Laufwerk“ nach dem Ausschalten. Um Mißverständnissen vorzubeugen, sei erwähnt, daß der Inhalt einer RAM-Disk nicht gespeichert wird, sondern nur deren Größe. Aber so ist es nun einmal bei einem virtuellen Laufwerk. Auch bei den Einstellungen zur seriellen Schnittstelle ist einiges mehr möglich als beim GEM-Erstwerk des Atari.

ANZAHL DER FENSTER IST STARK EINGESCHRÄNKT

Der Atari-GEM bietet vier Fenster. Es waren in den ersten Versionen noch mehr, doch diese Einschränkung war ebenfalls eine Folge der außergerichtlichen Einigung mit Apple. Allerdings ist es bei der Beweglichkeit der Windows geblieben. Diese haben kein vorgeschriebenes Format, können beliebig vergrößert oder verkleinert und auf dem Bildschirm bewegt werden. Dabei stellt sich dann auch schnell heraus, daß vier dieser Teilbildschirme für die alltägliche Arbeit genügen. Mehr wäre verwirrend.

Beim Schneider muß sich der Anwender mit zwei Fenstern begnügen, und das ist manchmal zu knapp bemessen. Auch mit der Beweglichkeit ist es nicht weit her, die beiden teilen sich den Bildschirm genau zur Hälfte und verbleiben an ihrem Platz. Im MS-DOS Rechner kann das Window nicht nach Belieben vergrößert werden. Klickt man das dazugehörige Symbol an, dann wird der gesamte Bildschirm genutzt und dabei natürlich ein Fenster überdeckt.

In der Icon-Darstellung gleichen sich die beiden Programme. Dateien erscheinen als Viereck mit „Eselsohr“, Programme haben einen schwarzen Balken am oberen Rand. Zusätzlich können beim Schneider noch kleine Zeichnungen im Programm-Icon gezeichnet sein, wie dies bei dem Editor RPED und dem NVR-Einstellungsprogramm der Fall ist. Allerdings kann ein Anwender ohne hervorragende Programmierkenntnisse keinen Einfluß auf die Bildchen nehmen. Dies nur für den Fall, daß jemand den Icon-Editor des Commodore Amiga vor Augen hat.

Bei den Textdateien bietet der Atari einen wesentlichen Vorteil gegenüber seinem Nachkommen. Klickt man das „Blatt Papier“ an,

ne einiges mehr möglich. Darunter auch der TYPE- oder PRINT-Befehl, mit dem Dateien direkt und ohne Umweg über die passende Anwendung dargestellt werden können.

DIE UTILITY-AUSSTATTUNG

Tja, wenn alles so gekommen wäre, wie es Jack Tramiel prophezeite, dann hätte der Atari PC das ultimative Desktop, es gäbe nichts zu verbessern. Noch kurz bevor die ersten Geräte im Handel waren, war sogar von GEM Write und GEM Draw die Rede, aber es blieb bei den Ankündigungen. So, wie der Speicherriese heute auf dem Schreibtisch steht, ist das Desktop äußerst mager. GEM Write wurde ersatzlos gestri-

sonst können sich die Kleinigkeiten sehen lassen. Der Taschenrechner hilft sicherlich in dem einen oder anderem Programm, welches an sich zwar nichts mit Mathematik zu tun hat, aber in dem man doch ein paar Berechnungen durchführen muß. Der Druckerspouler dient zur Aufnahme einiger Dateien, die dann bei der Weiterarbeit im Hintergrund ausgedruckt werden. Zu guter Letzt kann im Pulldown-Menü der Snapshot angeklickt werden, um einen Bildausschnitt zur Weiterverarbeitung abzuspeichern.

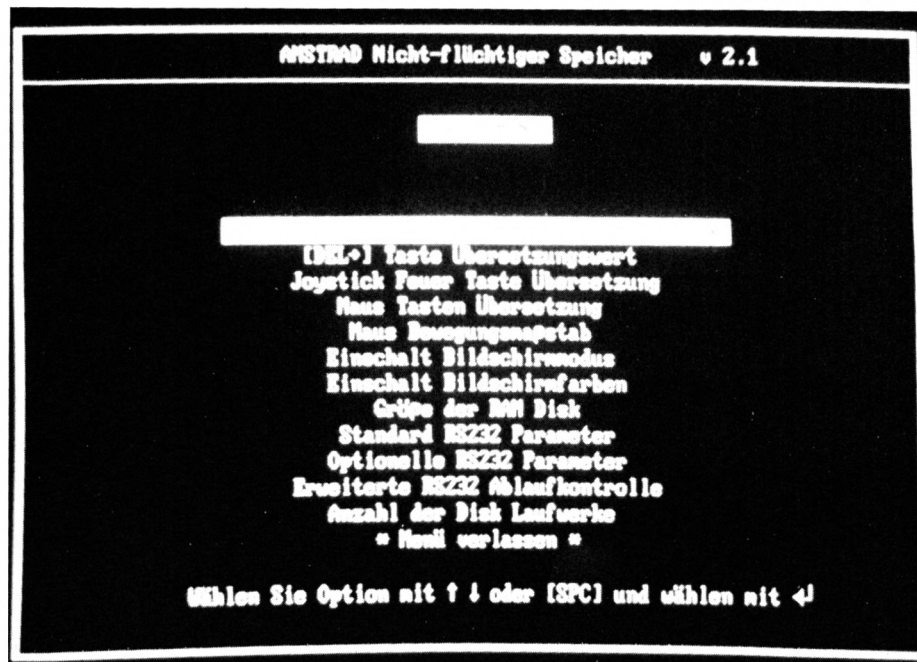
Auf der Diskette befinden sich noch weitere Hilfsprogramme wie der Editor. Natürlich ist auch „Doodle“ vorhanden, und es ist beim PC 1512 gleichermaßen überflüssig wie beim Atari. Als hätte man dies bei Schneider eingesehen, liefert man aber noch das vollwertige Zeichenprogramm „GEM Paint“ mit. Damit ist tatsächlich etwas anzufangen.

FAZIT: AUCH AN GEM IST DIE ZEIT NICHT SPURLOS VORBEIGEGANGEN

Vergleicht man beide Desktop-Programme, die den Namen GEM führen, dann werden schnell die Unterschiede klar: In fast allen Belangen hat der Schneider die Nase vorn, daran ändert auch der fixe Motorola 68000er Prozessor nichts. Im Gegenteil, gerade bei einigen zeitkritischen Optionen (Diskcopy und Format) ist der PC etwas schneller. Es hat ja auch nichts mit dem Prozessor zu tun, sondern liegt an Unzulänglichkeiten der Software.

Zwischen den beiden „Schreibtischsimulationen“ liegen mittlerweile auch zwei Jahre, und Digital Research wußte beim Schneider schon, wo der Atari Fehler hat. Daß man sich bei der älteren Version zu einem ROM überreden ließ, war ein Fehler. Hinzugekommen ist beim PC eine bessere Ausstattung mit Software und Utilities zum Wohle der Anwender. An die langsame Grafikverarbeitung mußte man allerdings Konzessionen machen, wie die Anzahl der Fenster und deren Unbeweglichkeit zeigt. Trotzdem ist der Schneider PC als Sieger aus diesem Vergleichskampf hervorgegangen.

GS



dann darf man sich den Inhalt auf den Drucker ausgeben lassen oder auf den Bildschirm. Steuerzeichen können zwar ein wenig Durcheinander erzeugen, aber es reicht immerhin, sich ein File noch einmal anzusehen, bevor man es in den Papierkorb versenkt. Beim PC 1512 gibt es diesen einfachen Weg nicht, ebensowenig den Papierkorb, der durch den nüchternen Befehl „Löschen“ ersetzt wurde.

Wettgemacht wird der Umstand durch die beiden „Türen“ zum Betriebssystem. Ob man sich nun für immer (besser: für die weitere Arbeit) von GEM verabschiedet oder einen Rückweg offen läßt, auf jeden Fall ist auf Betriebssystem-Ebe-

chen. Es fehlt sogar ein einfacher Editor wie RPED oder EDLIN. Mangels Schnittstelle zum Betriebssystem (TOS ergäbe solch eine Utility auch keinen Sinn. Für das Zeichenprogramm DRAW kam das Progrämmchen „Doodle“ in die Konfiguration. Was man damit überhaupt anfangen kann, ist immer noch nicht geklärt, bislang dient es verspielten Gemütern als Zeitvertreib. Soweit es den Atari betrifft, ist die Liste komplett. Die Uhr, die man sich auf den Bildschirm legen kann, disqualifiziert sich selbst, da man sie nach jedem Reset neu einstellen muß.

Beim Schneider läuft die Uhr zumindest ein halbes Jahr lang. Auch

VERDIENEN SIE GELD MIT IHREM COMPUTER! WIR ZAHLEN BIS ZU 1000 DM!

Haben Sie einen CPC 464 oder 664? Einen 6128?
Können Sie programmieren? In Basic oder Maschinensprache? Dann bietet SCHNEIDER CPC-WELT Ihnen die Möglichkeit, mit diesem Hobby Geld zu verdienen!

Wie? Ganz einfach. Sie senden uns die Programme, die Sie für einen Abdruck als geeignet halten, zusammen mit einer Kurzbeschreibung, aus der auch die verwendete Hardware — eventuelle Erweiterungen — benutzte Peripherie — hervorgehen muß, ein.

Benötigt werden: eine Datenkassette oder Diskette!
Wenn die Redaktion sich überzeugt hat, daß dieses Programm läuft und sich zum Abdruck eignet, zahlen wir Ihnen pro abgedrucktem Programm je nach Umfang bis

zu DM 300,-! Für das „Listing des Monats“ sogar DM 1.000,-. Sie erhalten Ihre Kassette/Diskette selbstverständlich zurück, wenn Sie einen ausreichend frankierten Rückumschlag mit Ihrer Adresse beifügen.

Bei der Einsendung müssen Sie mit Ihrer Unterschrift garantieren, daß Sie der alleine Inhaber der Urheber-Rechte sind! Benutzen Sie bitte anhängendes Formular! (Wir weisen darauf hin, daß auch die Reaktion englische Fachzeitschriften liest und „umgestaltete“ Programme ziemlich schnell erkennt).

Um Ihnen die Arbeit zu erleichtern, finden Sie hier ein Formular. Sie können es ausschneiden oder fotokopieren.

Name des Einsenders: _____
Straße/Hausnr./Tel.: _____
PLZ/Ort: _____

Hiermit biete ich Ihnen zum Abdruck folgende(s) Programm(e) an:

Benötigte Geräte: _____

Beigefügt Listings Kassette Diskette

Ich versichere, der alleinige Urheber des Programms zu sein!

Hiermit ermächtige ich die Redaktion, dieses Programm abzudrucken und wirtschaftlich zu verwerten. Sollte es in den Kassetten-Service aufgenommen werden, erhalte ich auch dafür eine entsprechende Vergütung. Das Copyright geht an den Verlag über.

Rechtsverbindliche Unterschrift

Unterschrift des Erziehungsberechtigten bei Minderjährigen

**SCHNEIDER CPC-WELT
PROGRAMM-REDAKTION
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM**

Utility:

Checksummer

Die regelmäßige Seite für den Checker. Damit tippen Sie die Listings aus der CPC-Welt ohne Fehler ab.

Speed-up

Die Bildschirmausgaben des CPC werden enorm beschleunigt. Insbesondere bei Grafik-Screens macht sich das kurze Listing bezahlt. Schon bald spart der CPC mehr Zeit bei der Programmausführung, als Sie für das Eintippen verwendet haben.

Anti-Era

Der Befehl ERASE ist schnell ausgeführt. Was aber, wenn man etwas vorschnell ans_Werk ging und eine_wichtige Datei löschte? „Anti-Era“ bringt Ihnen das Programm zurück. Voraussetzung ist, daß Sie zwischenzeitlich noch nichts Neues abgespeichert haben.

Anwendung:

Polygraf

Der Name verrät es schon: Endlich ein umfangreiches Grafikprogramm für Diagramme. Nicht nur für Geschäftsgrafik.

Musik-Demo

Abtippen und zuhören. Diesmal ein paar Spezialitäten für Klassik-Fans.

Privat-Manager

Auch das Privatleben will verwaltet sein. Eine kleine Hilfe bietet dieser Manager. Adreßbuch, Kalender, Haushaltsbuch, Taschenrechner und Notizzettel machen das Listing zum Superprogramm.

Spiel:

Solitär

Solitär ist Ihnen vielleicht unter dem Namen „Springerle“ bekannt. Diesmal ersetzt der CPC das Spielbrett und paßt auf, daß Sie nicht schummeln.

Hyperinversi

Sieben Levels sorgen dafür, daß Ihnen dieses Spiel so schnell nicht langweilig wird. Tauschen Sie Buchstaben und Zeichen, bis die richtige Reihenfolge hergestellt ist.

Softbox:

Alle Programme dieser Ausgabe können Sie als Softbox bestellen. Kein Abtippen mehr, sondern sofort Nutzen oder Spielspaß. Zusätzlich gibt es auf der Diskette 2/88 (nicht auf Kassette) das:

Bonus-Programm

Universaldatei unter CP/M

Das Programm nutzt den gesamten Speicherplatz einer Diskette. Die Eingabemaske können Sie selbst entwerfen, so daß sich das Verwaltungsprogramm für alle Zwecke einrichten läßt. Natürlich ist auf der Diskette eine ausführliche Bedienungsanleitung als LESMICH.BAS-Programm enthalten.

Bestell-Coupon

Den Bestell-Coupon finden Sie auf Seite 42

Checker

CHECKER ist ein Eingabeprüfprogramm, welches Ihnen helfen soll, Abtippfehler zu verhindern. Um damit arbeiten zu können, muß es erst (sehr sorgfältig!) abgetippt und abgespeichert werden.

Ein kleiner Hinweis, falls das Programm zwar ohne Fehlermeldung abläuft, aber trotzdem nicht korrekt arbeitet: Die Zeilen mit den DATA-Statements werden zwar mittels einer einfachen Prüfsummen-Routine auf grobe Fehler hin geprüft, Vertauschungen von DATA-Werten innerhalb einer Zeile werden aber nicht erkannt. Deshalb bitte wirklich Sorgfalt beim Abschreiben walten lassen.

Nach einem Programmablauf stehen zwei RSX-Befehle zur Verfügung: EIN und AUS. RSX-Befehle beginnen mit einem senkrechten Strich (Taste neben P mit Shift eingeben), dann folgt der Befehlsname. Mit EIN wird der Checksummer aktiviert, mit AUS wieder abgeschaltet.

Bei aktivem Checksummer erfolgt – nach der Eingabe einer Basiczeile – die Ausgabe einer vierstelligen Hexadezimalzahl, die in eckigen Klammern eingeschlossen ist.

Da unser CHECKER – im Gegensatz zu anderen Checksummen – auch die Zeilennummer mit überprüft, muß auch diese korrekt sein, damit die Prüfsumme stimmt.

Haben Sie ein Listing fehlerfrei abgetippt, dann speichern Sie dieses ab! Erst dann sollten Sie das Programm starten, denn es könnte ja sein, daß dort, wo unser CHECKER seinen Platz hat, selbst aktiv wird. Oder aber es besitzt eigene RSX-Befehle mit gleicher Aufrufbefehlsfolge und es treten daher Fehler auf! Am besten setzen Sie vor einem Programmablauf den CPC zurück!

Da die Firmware der drei CPC-Typen leider unterschiedlich ist, verhält sich unser „Eingabe-Prüfer“ beim Befehl AUTO unterschiedlich. Dieser Befehl kann beim CPC464 nicht benutzt werden, da dann keine Prüfsummen ausgegeben werden. Beim CPC664 und 6128 ist der Einsatz möglich und kann dazu dienen, ein abgetipptes Programm nachträglich zu überprüfen und zu korrigieren. Das Programm darf aber noch nicht gelaufen sein! Gegebenenfalls muß es also neu geladen werden.

Noch ein letzter Tip: CHECKER darf nicht zweimal hintereinander gestartet werden, da die zweimalige Einbindung von RSX-Befehlen (mit gleichem Namen) zu Fehlverhalten führt. Die RSX-Befehle selbst können Sie natürlich, so oft Sie wollen, aufrufen.

```
100 MEMORY &A2FF:'CHECKER.HEX
110 a=&A300:e=&A3F6:zb=1000:e=e+1
120 PRINT"Moment bitte...":FOR i=a TO e:
READ d$:IF LEFT$(d$,1)="/"THEN flag=1
130 IF(flag AND p(>VAL(d$)))THEN PRINT"Fe
hler in Zeile "zb+1:END
```

```
140 IF(flag AND i=e)THEN 180
150 IF flag THEN i=i-1:zb=zb+1:p=0:d$=""
:flag=0:GOTO 170
160 POKE i,VAL("&"d$):p=p+VAL("&"d$)
170 IF i<e THEN NEXT i
180 typ=PEEK(&BD71):IF typ=&E8 THEN 260
190 IF typ=&55 THEN 220
200 IF typ=&14 THEN 240
210 PRINT"CPC nicht erkannt !":END
220 POKE &A331,&D4:POKE &A338,&69:POKE &
A339,&E8:POKE &A399,&D4,&A39D,&AA:POKE &
A39E,&E7:POKE &A3C0,&2:POKE &A3C1,&AC
230 POKE &A3CF,&5B:POKE &A3E0,&5B:POKE &
A3F2,&2:POKE &A3F3,&AC:POKE &A3EF,&8A:GO
TO 260
240 POKE &A331,&CF:POKE &A338,&64:POKE &
A339,&E8:POKE &A399,&CF:POKE &A39D,&A5:P
OKE &A39E,&E7:POKE &A3C0,&2:POKE &A3C1,&
AC
250 POKE &A3CF,&5E:POKE &A3E0,&5E:POKE &
A3F2,&2:POKE &A3F3,&AC:POKE &A3EF,&8A
260 CALL &A300:END
1001 DATA 01,0A,A3,21,19,A3,CD,D1,&0329
1002 DATA BC,C9,12,A3,C3,C3,A3,C3,&0526
1003 DATA D4,A3,45,49,CE,41,55,D3,&043C
1004 DATA 00,00,00,00,00,CD,BF,A3,&022F
1005 DATA F5,C5,D5,E5,2A,EF,A3,28,&0558
1006 DATA 6A,E5,CD,98,A3,E1,30,63,&04CB
1007 DATA CD,04,EE,ED,53,EB,A3,CD,&055A
1008 DATA A3,E7,22,E5,A3,ED,43,E7,&054B
1009 DATA A3,09,22,E9,A3,21,00,00,&027B
1010 DATA 22,ED,A3,3A,E7,A3,32,EE,&0496
1011 DATA A3,D6,05,47,DD,21,40,00,&0303
1012 DATA B7,DD,7E,00,5F,3A,ED,A3,&043B
1013 DATA 83,B7,07,32,ED,A3,DD,23,&0403
1014 DATA 10,EE,2A,EB,A3,B7,85,B7,&04A9
1015 DATA 07,B7,84,32,ED,A3,2A,ED,&041B
1016 DATA A3,3E,20,CD,5A,BB,CD,5A,&040A
1017 DATA BB,3E,5B,CD,5A,BB,7C,CD,&047F
1018 DATA A2,A3,7D,CD,A2,A3,3E,5D,&046F
1019 DATA CD,5A,BB,E1,D1,C1,F1,C9,&060F
1020 DATA CD,04,EE,D0,CD,D2,E6,37,&054B
1021 DATA 9F,C9,5F,0F,0F,0F,0F,E6,&02E9
1022 DATA 0F,CD,B3,A3,7B,E6,0F,CD,&046F
1023 DATA B3,A3,C9,FE,0A,38,02,C6,&0427
1024 DATA 07,C6,30,CD,5A,BB,C9,CF,&0477
1025 DATA 98,AA,F7,3E,FF,32,00,AC,&0454
1026 DATA 01,03,00,21,F4,A3,11,3A,&0207
1027 DATA BD,ED,B0,C9,3E,00,32,00,&0393
1028 DATA AC,01,03,00,21,F1,A3,11,&0276
1029 DATA 3A,BD,ED,B0,C9,00,00,00,&035D
1030 DATA 00,00,00,00,00,00,00,A4,&00A4
1031 DATA AC,CF,98,AA,C3,1D,A3,&0440
```

Privat-Manager

Und wieder ein Leckerbissen für alle Leser: Ein „Privat-Manager“ für den aktiven User, der mehr aus seinem CPC herausholen will.

In diesem Programm unseres Autors Franz-Josef Wahl sind Routinen integriert, die sich sehen lassen können. Sie haben vom Menü aus folgende Optionen zur Wahl:

1. Adreßdatei

Speichern von bis zu 100 Adressen mit Name, Wohnort, Telefonnummer und Geburtstag. Dazu die Möglichkeiten zum Editieren, Suchen, Löschen sowie die Ausgabe auf dem Drucker oder dem Bildschirm. Falls der Geburtstag nicht eingegeben werden soll, bitte die Punktstaste auf dem Zehnerblock drücken.

Eine Adresse kann geschützt werden, wenn vor dem Namen ein „!“ eingegeben wird. Bei der Ausgabe wird dieser Datensatz nicht angezeigt. Ausgegeben werden kann diese Anschrift nur über die Suchroutine. Sie müssen dann natürlich auch das „!“ dem Namen vorausstellen.

2. Kalenderberechnung

Ausgabe eines beliebigen Jahreskalenders oder Wochentagsberechnungen.

3. Finanzprogramm

Ein Haushaltsbuch in kompakter Version. Es berechnet Einnahmen, Ausgaben und sämtliche Quersummen. Anzeige und Berechnung von beliebigen Monatsbereichen eines Jahres sind ebenfalls möglich. Eingabe: Zuerst mit den Cursortasten das entsprechende Feld anwählen, dann Enter drücken. Nun wieder mit den Cursortasten die Spalten anwählen, wieder mit Enter bestätigen, dann die Zahleneingabe machen und ebenfalls Enter drücken!

4. Taschenrechner

Alle Grundrechenarten. Einfach mit Cursortasten anklicken.

5. Notizzettel

Zehn kleine Notizen können im Speicher gehalten und wieder aufgerufen werden.

6. Speichern und Laden

Die Daten werden abgespeichert oder eingelesen.

7. Farben

Das Ändern der vier Grundfarben ist möglich.

8. Cat

Ausgabe des Directory sowie Ändern des Userbereichs.

Das Programm läuft auf allen CPC-Typen, auf dem 464 allerdings nur mit Floppy. JE

```

10 '***** <2397>
15 '* PRIVAT-MANAGER * <2324>
20 '* VON * <239D>
25 '* FRANZ-JOSEF WAHL * <230F>
30 '* FUER * <238E>
35 '* SCHNEIDER CPC-WELT * <23A6>
40 '* CPC 464/664/6128 JE* <23FA>
45 '***** <23DD>

```

```

50 OPENOUT"dummy":MEMORY HIMEM-1:C
LOSEOUT <15BF>
60 '...initial <11F6>
70 INK 0,0:INK 1,24:INK 2,6:INK 3,
0,18:PAPER 0:PEN 1:BORDER 0:q=0:KE
Y 138,"",""+CHR$(13) <3C03>
80 CLEAR:DIM M(12),W$(7),K$(7),a$(
311),mo$(12),zahl$(2),zahl(2),pa$(
16),am$(16,12),pe$(4),em$(4,12),mh
$(12),es(16),ge(12),ga(12):FOR N=1
TO 100:A$(N)="!":NEXT:U$=STRING$(
&28,"-") <CB30>
90 FOR n=301 TO 310:a$(n)="!":NEXT
<238B>
100 SPEED INK 30,30 <0C75>
110 DATA 31,28,31,30,31,30,31,31,3
0,31,30,31,MONTAG,MO,DIENSTAG,DI,M
ITTWOCH,MI,DONNERSTAG,DO,FREITAG,F
R,SAMSTAG,SA,SONNTAG,SO,JA,FE,MA,A
P,MA,JU,JL,AU,SE,OK,NO,DE <9FA6>
120 FOR N=1 TO 12:READ M(N):NEXT:F
OR N=1 TO 7:READ W$(N),K$(N):NEXT <4019>
130 FOR N=1 TO 12:READ MO$(N):NEXT
<1EDE>
140 DATA "JAN. ", "FEB. ", "
MAERZ ", "APRIL ", "MAI "
", "JUNI ", "JULI ", "AUGUST
", "SEPT. ", "OKT. ", "NOV.
", "DEZ. " <9624>
150 FOR n=1 TO 12:READ mh$(n):NEXT
<1E80>
160 wt$=" MO DI MI DO FR SA SO M
O DI MI DO FR SA SO MO DI MI DO FR
SA SO " <4FA6>
170 MODE 1:INPUT"Datum (T,M,J)",GT
$,GM$,GJ$:IF VAL(gt$)<1 OR VAL(gt$
)>31 OR VAL(GM$)<1 OR VAL(GM$)>12
OR VAL(GJ$)<1 THEN 170 ELSE gt=VAL
(gt$):gm=VAL(gm$):gj=VAL(gj$):y$=g
t$+"."+gm$+"."+gj$ <BD64>
180 IF GM>29 AND GM=2 THEN 180 ELS
E GOSUB 2660:T$=W0$:ta=gt:ma=gm:ja
=gj <4FA7>
190 '...hauptmenue <1457>
200 MODE 1:PEN 1:PAPER 0:PRINT CHR
$(24);" PRIVAT-MANAGER VON FJW-SOF
TWARE V 3.0 "CHR$(24);:PRINT <4969>
210 PEN 2:PRINT"HAUPTMENUE STAND:
"a$(311):PRINT:PEN 1:PRINT" 1 KALE
NDERBERECHNUNGEN":PRINT" 2 ADRESSE
N":PRINT" 3 NOTIZEN":PRINT" 4 FINA
NZEN":PRINT" 5 RECHNER":PRINT" 6 S
PEICHERN":PRINT" 7 LADEN":PRINT" 8
FARBEN":PRINT" 9 CATalog":PRINT:P
RINT" E Ende" <C4C5>
220 LOCATE 2,17:PEN 3:PRINT"Bitte
waehlen !"CHR$(7):LOCATE 1,25:PEN
1:PRINT CHR$(24)" ";T$" DEN"ta"."m

```

LISTINGS

```

a" "ja;CHR$(24) <60DE>
230 s$=INKEY$:IF s$=""THEN 230 ELS
E PEN 1 <1D3A>
240 IF UPPER$(s$)="E"THEN 2780 ELS
E N=VAL(s$) <25DB>
250 ON N GOTO 270,570,1300,1480,19
20,2280,2400,2580,2720 <2E47>
260 ' ...kalender <13E0>
270 MODE 1:PEN 2:PRINT"KALENDERBER
ECHNUNGEN":PRINT u$:PEN 1:PRINT:PR
INT"[1] JAHRESBERECHNUNG":PRINT,,,
"[2] TAGESBERECHNUNG":PRINT,,, "[3]
HAUPTMENUE":PEN 3:PRINT,,, "BITTE
WAEHLN !" <8DF9>
280 S$=INKEY$:IF S$=""THEN 280 ELS
E n=VAL(S$) <288D>
290 ON n GOTO 310,510,200 <169C>
300 ' ...jahre <0F3C>
310 MODE 2:PEN 1:sb=0:INPUT"Welche
s Jahr";gj:IF gj<1 THEN 310 ELSE g
t=1:gm=1:GOSUB 2660 <4BD2>
320 CLS:PRINT CHR$(24);WT$;CHR$(24
):FOR n=1 TO 12:LOCATE 1,n+n:PRINT
mo$(n):PRINT:NEXT <44FF>
330 GOSUB 2660:a=0+(3*t):c=2 <20AE>
340 FOR n=1 TO 31:LOCATE a,c:PRINT
n:a=a+3:IF a>65 THEN a=3:c=c+1 <48A8>
350 NEXT <0680>
360 gm=2:GOSUB 2660:a=0+(3*t):c=c+
1:FOR n=1 TO sa:LOCATE a,c:PRINT n
:a=a+3:IF a>65 THEN a=3:c=c+1 <749E>
370 NEXT <06A8>
380 FOR mo=3 TO 12 <10C6>
390 GOSUB 480 <09E6>
400 FOR n=1 TO m(mo):LOCATE a,c:PR
INT n:a=a+3:IF a>65 THEN a=3:c=c+1
<51F7>
410 NEXT:NEXT <0806>
420 FOR n=44 TO 524 STEP 24:MOVE n
,400:DRAW n,1:NEXT <2885>
430 WINDOW 67,80,1,25:PRINT CHR$(2
4):CLS <1AA6>
440 PRINT" * "gj" *":PRINT:IF sb=1
THEN PRINT" Schaltjahr -----
----" <3CCE>
450 PRINT:PRINT" 1 Menue":PRINT:PR
INT" 2 Neues Jahr":PRINT:PRINT" Wa
ehlen Sie -----"CHR$(24) <4BF8>
460 s$=INKEY$:IF s$=""THEN 460 ELS
E s=VAL(s$) <28D7>
470 ON s GOTO 270,310 <1202>
480 c=c+1:gm=mo:GOSUB 2660:a=0+(3*
t):RETURN <33E3>
490 GOTO 490 <091E>
500 ' ...tage <0E94>
510 MODE 1:WINDOW#1,1,40,5,20:PEN
2:PRINT"WOCHENTAGBERECHNUNG":PRINT
U$:LOCATE 1,25:PRINT"Punkt auf Za

```

```

hlenblock = MENUE":WINDOW SWAP 1:P
EN 1 <6376>
520 INPUT"TAG,MONAT,JAHR";GT,GM,GJ
:IF GT=0 AND GM=0 AND GJ=0 THEN 27
0 ELSE IF GT<1 OR GT>31 OR GM<1 OR
GM>12 OR GJ<1 THEN 520 <758D>
530 IF GT>29 AND GM=2 THEN 80 <1A48>
540 GOSUB 2660:PEN 2:PRINT WO$:PEN
1:GOTO 520 <1B6F>
550 ' <0759>
560 ' ...adressen <12D9>
570 MODE 1:PEN 2:z=0:FOR N=1 TO 10
0:IF a$(n)="!"THEN z=z+1 <38F6>
580 NEXT <064D>
590 PRINT"ADRESSEN FR
EIPLAETZE":z:PRINT U$ <347B>
600 PEN 1:PRINT,,, "[1] NEUER EINT
RAG":PRINT,,, "[2] LOESCHEN":PRINT
,,, "[3] SUCHEN":PRINT,,, "[4] INH
ALT AUF MONITOR":PRINT,,, "[5] INH
ALT AUF DRUCKER":PRINT,,, "[6] SOR
TIEREN":PRINT,,, "[7] HAUPTMENUE" <AD2C>
610 S$=INKEY$:IF S$=""THEN 610 ELS
E A=VAL(S$) <2869>
620 IF A<1 OR A>7 THEN 610 ELSE ON
A GOTO 640,730,780,850,950,1070,2
00 <3A77>
630 'eintrag <0ED3>
640 MODE 1:PEN 2:PRINT"NEUER EINTR
AG":PRINT U$:WINDOW#1,1,40,3,25:WI
NDOW SWAP 1 <3395>
650 Z=1 <0BEC>
660 IF LEFT$(A$(Z),1)="!"AND LEN(A
$(Z))=1 THEN 680 ELSE Z=Z+1:IF Z=1
01 THEN PRINT"Adressendeitei voll":
CALL &BB06:GOTO 570 <659F>
670 GOTO 660 <09D9>
680 PEN 1:PRINT,,, "Nr."z:PRINT:INP
UT"Name";a$(z):a$(z)=UPPER$(a$(z))
:INPUT"Adresse";a$(z+100):a$(z+100
)=UPPER$(a$(z+100)):INPUT"Telefon"
;a$(z+200):a$(z+200)=UPPER$(a$(z+2
00)) <B6CA>
690 INPUT"Geboren (TT,MM,JJJJ)";s1
$,s2$,s3$:IF s1$=""THEN 700 ELSE s
$=s1$+"."+s2$+"."+s3$:IF LEN(s$)<
10 THEN 690 ELSE a$(z+200)=s$+a$(z
+200) <8F20>
700 PEN 2:PRINT,,, "Weitere Eingabe
n?":PEN 1 <2315>
710 S$=INKEY$:IF UPPER$(S$)="J"THE
N 660 ELSE IF UPPER$(S$)="N"THEN 5
70 ELSE 710 <3606>
720 ' ...loeschen <128C>
730 MODE 1:PEN 2:PRINT"LOESCHEN":P
RINT u$:LOCATE 1,25:PRINT"ENTER-Me
nue":WINDOW#1,1,40,5,20:WINDOW SWA
P 1 <433D>

```

```

740 CLS:PEN 1:LOCATE 1,1:INPUT"Adr
essennummer";z:IF z=0 THEN 570 ELS
E IF z>100 THEN 740 ELSE PRINT:GOS
UB 1010:PEN 2:PRINT,,,"Loeschen ?"
:PEN 1                                     <5E7B>
750 s$=INKEY$:IF s$=""THEN 750 ELS
E IF UPPER$(s$)="J"THEN a$(Z)="!":
a$(Z+100)="" :a$(Z+200)=""             <59EA>
760 GOTO 740                               <090F>
770 '...suchen                             <10BE>
780 MODE 1:PEN 2:PRINT"ADRESSE SUC
HEN":PRINT u$:WINDOW#1,1,40,3,25:W
INDOW SWAP 1                               <34B8>
790 CLS:PEN 1:LOCATE 1,1:INPUT"Ges
uchter Name";g$:a=LEN(g$):PRINT:g$
=UPPER$(g$)                               <43C7>
800 z=0:FOR n=1 TO 100:IF LEFT$(a$
(n),a)=g$THEN z=n:GOSUB 1010            <3FE1>
810 NEXT:IF z=0 THEN PRINT,,,"Nicht
vorhanden"                               <24DA>
820 PEN 2:PRINT,,,"Neue Suche ?":P
EN 1                                       <1D8B>
830 s$=INKEY$:IF s$=""THEN 830 ELS
E IF UPPER$(s$)="J"THEN 790 ELSE 5
70                                         <314B>
840 '...inhalt                             <100F>
850 MODE 1:PEN 2:PRINT"GESAMTINHAL
T":PRINT U$:LOCATE 1,25:PRINT"M-en
ue ENTER-Weiter L-oeschen D-ruc
k":WINDOW#1,1,40,4,23:WINDOW SWAP
1                                           <6327>
860 PEN 1:N=0:FOR Z=1 TO 100:IF LE
FT$(a$(Z),1)="!"THEN 880 ELSE GOSU
B 1010:PRINT:N=N+1                       <4702>
870 S$=INKEY$:IF S$=""THEN 870 ELS
E IF UPPER$(S$)="M"THEN 570 ELSE I
F UPPER$(s$)="L"THEN a$(z)="!":PEN
2:PRINT"Nr."z" geloescht":PEN 1:P
RINT ELSE IF UPPER$(s$)="D"THEN GO
SUB 900 ELSE 880                         <8258>
880 NEXT                                   <06A6>
890 CALL &BB06:GOTO 570                   <0EA7>
900 PRINT CHR$(7):PRINT#8,:PRINT#8
,"NAME: "a$(z)                           <29BE>
910 IF a$(z+100)<>""THEN PRINT#8,"
ADRESSE:"a$(z+100)                       <32B2>
920 IF a$(z+200)=""THEN 930 ELSE I
F MID$(a$(z+200),3,1)=". "THEN PRIN
T#8,"TEL.: "MID$(a$(z+200),11)EL
SE PRINT#8,"TEL.: "a$(z+200)           <7409>
930 IF MID$(a$(z+200),3,1)=". "THEN
PRINT#8,"GEB.: "LEFT$(a$(z+200)
,10)                                       <41CB>
940 RETURN                                 <0681>
950 '...drucker                            <1127>
960 MODE 1:PEN 1:PRINT"BITTE WARTE
N":FOR z=1 TO 100:IF LEFT$(a$(z),1)
)<>""!THEN PRINT#8,:PRINT#8,"NAME:
a$(z)ELSE 1000                           <5BBF>
970 IF a$(z+100)<>""THEN PRINT#8,"
ADRESSE:"a$(z+100)                       <3229>
980 IF a$(z+200)=""THEN 990 ELSE I
F MID$(a$(z+200),3,1)=". "THEN PRIN
T#8,"Tel.:"MID$(a$(z+200),11)ELSE
PRINT#8,"Tel.:"a$(z+200)               <6E01>
990 IF MID$(a$(z+200),3,1)=". "THEN
PRINT#8,"Geboren:";LEFT$(a$(z+200)
),10)                                       <4250>
1000 NEXT:GOTO 570                        <0B13>
1010 '...list-rout.                       <142D>
1020 PEN 1:PRINT:PRINT"Nr."Z:PEN 2
:PRINT"Name:";:PEN 1:PRINT a$(Z)       <309E>
1030 PEN 2:IF a$(z+100)<>""THEN PR
INT"Adresse:";:PEN 1:PRINT a$(Z+10
0)                                         <38F4>
1040 PEN 2:IF a$(z+200)=""THEN 105
0 ELSE IF MID$(a$(z+200),3,1)=". "T
HEN PRINT"Tel.:";:PEN 1:PRINT MID$
(a$(Z+200),11)ELSE PRINT"Tel.:";:P
EN 1:PRINT a$(z+200)                   <777E>
1050 PEN 2:IF MID$(a$(z+200),3,1)=
". "THEN PRINT"Geboren:";:PEN 1:PRI
NT LEFT$(a$(z+200),10)                   <470F>
1060 RETURN                               <0673>
1070 '---SORTIEREN                       <1398>
1080 MODE 1:PEN 3:LOCATE 8,10:PRIN
T"Sortiervorgang laeuft !"             <2BCD>
1090 N%=100:Z%=1:BY(1)=1:BY(2)=N%     <2A90>
1100 LI%=BY(Z%):RE%=BY(Z%+1):Z%=Z%
-2                                         <34A6>
1110 I%=LI%:J%=RE%                       <1A13>
1120 M%=INT((LI%+RE%)/2):GW$=A$(M%
)                                         <2EDF>
1130 IF A$(I%)>GW$THEN 1150             <1A62>
1140 I%=I%+1:GOTO 1210                  <150B>
1150 IF A$(J%)<=GW$THEN 1170           <1AED>
1160 J%=J%-1:GOTO 1210                  <15ED>
1170 IF I%>J%THEN 1210                  <1312>
1180 AA$=A$(I%):BB$=A$(I%+100):CC$
=A$(I%+200):A$(I%)=A$(J%):A$(I%+10
0)=A$(J%+100):A$(I%+200)=A$(J%+200
):A$(J%)=AA$:A$(J%+100)=BB$:A$(J%+
200)=CC$                                   <C4F4>
1190 I%=I%+1                             <1061>
1200 J%=J%-1                             <108B>
1210 IF I%<=J%THEN 1130                 <130C>
1220 IF I%>=RE%THEN 1240                <14D2>
1230 Z%=Z%+2:BY(Z%)=I%:BY(Z%+1)=RE
%                                           <3317>
1240 RE%=J%                               <0F9F>
1250 IF LI%<RE%THEN 1110                 <158E>
1260 IF Z%<>-1 THEN 1100                 <11C6>
1270 z=0:FOR n=1 TO 100:IF a$(n)<>
"! "THEN erste=n:GOTO 1280 ELSE NEX
T                                           <3C6E>
1280 z=1:FOR n=erste TO 100:a$(z)=

```

LISTINGS

```

a$(n):a$(z+100)=a$(n+100):a$(z+200
)=a$(n+200):z=z+1:NEXT          <79E9>
1290 FOR n=erste TO 100:a$(n)="!":
NEXT:GOTO 570                      <2C40>
1300 '...notizen                   <11BB>
1310 MODE 1:PEN 2:PRINT"NOTIZEN":P
RINT u$:WINDOW#1,20,35,4,21:WINDOW
#2,20,35,22,22:n=0                <41F3>
1320 PEN 1:PRINT"1 - Blaettern":PR
INT,,,"2 - Loeschen":PRINT,,,"3 -
Eintrag":PRINT,,,"4 - Hauptmenue":
PEN 3:PRINT,,,"Bitte waehlen!":WI
NDOW SWAP 1:PEN 2:PRINT CHR$(24):C
LS                                  <7C22>
1330 s$=INKEY$:IF s$=""THEN 1330 E
LSE a=VAL(s$):CLS:INK 3,q          <3251>
1340 IF a<1 OR a>4 THEN 1330 ELSE
IF a=4 THEN INK 3,0,18:PRINT CHR$(
24)                                 <3096>
1350 ON a GOSUB 1380,1410,1430,200
                                     <1A6A>
1360 INK 3,0,18:GOTO 1330          <11D2>
1370 '...blaettern                 <13D9>
1380 IF n=10 THEN n=1 ELSE n=n+1  <211B>
1390 PRINT a$(300+n):PRINT#2,"Nr."
n:RETURN                            <2493>
1400 '---loeschen---              <151B>
1410 a$(300+n)="!":CLS:PRINT#2,"Nr
."n"geloescht":RETURN              <34F1>
1420 '...eintrag                   <11D9>
1430 n=1                             <0B49>
1440 IF a$(300+n)="!"THEN 1460 ELS
E n=n+1                              <2926>
1450 IF n=11 THEN PRINT#2,"Alles v
oll":n=0:RETURN ELSE 1440          <2CFC>
1460 CLS:PRINT#2,"Eintrag in"n:INP
UT a$(300+n):RETURN                <2D56>
1470 '...finanzen                 <12FB>
1480 KEY 138,"":MODE 2:PEN 1:WIND
OW#1,1,80,1,24:PAPER#1,0:PEN#1,1:W
INDOW#2,1,80,25,25:PAPER#2,1:PEN#2
,0:CLS#2:GOSUB 1870:WINDOW SWAP 1 <541C>
1490 mo1=1:mo2=6:a=1:b=2:GOSUB 151
0:GOSUB 1590:GOTO 1650             <334A>
1500 '...bildaufbau               <14CD>
1510 a=1:b=2:CLS:PRINT CHR$(24);"A
usgaben ";:FOR n=mo1 TO mo2:P
RINT mh$(n);:NEXT:PRINT"GESAMT
"CHR$(24);                          <6768>
1520 FOR n=1 TO 16:LOCATE 2,n+1:PR
INT pa$(n):NEXT                    <2824>
1530 m=15:FOR n=mo1 TO mo2:FOR z=1
TO 16:LOCATE m,z+1:PRINT USING"##
##.##";VAL(am$(z,n)):NEXT:m=m+9:NE
XT                                  <69EF>
1540 PRINT CHR$(24);"Einnahmen "CH
R$(143)CHR$(143)"
                                     "CHR$(24)
                                     <6664>
1550 FOR n=1 TO 4:LOCATE 2,n+18:PR
INT pe$(n):NEXT:m=15:FOR n=mo1 TO
mo2:FOR z=1 TO 4:LOCATE m,z+18:PRI
NT USING"####.##";VAL(em$(z,n)):NE
XT:m=m+9:NEXT                      <8D13>
1560 PRINT CHR$(24);"Gesamt :
                                     "CHR$(
(24));:IF m=69 THEN 1570 ELSE WINDO
W#4,m+10,80,1,24:CLS#4             <881B>
1570 PRINT"UEBRIG ":po=m:la=po-11
:cp=mo1:RETURN                     <3942>
1580 '...berechnen                <1326>
1590 er=0:FOR n=1 TO 16:FOR z=mo1
TO mo2:er=er+VAL(am$(n,z)):NEXT:es
(n)=er:LOCATE po,n+1:PRINT USING"#
###.##";er:er=0:NEXT              <8A93>
1600 er=0:eg=0:z=15:PRINT CHR$(24)
;:FOR n=mo1 TO mo2:FOR m=1 TO 16:e
r=er+VAL(am$(m,n)):NEXT:LOCATE z,1
8:PRINT USING"####.##";er:ga(n)=er
:eg=eg+er:er=0:z=z+9:NEXT:LOCATE z
,18:PRINT USING"####.##";eg:PRINT
CHR$(24);                          <E077>
1610 er=0:FOR n=1 TO 4:FOR m=mo1 T
O mo2:er=er+VAL(em$(n,m)):NEXT:es(
n)=er:er=0:LOCATE z,n+18:PRINT USI
NG"####.##";es(n):NEXT           <8F40>
1620 eg=0:er=0:z=15:PRINT CHR$(24)
;:FOR n=mo1 TO mo2:FOR m=1 TO 4:er
=er+VAL(em$(m,n)):NEXT:LOCATE z,23
:PRINT USING"####.##";er:ge(n)=er
:eg=eg+er:er=0:z=z+9:NEXT:LOCATE z
,23:PRINT USING"####.##";eg:PRINT C
HR$(24);                          <DFAD>
1630 eg=0:z=15:FOR n=mo1 TO mo2:er
=ge(n)-ga(n):LOCATE z,24:PRINT USI
NG"####.##";er:eg=eg+er:z=z+9:NEXT
:LOCATE z,24:PRINT USING"####.##";
eg:PRINT CHR$(7);:RETURN          <A60C>
1640 '...inkey$                   <1016>
1650 s$=INKEY$:IF s$=""THEN 1650 E
LSE IF s$(<)CHR$(13)THEN SOUND 1,30
,2                                  <2F04>
1660 IF UPPER$(s$)="N"THEN PRINT C
HR$(7);:GOSUB 1590                 <1FED>
1670 IF UPPER$(s$)="Z"THEN PRINT C
HR$(7);:GOSUB 1820                 <1FEE>
1680 IF UPPER$(s$)="E"THEN KEY 138
,,,"+CHR$(13):GOTO 200             <2722>
1690 IF UPPER$(s$)="H"THEN GOSUB 1
900                                  <17DF>
1700 IF s$=CHR$(240)THEN IF b=2 TH
EN 1650 ELSE LOCATE a,b:PRINT" ":I
F b=19 THEN b=17:GOSUB 1810 ELSE b
=b-1:GOSUB 1810                    <56B8>
1710 IF s$=CHR$(241)THEN IF b=22 T

```

```

HEN 1650 ELSE LOCATE a,b:PRINT" ":
IF b=17 THEN b=19:GOSUB 1810 ELSE
b=b+1:GOSUB 1810 (57A7)
1720 IF s$=CHR$(242)THEN IF a=1 TH
EN 1650 ELSE LOCATE a,b:PRINT" ":I
F a=13 THEN a=1:GOSUB 1810 ELSE a=
a-9:c p=c p-1:GOSUB 1810 (635B)
1730 IF s$=CHR$(243)THEN IF a=1a T
HEN 1650 ELSE LOCATE a,b:PRINT" ":
IF a=1 THEN a=13:GOSUB 1810 ELSE a
=a+9:c p=c p+1:GOSUB 1810 (6744)
1740 IF s$=CHR$(13)THEN 1760 ELSE
1650 (1A42)
1750 '...eingabe (11C4)
1760 IF a=1 AND b<18 THEN WINDOW#3
,1,12,b,b:PAPER#3,1:PEN#3,0:CLS#3:
INPUT#3,pa$(b-1):IF LEN(pa$(b-1))>
12 THEN 1760 ELSE PAPER#3,0:PEN#3,
1:CLS#3:LOCATE 2,b:PRINT pa$(b-1):
GOSUB 1810:GOTO 1650 (95C1)
1770 IF a=1 AND b>18 THEN WINDOW#3
,1,12,b,b:PAPER#3,1:PEN#3,0:CLS#3:
INPUT#3,pe$(b-18):IF LEN(pe$(b-18)
)>12 THEN 1770 ELSE PAPER#3,0:PEN#
3,1:CLS#3:LOCATE 2,b:PRINT pe$(b-1
8):GOSUB 1810:GOTO 1650 (9816)
1780 IF b<18 THEN WINDOW#3,a+1,a+1
0,b,b:PAPER#3,1:PEN#3,0:CLS#3:INPU
T#3,am$(b-1,c p):IF VAL(am$(b-1,c p)
)>9999 THEN 1780 ELSE PAPER#3,0:PE
N#3,1:CLS#3:LOCATE a+2,b:PRINT USI
NG"####.##";VAL(am$(b-1,c p)):GOTO
1650 (BA9E)
1790 WINDOW#3,a+1,a+10,b,b:PAPER#3
,1:PEN#3,0:CLS#3:INPUT#3,em$(b-18,
c p):IF VAL(em$(b-18,c p))>9999 THEN
1790 ELSE PAPER#3,0:PEN#3,1:CLS#3
:LOCATE a+2,b:PRINT USING"####.##"
;VAL(em$(b-18,c p)):GOTO 1650 (B434)
1800 '...cursor (10C1)
1810 LOCATE a,b:PRINT">":RETURN (165C)
1820 '...zeitraumwechsel (1909)
1830 CLS#2:INPUT#2,"Erster Anzeig
emonat (1-12)";mo1:IF mo1<1 OR mo1
>12 THEN 1830:CLS#2 (4D65)
1840 INPUT#2,"Zweiter Anzeigemonat
";mo2:IF mo2<mo1 OR mo2>12 THEN 18
40 (4392)
1850 IF mo2-mo1>5 THEN PRINT#2,"Nu
r 6 Monate koennen angezeigt werde
n":FOR n=1 TO 2000:NEXT:GOTO 1830 (544D)
1860 GOSUB 1870:GOSUB 1510:GOSUB 1
590:RETURN (15E3)
1870 '...menue (0F3C)
1880 CLS#2:PRINT#2,"E-ende N-neu
berechnung Z-zeitraumwechsel H-h
ilfe":RETURN (459A)
1890 '...hilfe (0F0D)
1900 PRINT#2,"Cursor mit Pfeiltas
ten bewegen.Zur Eingabe ENTER.Eing
abe mit ENTER beenden.":CALL &BB06
:GOSUB 1880:RETURN (6363)
1910 MODE 2:RETURN (09D3)
1920 '...rechner (118C)
1930 MODE 0:a=2:b=10:GOSUB 2220 (1B97)
1940 stand=2:PRINT#1,"0." (18FA)
1950 KEY DEF 29,1,42:KEY DEF 28,1,
43:KEY DEF 9,1,99 (2127)
1960 s$=INKEY$:IF s$=""THEN 1960 E
LSE IF s$="c"THEN s$=s1$:GOTO 2060
ELSE LOCATE a,b:PRINT" " (4599)
1970 IF ASC(s$)=240 AND b>10 THEN
b=b-2:rb=rb-1:GOTO 1990 (380E)
1980 IF ASC(s$)=240 AND b=10 THEN
b=20:rb=6 (290A)
1990 IF ASC(s$)=241 AND b<20 THEN
b=b+2:rb=rb+1:GOTO 2010 (3847)
2000 IF ASC(s$)=241 AND b=20 THEN
b=10:rb=1 (295A)
2010 IF ASC(s$)=242 AND a>2 THEN a
=a-2:ra=ra-1:GOTO 2030 (3740)
2020 IF ASC(s$)=242 AND a=2 THEN a
=6:ra=3 (275D)
2030 IF ASC(s$)=243 AND a<6 THEN a
=a+2:ra=ra+1:GOTO 2050 (37B4)
2040 IF ASC(s$)=243 AND a=6 THEN a
=2:ra=1 (2766)
2050 s1$=zeich$(rb,ra):LOCATE a,b:
PRINT CHR$(243):GOTO 1960 (3870)
2060 IF po=1 AND ASC(s$)>47 AND AS
C(s$)<58 THEN zahl$(1)="" :SOUND 1,
37,4:zahl$(2)="" :stand=2:po=0 (5CB0)
2070 IF ASC(s$)<48 OR ASC(s$)>57 T
HEN GOTO 2080 ELSE IF LEN(zahl$(st
and))>12 THEN 1940 ELSE zahl$(stan
d)=zahl$(stand)+s$:PRINT#1,zahl$(s
tand):zahl$(stand)=VAL(zahl$(stand)
):SOUND 1,37,4:GOTO 1960 (B6DA)
2080 IF ASC(s$)=45 THEN x$="-" :SOU
ND 1,37,4 (22E7)
2090 IF ASC(s$)=47 THEN x$="/" :SOU
ND 1,37,4 (226E)
2100 IF ASC(s$)=43 THEN x$="+" :SOU
ND 1,37,4 (222E)
2110 IF ASC(s$)=42 THEN x$="*" :SOU
ND 1,37,4 (22B0)
2120 IF UPPER$(s$)="C"THEN SOUND 1
,50,10:GOTO 1940 ELSE IF UPPER$(s$
)="E"THEN KEY DEF 28,1,59:KEY DEF
29,1,58:PAPER#1,0:PEN#1,1:SOUND 1,
37,4:GOTO 200 (5C18)
2130 IF s$=""THEN 2170 (129B)
2140 IF ASC(s$)=46 THEN zahl$(stan
d)=zahl$(stand)+".":SOUND 1,37,4:P
RINT#1,zahl$(stand):GOTO 1960 (5C64)
2150 IF po=0 THEN zahl(1)=zahl(2) (2353)

```

LISTINGS

```

2160 stand=2:zahl$(2)="" :GOTO 1960
                                     <2287>
2170 SOUND 1,37,4:PO=1:IF z$="-"TH
EN e=zahl(1)-zahl(2):zahl(1)=e:sta
nd=2:PRINT#1,e:GOTO 1960          <62AA>
2180 IF z$="/"THEN e=zahl(1)/zahl(
2):zahl(1)=e:stand=2:PRINT#1,e:GOT
O 1960                             <52F8>
2190 IF z$="+"THEN e=zahl(1)+zahl(
2):zahl(1)=e:stand=2:PRINT#1,e:GOT
O 1960                             <5210>
2200 IF z$="*"THEN e=zahl(1)*zahl(
2):zahl(1)=e:stand=2:PRINT#1,e:GOT
O 1960                             <52BB>
2210 '                               <0756>
2220 RESTORE 2250:MODE 0:WINDOW#1,
1,20,1,1:PEN#1,0:PAPER#1,1:CLS#1:F
OR n=1 TO 18:READ a,b,s$:LOCATE a,
b:PRINT CHR$(24);s$;CHR$(24):NEXT <65C5>
2230 FOR n=3 TO 25:PEN 2:LOCATE 11
,n:PRINT"/":NEXT:LOCATE 13,3:PRINT
"RECHNER":LOCATE 13,4:PRINT"-----
-":LOCATE 13,9:PRINT"E ENDE":LOCAT
E 13,11:PRINT"C NEU":LOCATE 16,14:
PRINT CHR$(240):LOCATE 15,15:PRINT
CHR$(242)" "CHR$(243):LOCATE 16,1
6:PRINT CHR$(241)                 <9A23>
2240 FOR n=1 TO 6:FOR m=1 TO 3:REA
D zeich$(n,m):NEXT:NEXT:a=2:b=10:r
a=1:rb=1:RETURN                  <52B0>
2250 DATA 3,20,0,5,20,.,7,20,=,3,1
8,1,5,18,2,7,18,3,3,16,4,5,16,5,7,
16,6,3,14,7,5,14,8,7,14,9,3,12,+ ,5
,12,-,7,12,* ,3,10,/ ,5,10,E,7,10,C <840F>
2260 DATA / ,E,C,+ ,-,*,7,8,9,4,5,6,
1,2,3,0,.,=                       <2A9B>
2270 '...speichern                <1358>
2280 MODE 1:PEN 2:PRINT"SPEICHERN"
:PRINT u$:a$(311)=y$              <2C19>
2290 PRINT,,,"Normal speichern (Ak
tualisieren) - ENTER":PRINT,,,"Spe
ichern unter eigenem Namen ---- TA
STE"                               <62FC>
2300 s$=INKEY$:IF s$=""THEN 2300 E
LSE IF s$("<") "THEN 2330         <2832>
2310 CLS:INPUT"Unter welchem Namen
Speichern";s$:IF LEN(s$)>8 THEN 2
310                                 <3C35>
2320 CLS:PRINT"Daten werden gespei
chert unter:":PRINT s$+".DAT --- B
itte warten !":OPENOUT s$+".DAT":G
OTO 2340                            <5B69>
2330 CLS:PRINT"Daten werden aktual
isiert.Bitte warten !":OPENOUT"pm-
data.dat"                           <411C>
2340 FOR n=1 TO 311:PRINT#9,a$(n):
NEXT                                <2197>
2350 FOR n=1 TO 16:PRINT#9,pa$(n):
NEXT:FOR n=1 TO 4:PRINT#9,pe$(n):N
EXT                                  <3DE8>
2360 FOR n=1 TO 16:FOR m=1 TO 12:P
RINT#9,am$(n,m):NEXT:NEXT         <330A>
2370 FOR n=1 TO 4:FOR m=1 TO 12:PR
INT#9,em$(n,m):NEXT:NEXT         <32F5>
2380 CLOSEOUT:GOTO 200            <0BDA>
2390 'laden                        <0C5F>
2400 MODE 1:BORDER 6,0:PRINT,,," A
LLE DATEN WERDEN DADURCH GELOESCHT
!":LOCATE 16,12:PRINT"LADEN ?"    <4B86>
2410 S$=INKEY$:IF S$=""THEN 2410 E
LSE IF UPPER$(S$)="N"THEN BORDER q
:GOTO 190 ELSE BORDER q           <3A7B>
2420 MODE 1:PRINT"LADEN":PRINT u$ <1685>
2430 PEN 2:PRINT,,,"Aktuelle Daten
laden ----- ENTER":PRINT,
,,,"Spezielle Daten laden -----
--- TASTE"                         <6518>
2440 s$=INKEY$:IF s$=""THEN 2440 E
LSE IF s$("<") "THEN 2470         <2813>
2450 CLS:PEN 1:INPUT"Unter welchem
Namen laden";s$:IF LEN(s$)>8 THEN
2450                                 <3B1D>
2460 CLS:PRINT s$+".DAT wird gelad
en.Bitte warten !":OPENIN s$+".DAT
":GOTO 2480                         <4199>
2470 CLS:PRINT"Aktuelle Daten werd
en geladen. Bitte warten
!":OPENIN"pm-data.dat"            <4F4F>
2480 FOR n=1 TO 311:INPUT#9,a$(n):
NEXT                                <2175>
2490 FOR n=1 TO 16:INPUT#9,pa$(n):
NEXT:FOR n=1 TO 4:INPUT#9,pe$(n):N
EXT                                  <3DF4>
2500 FOR n=1 TO 16:FOR m=1 TO 12:I
NPUT#9,am$(n,m):NEXT:NEXT         <3369>
2510 FOR n=1 TO 4:FOR m=1 TO 12:IN
PUT#9,em$(n,m):NEXT:NEXT         <329D>
2520 CLOSEIN:MODE 1:PEN 1:PRINT"GE
BURTSTAGE":PRINT u$:PEN 2        <249E>
2530 FOR n=201 TO 300:IF MID$(a$(n
),3,1)<)"THEN 2560 ELSE z=VAL(MI
D$(a$(n),4,2))                    <482D>
2540 IF z=ma OR z=ma+1 THEN GOSUB
2570                                <229F>
2550 IF ma=12 AND z=1 THEN GOSUB 2
570                                  <1A6A>
2560 NEXT:PEN 1:PRINT,,,"Keine wei
teren Geburtstage.Druecke Taste":C
ALL &BB06:GOTO 200                 <4201>
2570 PRINT a$(n-200)":LEFT$(a$(n)
,10):RETURN                         <296F>
2580 '...farben                    <10EF>
2590 WINDOW#2,1,40,16,16:h$=CHR$(2
42)+ " COPY "+CHR$(243):m=0:n=0:PRI
NT#2,"Hintergrund "h$:GOSUB 2600:m
=1:n=26:PRINT#2,"Farbe 1"h$:GOSUB

```

```

2600:m=2:n=26:PRINT#2,"Farbe 2 "h$
:GOSUB 2600:PEN 2:GOTO 200      <AEAA>
2600 s$=INKEY$:IF s$=""THEN 2600 <19C0>
2610 IF s$=CHR$(242)THEN n=n-1:IF
n<0 THEN n=27                  <2D4B>
2620 IF s$=CHR$(243)THEN n=n+1:IF
n>26 THEN n=0                  <2D61>
2630 INK m,n:IF m=0 THEN BORDER n:
q=n                              <27BD>
2640 IF s$="c"THEN RETURN ELSE 260
0                                <15EE>
2650 '...tagesberechnung        <1981>
2660 sa=28:j=gj-1585:t=j*365:d=gj-
1584:b=d/4:st=INT(b):t=t+st:d=gj-
500:b=d/100:sh=INT(b):t=t-sh:d=gj-
1200:b=d/400:sv=INT(b):t=t+sv   <DB1F>
2670 FOR i=1 TO gm-1:t=t+m(i):NEXT
:t=t+gt:b=gj/4:IF INT(b)=b AND gm<
=2 THEN t=t-1:sa=29:sb=1       <7CA8>
2680 b=gj/100:IF INT(b)=b AND gm<
2 THEN t=t+1                    <351E>
2690 b=gj/400:IF INT(b)=b AND gm<
2 THEN t=t-1                    <368A>
2700 b=t/7:t=t-INT(b)*7:t=t+1:W0$=
W$(T):RETURN                    <4499>

```

```

2710 '...catalog                 <110C>
2720 MODE 2:WINDOW#2,1,80,1,1:WIND
OW#3,1,80,25,25:PRINT#2," USER:"us
er:PRINT#3,CHR$(24);" 1-CAT 2-US
ER AENDERN 3-HAUPTMENUE ";CHR$(2
4):WINDOW 1,80,3,23:CAT         <806E>
2730 A$=INKEY$:IF A$=""THEN 2730 E
LSE A=VAL(A$):IF A<1 OR A>3 THEN 2
730                               <3B7C>
2740 IF A=1 THEN CAT             <0E76>
2750 IF A=2 THEN INPUT#2," NEUE US
EREBENE EINGEBEN";USER:/USER,USER:
PRINT#2," USEREBENE NR."user:CAT <608E>
2760 IF A=3 THEN 200            <1016>
2770 GOTO 2730                  <0919>
2780 '...ende                   <0E44>
2790 MODE 0:BORDER q,6:PRINT,,," W
irklich beenden?"              <2987>
2800 s$=INKEY$:IF s$=""THEN 2800 E
LSE IF UPPER$(s$)="J"THEN/BASIC EL
SE BORDER q:GOTO 200           <3CDD>
2810 FOR n=1 TO 100:IF a$(n)="!"TH
EN a$(n+100)="":a$(n+200)="    <413D>
2820 NEXT                       <06D5>

```

Invertierungs- routine

Mit diesem Programm ist es auf einfache Weise möglich, den gesamten Bildschirm (oder Teile davon) zu invertieren, ohne daß der Bildschirminhalt verloren geht. Die Eingaben erfolgen nämlich über ein echtes Pull-Down-Menü. Darüber hinaus bietet das Programm die Möglichkeit, den eingegebenen Bereich je nach Mode auf bis zu 15 unterschiedliche Arten zu invertieren; damit lassen sich dann tolle Effekte erzielen.

Angenommen, Sie haben dieses Programm bereits in den Speicher geladen. Um ein bereits erstelltes Bild zu invertieren, ohne daß es durch Lademeldungen oder ähnliches zerstört wird, schalten Sie zunächst in den Mode um, unter dem das Bild abgespeichert wurde. Danach tippen Sie auf dem nun freien Bildschirm ein: Load"! <Bildname>":run

Auf dem Bildschirm erscheint jetzt in einem Window die Startmeldung der Invertierungsroutine, die je nach Mode unterschiedlich ist. Nach den erforderlichen Eingaben für die Grenzen des zu invertierenden Bereichs wird dann im Mode sofort, in den anderen Modi können Sie noch die Farbe angeben. Anschließend können Sie dann entweder wieder abspeichern oder weiter invertieren.

JE

```

1 '***** <2506>
2 '* INVERTIERUNGSRoutine * <25B9>
3 '* VON * <2538>
4 '* MARTIN SACHENBACHER * <253B>
5 '* FUER * <2592>
6 '* SCHNEIDER CPC-WELT * <2539>
7 '* CPC 464/664/6128 JE* <255F>
8 '***** <2514>
10 MEMORY &3E7F:DEFINT a-z:m=PEEK(
&B1C8):s=m*2*20+(m=0)*(-20)    <3A03>
20 FOR i=&3E80 TO &3E96:READ a:sum
=sum+a:POKE i,a:NEXT:IF sum<>1706
THEN PRINT"Fehler in DATA's":STOP <5A6F>
30 DATA 6,0,&e,0,&21,0,0,&cd,&4a,&
bc,&c9,&21,0,&c0,&11,0,&40,1,0,&40
,&ed,&b0,&c9                    <4F19>
40 CALL &3E8B:CALL &BC06,&40:IF m=
1 THEN b=16 ELSE b=8           <2A94>
50 GOSUB 310:CLS:PRINT CHR$(24);:I
F m>0 THEN MOVE 3*b,352:y=21*b-1:D
RAWR y,0                        <3FD7>
60 IF m>0 THEN PRINT"INVERTIERUNG
S-ROUTINE"CHR$(24)"<Martin Sachenba
cher>":PRINT"Grenzen des zu inver
tierenden Bereichs:"ELSE PRINT" Gr
enzen: "CHR$(24)                <83D0>
70 PRINT:PRINT l$" (1-"MID$(STR$(
s),2,2)");:INPUT":",l         <300C>
80 PRINT r$" ("MID$(STR$(1),2,2)"-
"MID$(STR$(s),2,2)");:INPUT":",r:

```

```

DRAWR-y,0,0 <481E>
90 PRINT:PRINT o$ " (1-25)";:INPUT
":",o <21C6>
100 PRINT u$ " ("MID$(STR$(o),2,2)"
-25)";:INPUT":",u:w=2^(2^(2-m))-1 <443B>
110 IF l<1 OR r>s OR l>r OR o<1 OR
u>25 OR o>u THEN PRINT CHR$(7);:C
LS:GOTO 50 <4811>
120 IF w=1 THEN a=1:GOTO 150 <18C4>
130 IF m=0 THEN CLS:INPUT" In welch
en Stiftsoll Farbstift umgewandel
t werden (1-15):",a:GOTO 150 <5730>
140 CLS:PRINT"Es gibt 3 moegliche
Invertierungsarten. Die Hintergr
undfarbe (Farbstift 0) kann indie
Farbstifte 1-3 umgewandelt werde
n.":PRINT:INPUT"Farbstift (1-3):",
a <A190>
150 IF a<1 OR a>w THEN PRINT CHR$(
7);:GOTO 130 ELSE a$=BIN$(a,2^(2-m
)) <3D69>
160 i$=RIGHT$(a$,1)+MID$(a$,2,1)+M
ID$(a$,3,1)+LEFT$(a$,1) <374C>
170 i$=RIGHT$(i$,2^(2-m)) <1D67>
180 FOR i=1 TO LEN(i$):z$=z$+STRIN
G$(2^m)*2,MID$(i$,i,1):NEXT <3F34>
190 inumask=VAL("&X"+z$) <1DCE>
200 POKE &3E81,inumask <1490>
210 CALL &BC06,&C0 <0DFF>
220 FOR x=1-1 TO r-1:FOR y=o-1 TO
u-1 <2C99>
230 POKE &3E85,y:POKE &3E86,x <1895>
240 CALL &3E80 <090F>
250 NEXT:NEXT:CALL &3E8B <0DCD>
260 CALL &BC06,&40:GOSUB 310:PRINT
" Invertieren ";:IF m=0 THEN PRINT <2D59>
270 PRINT"beendet":PRINT:PRINT" Bil
d abspeichern J/N" <29E2>
280 a$=LOWER$(INKEY$):IF a$="j"THE
N 290 ELSE IF a$="n"THEN 300 ELSE
280 <320C>
290 CLS:INPUT"Name:",n$:SAVE n$,b,
49152,16383 <2ACD>
300 WINDOW 1,s,1,25:CALL &BC06,&C0
:END <1CF0>
310 IF m=0 THEN WINDOW 2,19,3,11:C
LS:GOTO 330 ELSE WINDOW 3,25,3,12:
CLS <2C4E>
320 l$="Links":r$="rechts":PLOT 2*
b+(b/2),408-(16*3),1:DRAWR 22*b,0:
DRAWR 0,-144:DRAWR-22*b,0:DRAWR 0,
144:WINDOW 4,24,4,11:o$="oben":u$=
"unten":RETURN <8716>
330 l$=CHR$(242):r$=CHR$(243)+ " ":
PLOT 40,360,1:DRAWR 556,0:DRAWR 0,
-128:DRAWR-556,0:DRAWR 0,128:WINDO
W 3,18,4,10:o$=CHR$(240):u$=CHR$(2
41)+ " ":RETURN <6FC1>

```

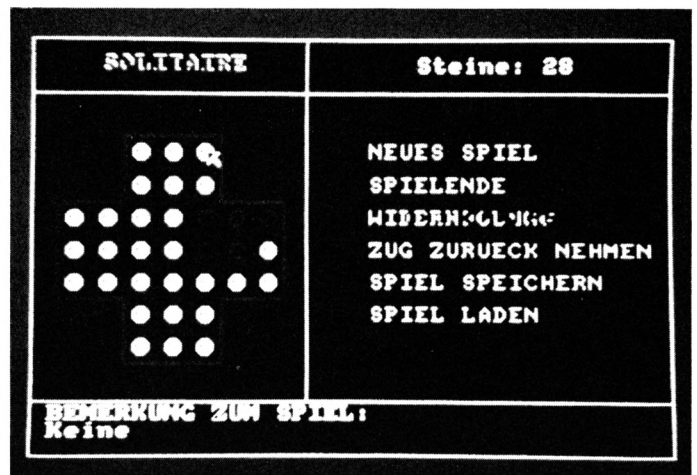
Solitaire

Das Brettspiel Solitaire ist ein uraltes Spiel, das schon viele Generationen immer wieder fasziniert hat. Unser Autor Ralph von Terwingen hat nun für den CPC umgesetzt.

Die Spielregeln sind denkbar einfach: Auf einem Brett befinden sich 45 Löcher mit mit 44 Steinen, ein Loch ist also nicht belegt. Das Ziel ist, möglichst viele Steine (vielleicht auch alle, wenn Sie es schaffen) vom Spielfeld zu nehmen. Sie können immer dann einen Stein entfernen, wenn dieser von einem anderen übersprungen wird. Und zwar vertikal oder horizontal. Diagonale Sprünge sind verboten, ebenso natürlich Sprünge außerhalb des Spielfeldes.

Das hört sich ganz einfach an. Aber Sie müssen darauf achten, keinen Stein isoliert stehen zu lassen. Und da liegt die Schwierigkeit. Um die Aufgabe zu lösen, brauchen Sie gute taktische Übersicht. Probieren Sie es!

Das Spiel läuft auf den CPC-Typen 664 und 6128. Da die Züge auch zurückgenommen werden können, haben Sie die Möglichkeit der Korrektur. Außerdem können Sie jeden beliebigen Spielstand abspeichern und später wieder einladen. Gesteuert wird mit den Cursor-Tasten. Durch Drücken der Copy-Taste wird der Stein markiert, mit dem Sie springen wollen. JE



```

10 '***** <24C3>
20 '* SOLITAIR * <241D>
30 '* VON * <247F>
40 '* RALPH VON TERWINGEN * <24E8>
50 '* FUER * <24E2>
60 '* SCHNEIDER CPC-WELT * <245A>
70 '* CPC 664/6128 JE* <245E>
80 '***** <244E>
130 ' <070E>
140 DIM lochs1(40):DIM lochz1(40):
DIM lochs2(40):DIM lochz2(40):DIM
steins(40):DIM steinz(40) <5E68>

```

```

150 zug=0:p=1:r=0 <1BF9> R$(231):NEXT:NEXT <3838>
160 MODE 1 <07E8> 500 PEN 3:LOCATE 9,13:PRINT CHR$(2
170 ' <075E> 30) <156B>
180 ' ***** ZEICHEN ***** <1DD1> 510 LOCATE 6,6:PRINT CHR$(150);STR
190 ' <0786> ING$(5,154);CHR$(156) <2171>
200 SYMBOL AFTER 0 <08C0> 520 FOR z=1 TO 3:LOCATE 6,6+z:PRIN
210 SYMBOL &9F,&0,&7E,&7C,&78,&7C, <2992> T CHR$(149):LOCATE 12,6+z:PRINT CH
&6E,&47,&3 <29FA> R$(149):NEXT <3523>
220 SYMBOL &E7,&3C,&6E,&DF,&9F,&BF <2930> 530 LOCATE 2,10:PRINT CHR$(150);ST
,&FF,&7E,&3C <07EA> RING$(3,154);CHR$(153):LOCATE 12,1
230 SYMBOL &E6,&3C,&42,&81,&81,&81 <1F8F> 0:PRINT CHR$(147);STRING$(3,154);C
,&81,&42,&3C <2930> HR$(156) <417A>
240 ' <07EA> 540 FOR z=1 TO 5:LOCATE 2,10+z:PRI
250 ' ***** TITELBILD ***** <1F8F> NT CHR$(149):LOCATE 16,10+z:PRINT
260 ' <0714> CHR$(149):NEXT <37F6>
270 GRAPHICS PEN 1 <081B> 550 LOCATE 2,16:PRINT CHR$(147);ST
280 LOCATE 16,11:PRINT"SOLITAIRE" <18E7> RING$(3,154);CHR$(156)::LOCATE 12,
290 LOCATE 13,13:PRINT"SPIELREGELN <1F18> 16:PRINT CHR$(150);STRING$(3,154);
(J/N)" <1239> CHR$(153) <429F>
300 PRINT CHR$(23);CHR$(3) <28CD> 560 FOR z=1 TO 3:LOCATE 6,16+z:PRI
310 TAG:MOVE 24*8,26*8+1:PRINT"SPI <2377> NT CHR$(149):LOCATE 12,16+z:PRINT
ELREGELN(J/N)"; <06DC> CHR$(149):NEXT <3767>
320 GRAPHICS PEN 3:MOVE 30*8,30*8+ <2377> 570 LOCATE 6,20:PRINT CHR$(147);ST
1:PRINT"SOLITAIRE"; <06DC> RING$(5,154);CHR$(153) <226D>
330 TAGOFF <06DC> 580 PLOT 1,1:DRAW 639,1:DRAW 639,3
340 GRAPHICS PEN 1:PRINT CHR$(23); <1671> 99:DRAW 1,399:DRAW 1,1 <25CD>
CHR$(0) <1671> 590 PLOT 270,50:DRAW 270,400:PLOT
350 WHILE ab$="" :ab$=INKEY$:r=r+6: <E83A> 1,50:DRAW 640,50 <2331>
PLOT 1+r,1+r:DRAW 1+r,400-r:PLOT 1 <14D0> 600 PLOT 1,350:DRAW 640,350 <14D0>
+r,400-r:DRAW 640-r,400-r:PLOT 640 <12A6> 610 PRINT CHR$(23);CHR$(3) <12A6>
-r,400-r:DRAW 640-r,1+r:PLOT 640-r <3DAE> 620 LOCATE 5,2:PEN 3:PRINT"SOLITAI
,1+r:DRAW 1+r,1+r:IF r>140 THEN r= <0634> 630 TAGOFF <0634>
0:IF p=0 THEN p=1:GRAPHICS PEN p E <19C4> 640 LOCATE 2,23:PEN 1:PRINT"BEMERK
LSE IF p=1 THEN p=0:GRAPHICS PEN p <16BA> 650 WINDOW#1,2,39,24,24:PRINT#1,be
<07C9> 660 PEN 1:LOCATE 25,2:TAGOFF:PRINT <214B>
360 LOCATE 13,15:PRINT CHR$(INT(RN <073F> "Steine: ";steine <2441>
D*4)+248):LOCATE 28,15:PRINT CHR$( <21F7> 670 LOCATE 22,7:PRINT"NEUES SPIEL" <190B>
INT(RND*4)+248) <0767> <190B>
370 WEND <0767> 680 LOCATE 22,9:PRINT"SPIELEENDE" <177D>
380 IF ab$="n" THEN 390 ELSE GOSUB <38CF> 690 LOCATE 22,11:PRINT"WIEDERHOLUN
1400 <0DE9> G" <1BF6>
390 PRINT CHR$(23);CHR$(0):GRAPHIC <2007> 700 LOCATE 22,13:PRINT"ZUG ZURUECK
S PEN 1 <07C8> NEHMEN" <214D>
400 MODE 1 <07C8> 710 LOCATE 22,15:PRINT"SPIEL SPEIC
410 ' <073F> HERN" <1EDD>
420 ' *ZEICHEN DES SPIELFELDES* <21F7> 720 LOCATE 22,17:PRINT"SPIEL LADEN
430 ' <0767> " <1AC3>
440 PRINT CHR$(23);CHR$(0):steine= <39F3> 730 PRINT CHR$(23);CHR$(2) <128F>
32:bemerk$=CHR$(11)+"Keine" <39F3> 740 TAG:MOVE 21*16-2,38*8-2:PRINT"
450 zug=0 <0DE9> NEUES SPIEL"; <263C>
460 FOR farbe=0 TO 3:INK farbe,0:N <2296> 750 MOVE 21*16-2,34*8-2:PRINT"SPIE
EXT <2007> LLENDE"; <2296>
470 PEN 1 <07C8> 760 MOVE 21*16-2,30*8-2:PRINT"WIED
480 FOR z=1 TO 6 STEP 2:FOR s=1 TO
14 STEP 2:LOCATE 2+s,10+z:PRINT C
HR$(231):NEXT:NEXT <39F3>
490 FOR s=1 TO 6 STEP 2:FOR z=1 TO
14 STEP 2:LOCATE 6+s,6+z:PRINT CH

```

LISTINGS

```

ERHOLUNG" <2591> <245A>
770 MOVE 21*16-2,26*8-2:PRINT"ZUG <1085>
ZURUECK NEHMEN" <2A0F>
780 MOVE 21*16-2,22*8-2:PRINT"SPIE <1A4A>
L SPEICHERN" <2775>
790 MOVE 21*16-2,18*8-2:PRINT"SPIE <1462>
L LADEN" <2376>
800 TAGOFF <068A>
810 PRINT CHR$(23);CHR$(0):PLOT 63 <2C0A>
9,1:DRAW 639,399 <226C>
820 PRINT CHR$(23);CHR$(1) <1706>
830 INK 1,26:INK 2,15:INK 3,6 <099E>
840 ' <0796>
850 ' ***** SPIELANFANG ***** <1DD7>
860 ' <07BE>
870 z=13*16-8:s=8*16+8 <2988>
880 TAG:MOVE s,z:PRINT CHR$(159); <20E2>
TAGOFF <1C7A>
890 i$=INKEY$ <1A28>
900 IF loesch=1 THEN TAG:MOVE s,z: <201A>
PRINT CHR$(159);:loesch=0 <19F4>
910 IF i$=""THEN 890 <33E9>
920 TAG:MOVE s,z:PRINT CHR$(159);: <11F3>
TAGOFF <1C0E>
930 IF i$=CHR$(240)THEN z=z+32 <1EE6>
940 IF i$=CHR$(241)THEN z=z-32 <1E06>
950 IF i$=CHR$(242)THEN s=s-32 <1EFD>
960 IF i$=CHR$(243)THEN s=s+32 <1E00>
970 IF z>328 OR z=328 THEN z=z-32 <249A>
980 IF z<72 OR z=72 THEN z=z+32 <22D6>
990 IF s<8 OR s=8 THEN s=s+32 <205C>
1000 IF s>648 OR s=648 THEN s=s-32 <242E>
1010 TAG:MOVE s,z:PRINT CHR$(159); <1CC2>
:TAGOFF <2909>
1020 IF i$=CHR$(224)THEN z=z+8:s=s <4103>
-8 <09F1>
1030 IF i$=CHR$(224)AND s/16+1>21 <3F56>
AND s/16+1<40 THEN 1700 ELSE IF i$ <0C6C>
=CHR$(224)THEN 1050 <118F>
1040 GOTO 890 <2FD8>
1050 TAGOFF:sa=s:za=z:spo=s/16+1:z <2E08>
po=26-z/16 <2FC9>
1060 i$=INKEY$ <3F56>
1070 IF i$=""THEN 1060 <0C6C>
1080 IF i$=CHR$(240)THEN z=z+64:z3 <118F>
=-2:s3=0 <2FD8>
1090 IF i$=CHR$(241)THEN z=z-64:z3 <2E08>
=2:s3=0 <2FC9>
1100 IF i$=CHR$(242)THEN s=s-64:s3 <3F56>
=-2:z3=0 <0C6C>
1110 IF i$=CHR$(243)THEN s=s+64:s <118F>
3=2:z3=0 <2FD8>
1120 IF z>328 OR z=328 THEN z=z-32 <2E08>
<24C6>
1130 IF z<72 OR z=72 THEN z=z+32 <2202>
1140 IF s<8 OR s=8 THEN s=s+32 <2088>
1150 IF s>648 OR s=648 THEN s=s-32
1160 i$=CHR$(224) <245A>
1170 TAG:MOVE s,z:PRINT CHR$(159); <1085>
1180 FOR zeit=1 TO 100:NEXT <1A4A>
1190 MOVE s,z:PRINT CHR$(159);:MOV <1462>
E s,z:PRINT CHR$(159); <2C0A>
1200 TAG:IF i$=CHR$(224)THEN 1250 <1706>
1210 GOTO 1060 <099E>
1220 ' <0796>
1230 ' *** ZUG AUSWERTEN *** <1DD7>
1240 ' <07BE>
1250 spo2=s/16+1:zpo2=26-z/16 <2988>
1260 za=za-8:sa=sa+8 <20E2>
1270 TAG:MOVE sa,za:PRINT CHR$(159 <1C7A>
); <1A28>
1280 TAG:MOVE s,z:PRINT CHR$(159); <201A>
1290 za=za+8:sa=sa-8 <19F4>
1300 IF z4<>0 AND s4<>0 THEN 1350 <33E9>
1310 TAGOFF:LOCATE spo,zpo:zeich$= <11F3>
COPYCHR$(#0):LOCATE s/16+1,26-z/16 <1C0E>
:zeich2$=COPYCHR$(#0):LOCATE spo+s <1EE6>
3,zpo+z3:zeich3$=COPYCHR$(#0) <1E06>
1320 IF zeich$=CHR$(231)AND zeich2 <1EFD>
$=CHR$(230)AND zeich3$=CHR$(231)TH <1E00>
EN 1330 ELSE 1350 <249A>
1330 zug=zug+1:lochs1(zug)=spo:loc <22D6>
hz1(zug)=zpo:lochs2(zug)=spo+s3:lo <205C>
chz2(zug)=zpo+z3:steins(zug)=spo2: <242E>
steinz(zug)=zpo2 <1CC2>
1340 LOCATE spo,zpo:PEN 3:PRINT CH <2909>
R$(230):LOCATE spo2,zpo2:PEN 1:PRI <4103>
NT CHR$(231):LOCATE spo+s3,zpo+z3: <09F1>
PEN 3:PRINT CHR$(230):steine=stein <3F56>
e-1:GOTO 1360 <0C6C>
1350 PEN 1:TAGOFF:PRINT"":PRINT#1, <118F>
"Falsche Eingabe" <2FD8>
1360 loesch=1 <2E08>
1370 IF steine=1 THEN 1830 <2FC9>
1380 LOCATE 25,2:PEN 1:TAGOFF:PRIN <3F56>
T"Steine:";steine:z3=0:s3=0:z=z-8: <0C6C>
s=s+8:GOTO 890 <118F>
1390 ' <2FD8>
1400 ' *** SPIELREGELN *** <2E08>
1410 ' <2FC9>
1420 MODE 2 <3F56>
1430 LOCATE 35,1:PRINT"SPIELREGELN <0C6C>
:" <118F>
1440 TAGOFF:PRINT CHR$(23);CHR$(3) <2FD8>
<2E08> <2FC9>
1450 GRAPHICS PEN 1:PEN 1:TAG:MOVE <3F56>
34*8,25*16-4:PRINT"SPIELREGELN";: <0C6C>
TAGOFF <118F>
1460 LOCATE 35,2:PRINT"----- <2FD8>
--" <2E08>
1470 PRINT:PRINT <2FC9>
1480 PRINT"Man beginnt mit den Ste

```

```

inen zu springen.Uebersprungen kan
n nur jeweils ein Stein" <5876>
1490 LOCATE 1,6:PRINT"werden,und a
uch nur dann wenn hinter ihm ein L
och frei ist." <480E>
1500 PRINT <06BB>
1510 PRINT"Man kann nur,waagerecht
oder senkrecht springen,aber nich
t diagonal." <4CA0>
1520 PRINT <06E3>
1530 PRINT"Jeder uebersprungene St
ein wird aus dem Spiel genommen." <3F7C>
1540 PRINT <060D>
1550 PRINT"Die auf dem Spielbrett
ueberigebliebenen Steine - das sin
d alle die" <4BED>
1560 PRINT"man nicht mehr ueberspr
ingen kann - zaehlen als Verlustpu
nkte. <45E1>
1570 PRINT <0649>
1580 PRINT"Fuer Profis:" <148E>
1590 PRINT"So geschickt springen,d
as <20B1>
1600 PRINT"- nur ein Stein ueberig
bleibt, <25DF>
1610 PRINT"- dieser Stein am Schlu
ss genau im Mittelloch sitzt. <3BF9>
1620 PRINT <06AD>
1630 PRINT"Die Steuerung erfolgt u
eber den Cursor- und Copytaste." <3EFF>
1640 LOCATE 33,25:PRINT" TASTE DRU
ECKEN <1D35>
1650 zei2=-2:ab$="" <1807>
1660 GRAPHICS PEN 1:LOCATE 1,1:PRI
NT CHR$(23);CHR$(1) <1B88>
1670 PLOT-2,1:DRAW 640,1 <11A2>
1680 WHILE ab$="" :zei=zei+2:zei2=z
ei2+1:ab$=INKEY$:PLOT-2,zei:DRAW 6
40,zei:PLOT zei2,-2:DRAW zei2,400:
PLOT 640-zei2,-2:DRAW 640-zei2,400
:PLOT-2,400-zei:DRAW 640,400-zei:I
F zei2=200 THEN zei2=-2:zei=-2 <CAC0>
1690 WEND:RETURN <08C3>
1700 ' <0758>
1710 ' ***** MENUE ***** <1B28>
1720 ' <0780>
1730 z=z-8:s=s+8 <1C9A>
1740 TAG:MOVE s,z:PRINT CHR$(159);
:TAGOFF:PRINT CHR$(23);CHR$(0) <2A1F>
1750 z=z+8:s=s-8 <1C4C>
1760 IF 26-z/16=7 THEN 440 <1603>
1770 IF 26-z/16=9 THEN 1830 <1622>
1780 IF 26-z/16=11 THEN 2000 <1702>
1790 IF 26-z/16=13 THEN 2190 <1791>
1800 IF 26-z/16=15 THEN 2300 <1792>
1810 IF 26-z/16=17 THEN 2430 <173B>
1820 ' <074A>
1830 ' ***BEMERKUNG AUSGEBEN*** <20DE>
1840 ' <0772>
1850 PEN 1 <0795>
1860 RESTORE 1970 <0914>
1870 READ steine2,bemerk$:IF stein
e2=steine THEN 1880 ELSE IF steine
2=10 THEN 1890 ELSE 1870 <4DB3>
1880 PRINT#1,"Sie spielen ";bemerk
$ <211D>
1890 IF steine>9 THEN PRINT#1,"Wue
rg..." <20A2>
1900 :TAGOFF:PRINT CHR$(23);CHR$(0
) <1533>
1910 LOCATE 21,21:PRINT"Neues Spie
l (J/N)":ab$="" <298C>
1920 GRAPHICS PEN 1:PLOT 271,90:DR
AW 639,90 <180A>
1930 WHILE ab$="" :ab$=INKEY$:WEND <19B3>
1940 LOCATE 21,21:PRINT"
" <20A3>
1950 GRAPHICS PEN 0:PLOT 271,90:DR
AW 639,90:GRAPHICS PEN 1 <1CC7>
1960 IF ab$="n"THEN MODE 2:END ELS
E 440 <1989>
1970 DATA 1,wie ein Champion,2,seh
r gut,3,gut,4,ziemlich gut <3933>
1980 DATA 5,mittelmassig,6,wie ei
n Antaenger,7,schlecht,8,sehr schl
echt <4555>
1990 DATA 9,miserabel,10,* <1722>
2000 ' <07B1>
2010 ' ***WIEDERHOLUNG*** <1AE2>
2020 ' <07D9>
2030 PEN 1 <07FE>
2040 FOR z=1 TO 6 STEP 2:FOR s=1 T
O 14 STEP 2:LOCATE 2+s,10+z:PRINT
CHR$(231):NEXT:NEXT <392A>
2050 FOR s=1 TO 6 STEP 2:FOR z=1 T
O 14 STEP 2:LOCATE 6+s,6+z:PRINT C
HR$(231):NEXT:NEXT <386F>
2060 PEN 3:LOCATE 9,13:PRINT CHR$(
230) <15A2>
2070 ab$="" :steine=32 <1A36>
2080 PEN 1:LOCATE 25,2:PRINT"Stein
e:";steine <2230>
2090 FOR an=1 TO zug <1438>
2100 WHILE ab$="" :ab$=INKEY$:WEND:
IF ab$=CHR$(224)THEN zug=an-1:GOTO
820 <3BE9>
2110 LOCATE steins(an),steinz(an):
PEN 1:PRINT CHR$(231) <32B3>
2120 LOCATE lochs1(an),lochz1(an):
PEN 3:PRINT CHR$(230) <328F>
2130 LOCATE lochs2(an),lochz2(an):
PEN 3:PRINT CHR$(230) <32C7>
2140 ab$="" :steine=steine-1 <236B>
2150 PEN 1:LOCATE 25,2:PRINT"Stein
e:";steine <22BC>
2160 NEXT an <0B47>

```

```

2170 GOTO 820 <099E>
2180 ' <071A>
2190 ' ***ZUG ZURUECK NEHMEN*** <2081>
2200 ' <0742>
2210 steine=steine+1 <1AA8>
2220 IF steine>32 THEN steine=stei
ne-1:zug=0:GOTO 820 <361C>
2230 PEN 1:LOCATE 25,2:PRINT"Stein
e:";steine <225D>
2240 LOCATE steins(zug),steinz(zug
):PEN 3:PRINT CHR$(230) <3474>
2250 LOCATE lochs1(zug),lochz1(zug
):PEN 1:PRINT CHR$(231) <3446>
2260 LOCATE lochs2(zug),lochz2(zug
):PEN 1:PRINT CHR$(231) <345E>
2270 zug=zug-1 <14D1>
2280 GOTO 820 <097B>
2290 ' <07F6>
2300 ' *** SPIEL SPEICHERN *** <1F06>
2310 ' <0720>
2320 OPENOUT"solitair.dat" <145E>
2330 PRINT#9,zug <0F87>
2340 PRINT#9,steine <1237>
2350 FOR an=1 TO zug <1441>
2360 PRINT#9,steins(an):PRINT#9,st
einz(an) <2E73>
2370 PRINT#9,lochs1(an):PRINT#9,lo
chz1(an) <2E7C>
2380 PRINT#9,lochs2(an):PRINT#9,lo
chz2(an) <2EA8>
2390 NEXT an <0B14>
2400 CLOSEOUT <06EF>
2410 GOTO 820 <097F>
2420 ' <07FC>
2430 ' *** SPIEL LADEN *** <1B08>
2440 ' <0723>
2450 PEN 1 <0748>
2460 FOR z=1 TO 6 STEP 2:FOR s=1 T
O 14 STEP 2:LOCATE 2+s,10+z:PRINT
CHR$(231):NEXT:NEXT <3973>
2470 FOR s=1 TO 6 STEP 2:FOR z=1 T
O 14 STEP 2:LOCATE 6+s,6+z:PRINT C
HR$(231):NEXT:NEXT <38B7>
2480 PEN 3:LOCATE 9,13:PRINT CHR$(
230) <15EC>
2490 OPENIN"solitair.dat" <14B4>
2500 INPUT#9,zug <0FFA>
2510 INPUT#9,steine <1279>
2520 FOR an=1 TO zug <1496>
2530 INPUT#9,steins(an):INPUT#9,st
einz(an) <2E3E>
2540 INPUT#9,lochs1(an):INPUT#9,lo
chz1(an) <2EC8>
2550 INPUT#9,lochs2(an):INPUT#9,lo
chz2(an) <2EDF>
2560 LOCATE steins(an),steinz(an):
PEN 1:PRINT CHR$(231) <3239>
2570 LOCATE lochs1(an),lochz1(an):

```

```

PEN 3:PRINT CHR$(230) <3215>
2580 LOCATE lochs2(an),lochz2(an):
PEN 3:PRINT CHR$(230) <324D>
2590 NEXT an <0BA4>
2600 CLOSEIN <067C>
2610 PEN 1:LOCATE 25,2:PRINT"Stein
e:";steine <2256>
2620 GOTO 820 <0924>

```

Hyperinversi

Die Spielregeln sind einfach! Eine vorgegebene Zahlenfolge muß in die richtige Reihenfolge gebracht werden. Es werden zum Beispiel fünf Zahlen nebeneinander vorgeblendet, ein weiteres sechstes Feld ist leer. Jetzt können Sie mit den Cursortasten ein Feld anklicken und mit Copy bestätigen. Das angeklickte Feld wird dann mit dem freien vertauscht.

Statt mit Zahlen können Sie auch mit Symbolen oder Buchstaben die Tauscherei vornehmen. Viel Spaß beim Tippen und Spielen. JE

```

10 '***** <24C3>
20 '* HYPERINVERSI * <24B6>
30 '* VON * <2407>
40 '* AURELIANO CRAMERI * <24C9>
50 '* FUER * <247E>
60 '* SCHNEIDER CPC-WELT * <245A>
70 '* CPC 464/664/6128 JE* <24D1>
80 '***** <244E>
90 ' <07BF>
100 'Initialisierung <16A4>
110 ' <07E7>
120 ENV 1,3,5,1,15,-1,5 <1501>
130 ENT 1,5,1,15 <0EB3>
140 KEY DEF 1,0:KEY DEF 8,0 <1086>
150 MODE 0:CALL &BB4E:CALL &BC02:B
ORDER 10 <1524>
160 ' <074A>
170 'Titel,Menue <1223>
180 ' <0772>
190 LOCATE 1,4:PRINT STRING$(20,21
6):LOCATE 5,5:PRINT"IPERINVERSE":L
OCATE 1,6:PRINT STRING$(20,218) <3881>
200 LOCATE 2,10:PRINT"Soll ich die
Ziel-"," position bestimmen?" <37A7>
210 a$=INKEY$:IF a$=""THEN 210 ELS
E IF a$="j"THEN PO=1 <2C8B>
220 PRINT" 1 Zahlen"," 2 Buchstabe
n"," 3 Symbolen" <2F8F>
230 sp=VAL(INKEY$):IF sp<1 OR sp>3
THEN 230 <2661>
240 PRINT CHR$(10)" Level (3-9) ?"
<1CF7>
250 nu=VAL(INKEY$):IF nu<3 OR nu>1
0 THEN 250 <2751>

```

```

260 IF nu MOD 2=0 THEN xa=58:ya=11
ELSE xa=64:ya=10 <3474>
270 d=(ya-nu-1)/2:d2=nu+1 <2945>
280 ' <073C>
290 'Bildschirmaufbau <1811>
300 ' <0764>
310 LOCATE 2,10:PRINT CHR$(20) <1245>
320 FOR i=184 TO 216 STEP 2:MOVE d
**xa,i:DRAWR d2**xa,0,6:NEXT <3648>
330 MOVE d**xa,216:DRAWR d2**xa,0,1:
MOVE d**xa,184:DRAWR d2**xa,0 <4253>
340 FOR i=d TO d+d2:MOVE i**xa,184:
DRAWR 0,32:NEXT <3179>
350 FOR i=0 TO 38 STEP 2:MOVE i+d*
xa,i+218:DRAWR d2**xa-2*i,0,(i MOD
12)+2:NEXT <4EA3>
360 FOR i=d TO d+nu:MOVE i**xa+4,18
6:DRAWR xa-8,0,12:DRAWR 0,28,10:MO
VE XPOS-4,YPOS:DRAWR-xa+12,0,12:DR
AWR 0,-28,10:NEXT <67AE>
370 FOR i=d TO d+nu:MOVE i**xa+8,18
8:DRAWR xa-16,0,12:DRAWR 0,24,10:M
OVE XPOS-4,YPOS:DRAWR-xa+20,0,12:D
RAWR 0,-24,10:NEXT <689D>
380 ' <0703>
390 'Zeichenwahl <1285>
400 ' <072B>
410 IF re=1 THEN re=0:GOTO 560 <1AC0>
420 ON sp GOSUB 790,800,810 <1704>
430 RANDOMIZE TIME <0899>
440 b%=RND*nu+1 <14A2>
450 FOR i=1 TO nu+1 <14C9>
460 IF i=b% THEN c$=c$+" ":GOTO 500
<22F9>
470 a%=RND*nu+1:IF a%>nu THEN 470 <24FA>
480 FOR i2=1 TO i:IF CHR$(a%+li)=M
ID$(c$,i2,1) THEN 470 ELSE NEXT i2 <3EB8>
490 c$=c$+CHR$(a%+li) <1DE1>
500 NEXT i:IF po=1 THEN po=2:st$=c
$:c$="" :b2=b%:GOTO 430 ELSE IF po=
2 THEN en$=c$:po=1 <5B95>
510 IF po=0 THEN st$=c$:en$="" :b2
=b%:FOR i=1 TO nu:en$=en$+CHR$(i+l
i):NEXT <58C5>
520 st1$=st$:en1$=en$:b21=b2 <2B6A>
530 ' <0731>
540 'Anzeige <0E7E>
550 ' <0759>
560 LOCATE(13-nu)/2,17:PEN 11:PRIN
T"START: ";:PAPER 3:PEN 1:PRINT st
$:LOCATE(13-nu)/2,19:PAPER 0:PEN 1
1:PRINT"ZIEL: ";:PAPER 3:PEN 1:PR
INT en$:LOCATE(13-nu)/2,21:PAPER 0
:PEN 11:PRINT"ZUEGE: 0" <87BD>
570 PRINT CHR$(23);CHR$(1):PAPER 6
:TAG:PLOT-4,-4,2:FOR i=0 TO nu+1:M
OVE(i+d)**xa+16,206:PRINT MID$(st$,
i+1,1);:NEXT:PAPER 0 <626B>
580 ' <0795>
590 'Abfrage <0E0D>
600 ' <07BD>
610 f1=MAX(0,b2-3):f2=MIN(nu,b2+1)
:MOVE(b2-1+d)**xa+16,180:PRINT CHR$
(244); <5571>
620 f3=b2-1:f4=b2-1 <20B6>
630 IF INKEY(9)=0 AND f3<>b2-1 THE
N 710 <1FFD>
640 IF INKEY(1)=0 THEN f3=MIN(f3+1
,f2) <2583>
650 IF INKEY(8)=0 THEN f3=MAX(f3-1
,f1) <2570>
660 IF f3=f4 THEN 630 <15C1>
670 ' <0748>
680 'Zeichenaustauschung <194B>
690 ' <0770>
700 MOVE(f4+d)**xa+16,180:PRINT CHR
$(244);:MOVE(f3+d)**xa+16,180:PRINT
CHR$(244);:f4=f3:CALL &BB06:GOTO
630 <6086>
710 PLOT-4,-4,2:MOVE(f3+d)**xa+16,2
06:PRINT MID$(st$,f3+1,1); <3C65>
720 MOVE(b2-1+d)**xa+16,206:PRINT M
ID$(st$,f3+1,1); <35A3>
730 MID$(st$,b2,1)=MID$(st$,f3+1,1
):MID$(st$,f3+1,1)=" ":b2=f3+1 <4D55>
740 mo=mo+1:TAGOFF:LOCATE(13-nu)/2
,21:PRINT"ZUEGE:";:PEN 1:PRINT mo:
SOUND 2,50,0,0,1,1:IF st$=en$ THEN
GOSUB 850 ELSE PEN 11:TAG <61FE>
750 CALL &BB06:MOVE(f3+d)**xa+16,18
0:PRINT CHR$(244);:GOTO 610 <3199>
760 ' <07FC>
770 'Bereich bestimmen <188A>
780 ' <0726>
790 li=48:RETURN <0F85>
800 li=64:RETURN <0F9E>
810 li=224:RETURN <0FA8>
820 ' <0776>
830 'Ende <0B64>
840 ' <079E>
850 DATA 190,201,190,201,190,253,2
13,239,284,284,478,379,284,253 <3E7E>
860 DATA 379,301,253,239,239,379 <1EFA>
870 RESTORE:FOR i=1 TO 20:READ i2:
SOUND 1,i2/2:NEXT <2560>
880 LOCATE 1,7:PRINT CHR$(20):PRIN
T CHR$(23);CHR$(0) <1F79>
890 LOCATE 2,10:PRINT"1 neues Spie
l"," 2 neues Level"," 3 gleiche Si
tuation" <4337>
900 a=VAL(INKEY$):IF a=1 THEN RUN <1A26>
910 IF a=2 THEN PRINT" Level (3-9)
?":c$="" :en$="" :GOTO 250 <34B1>
920 IF a=3 THEN st$=st1$:en$=en1$:
b2=b21:re=1:mo=0:GOTO 310 <48BC>
930 GOTO 900 <0966>

```

MUSIK-DEMO

Für die Freunde von Musikdemos hat unser Autor Manfred Lipowski diesmal drei klassische Stücke für den CPC umgesetzt.

Alle drei Musikstücke sind als Demos gedacht und können nach Belieben in eigene Programme eingearbeitet werden. Den vollen Klanggenuß erhalten Sie jedoch erst dann, wenn Sie sie über eine Stereoanlage abspielen. Uns stand für den Test eine 100-Watt-Anlage zur Verfügung!

Musikdemo 1: Andante

Musikdemo 2: Capriccio

Musikdemo 3: Spanische Romanze

Da die Listings in reinem Basic geschrieben sind, können sie je nach Geschmack angepaßt werden. Sie laufen auf allen drei CPC-Typen.

JE

```

10 '***** <2397>
12 '*      MUSIKDEMO 1      * <231B>
14 '*      ANDANTE        * <2321>
16 '*      VON            * <2359>
18 '*      MANFRED LIPPOWSKI * <2357>
20 '*      FUER           * <2309>
22 '*      SCHNEIDER CPC-WELT * <23F1>
24 '*      CPC 464/664/6128  JE* <2380>
26 '***** <23B7>
40 INK 1,26:INK 2,24:INK 3,0:CLS:M
ODE 1:BORDER 0:INK 0,0:PAPER 0:PEN
 3:LOCATE 1,25:PRINT" CPC Soundmac
hine":PEN 1 <46B0>
50 PEN 2:LOCATE 3,3:PRINT CHR$(164
)+ " Manfred Lipowski":LOCATE 7,5:P
RINT"Juli 1987":LOCATE 9,7:PRINT"P
art 1" <49D3>
60 FOR a=0 TO 290:FOR b=2 TO 24 ST
EP 2:ORIGIN 0,0:IF TEST(a,b)=0 THE
N 80 <373A>
70 ORIGIN 25,25:c=a*2:PLOT a*2,b*2
+c/2.2+2:PLOT a*2,b*2+c/2.2,1 <51B7>
80 NEXT:NEXT:ENV 1,15,-1,11:ENV 2,
15,-1,6:ENV 3,15,-1,22:ENV 4,15,-1
,26:ENV 5,15,-1,32:ENT-2,2,-4,3,2,
4,3:ENT-6,2,-2,6,2,2,6 <65FD>
90 LOCATE 1,25:PRINT SPACE$(22):IN
K 3,2:PEN 3:LOCATE 26,16:PRINT"* A
ndante *":LOCATE 30,19:PRINT"von":
LOCATE 24,21:PRINT"Anton Diabelli"
:LOCATE 26,23:PRINT"(1781-1858)" <6D23>
100 DATA 379,0,239,0,319,0,426,0,2
53,0,319,0,379,0,239,0,319,0,506,0
,213,0,319,0,478,0,190,0,319,0,0,3
79,0,0,319,0,478,0,239,0,319,0,0,3
79,0,0,319,0 <9284>

```

```

110 DATA 506,0,213,0,319,0,536,0,2
25,0,319,0,506,0,213,0,319,0,478,0
,190,0,319,0,426,0,179,0,319,0,0,2
53,0,0,319,0,506,0,213,0,319,0,0,4
26,0,0,319,0,379,0,159,0,319,0,0,2
39,0,0,319,0,426,0,179,0,319,0,0,2
53,0,0,319,0 <D696>
120 DATA 478,0,190,0,319,0,0,239,0
,0,319,0,358,0,213,0,190,0,0,179,0
,0,213,0,638,0,239,0,319,0,0,379,0
,0,319,0,638,0,253,0,319,0,0,426,0
,0,319,0,478,0,239,0,379,0,0,319,0
,0,379,0,0,478,0 <B883>
130 DATA 506,0,213,0,319,0,0,319,0
,0,319,0,536,0,225,0,319,0,506,0,2
13,0,319,0,478,0,190,0,319,0,0,319
,0,0,319,0,379,0,239,0,319,0,0,319
,0,0,319,0,478,0,190,0,319,0,0,319
,0,0,319,0,506,0,213,0,319,0,478,0
,190,0,319,0 <D6B8>
140 DATA 426,0,179,0,319,0,0,319,0
,0,319,0,506,0,213,0,319,0,0,319,0
,0,319,0,379,0,159,0,319,0,0,239,0
,0,319,0,568,0,159,0,284,0,0,225,0
,0,284,0,426,0,169,0,284,0,0,213,0
,0,284,0,506,0,179,0,319,0,0,213,0
,0,319,0 <D21B>
150 DATA 478,0,190,0,319,0,0,239,0
,0,319,0,358,0,213,0,284,0,358,0,2
01,0,284,0,379,0,190,0,301,0,0,253
,0,0,301,0,758,0,190,0,284,0,0,239
,0,0,284,0,379,0,190,0,301,0,0,253
,0,0,301,0,758,0,190,0,284,0,0,239
,0,0,284,0 <D4C6>
160 DATA 379,0,190,0,301,0,0,253,0
,0,301,0,758,0,190,0,284,0,0,239,0
,0,284,0,379,253,190 <5654>
170 DATA 379,0,239,0,319,0,426,0,2
53,0,319,0,379,0,239,0,319,0,506,0
,213,0,319,0,478,0,190,0,319,0,0,3
79,0,0,319,0,478,0,239,0,319,0,0,3
79,0,0,319,0,506,0,213,0,319,0,536
,0,225,0,319,0,506,0,213,0,319,0,4
78,0,190,0,319,0 <DA5C>
180 DATA 426,0,179,0,319,0,0,253,0
,0,319,0,506,0,213,0,319,0,0,426,0
,0,319,0,379,0,159,0,319,0,0,239,0
,0,319,0,426,0,179,0,319,0,0,253,0
,0,319,0,478,0,190,0,319,0,0,239,0
,0,319,0,358,0,213,0,190,0,0,179,0
,0,213,0 <D2C8>
190 DATA 638,0,239,0,379,0,0,319,0
,0,379,0,638,0,253,0,426,0,0,319,0
,0,426,0,478,0,239,0,379,0,0,319,0
,0,379,0,0,478,0,478,0,239,478,0,3
19,0,319,0,506,0,319,638,0,179 <A478>
200 DATA 478,0,179,0,190,0,358,0,1
90,0,213,0,319,0,213,0,239,0,638,0

```

```

,239,0,253,0,478,0,239,478,0,319,0
,319,0,506,0,319,638,0,179,478,0,1
79,0,190,0,358,0,190,0,213,0,319,0
,213,0,239,0,638,0,239,0,253,0 <C6F7>
210 DATA 638,0,239,0,379,0,0,319,0
,0,379,0,638,0,253,0,426,0,0,319,0
,0,426,0,478,0,239,0,379,0,0,319,0
,0,379,0,0,478,0 <74A3>
220 DATA 1,1,1 <0C4F>
230 DIM ml(242),lm(242),fa(242):RE
STORE 100:FOR a=1 TO 242:READ b,c,
d:ml(a)=b:fa(a)=c:lm(a)=d:NEXT a:R
ANDOMIZE TIME:EVERY 5,0 GOSUB 440 <89CB>
240 LOCATE 6,11:PRINT"Optimal 40
":LOCATE 6,9:INPUT"Laenge :";la:IF
la>=9999 THEN RUN ELSE IF la<=19
OR la>=61 THEN 240 <56D8>
250 LOCATE 6,9:PRINT SPACE$(16);:L
OCATE 6,11:PRINT SPACE$(20); <2117>
260 ri=0:FOR a=1 TO 242:ri=ri+1:IF
fa(a)=0 AND ri=1 THEN fa(a)=ml(a)
+4 ELSE IF fa(a)=0 AND ri=2 THEN f
a(a)=lm(a)+4:ri=0 <9048>
270 ci=ci+1:ON ci GOTO 280,290,300
<25A7>
280 k1=1:k2=2:k3=4:GOTO 310 <21E7>
290 k1=2:k2=4:k3=1:GOTO 310 <21F0>
300 k1=4:k2=1:k3=2:ci=0 <24DA>
310 IF ml(a+1)=0 AND ml(a+2)=0 AND
ml(a+3)=0 AND ml(a+4)=0 AND lm(a+
1)=0 AND lm(a+2)=0 AND lm(a+3)=0 A
ND lm(a+4)=0 THEN 400 <89B2>
320 IF ml(a+1)=0 AND ml(a+2)=0 AND
ml(a+3)=0 AND lm(a+1)=0 AND lm(a+
2)=0 AND lm(a+3)=0 THEN 390 <69EC>
330 IF ml(a+1)=0 AND lm(a+1)=0 THE
N 380 <298B>
340 IF a=142 THEN la1=la*5:p=4 ELS
E la1=la:p=2 <3809>
350 SOUND k1,ml(a),la1,0,p,p:SOUND
k3,lm(a),la1,0,p,p:SOUND k2,fa(a)
,la1,0,p,2 <73A9>
360 NEXT a:LOCATE 6,9:PRINT"Nachma
l ?" <1CB8>
370 a$=INKEY$:IF a$=""THEN 370 ELS
E IF a$="j"OR a$="J"OR a$="y"OR a$
="Y"THEN SOUND 7,0,1:GOTO 240 ELSE
430 <50E4>
380 SOUND k1,ml(a),la*2,0,1,1:SOUN
D k3,lm(a),la*2,0,1,1:SOUND k2,fa(
a),la,0,2,6:a=a+1:SOUND k2,fa(a),l
a,0,2,2:GOTO 360 <951E>
390 SOUND k1,ml(a),la*4,0,3,3:SOUN
D k3,lm(a),la*4,0,3,3:SOUND k2,fa(
a),la,0,2,6:SOUND k2,fa(a+1),la,0,
2,2:SOUND k2,fa(a+2),la,0,2,2:SOUN
D k2,fa(a+3),la,0,2,2:a=a+3:GOTO 3
60 <D9DA>
400 IF a=238 THEN 420 <1180>
410 SOUND k1,ml(a),la*5,0,4,4:SOUN
D k3,lm(a),la*5,0,4,4:SOUND k2,fa(
a),la,0,2,6:SOUND k2,fa(a+1),la,0,
2,2:SOUND k2,fa(a+2),la,0,2,2:SOUN
D k2,fa(a+3),la,0,2,2:SOUND k2,fa(
a+4),la,0,2,2:a=a+4:GOTO 360 <FADE>
420 SOUND k1,ml(a),la*7,0,5,5:SOUN
D k3,lm(a),la*7,0,5,5:SOUND k2,fa(
a),la,0,2,6:SOUND k2,fa(a+1),la,0,
2,2:SOUND k2,fa(a+2),la,0,2,2:SOUN
D k2,fa(a+3),la,0,2,2:SOUND k2,fa(
a+4),la*3,0,1,2:a=a+5:GOTO 360 <FC3B>
430 MODE 1:BORDER 1:INK 0,1:PAPER
0:INK 1,24:PEN 1:END <1DC0>
440 Q=INT(RND*25)+1:INK 1,Q:RETURN
<1FEA>

10 '***** <2397>
12 '* MUSIKDEMO 2 * <2313>
14 '* CAPRICCIO * <237F>
16 '* VON * <2359>
18 '* MANFRED LIPPOWSKI * <2357>
20 '* FUER * <2309>
22 '* SCHNEIDER CPC-WELT * <23F1>
24 '* CPC 464/664/6128 JE* <2380>
26 '***** <23B7>
40 INK 1,26:INK 2,24:INK 3,0:CLS:M
ODE 1:BORDER 0:INK 0,0:PAPER 0:PEN
3:LOCATE 1,25:PRINT" CPC Soundmac
hine":PEN 1 <46B0>
50 PEN 2:LOCATE 3,3:PRINT CHR$(164
)+" Manfred Lipowski":LOCATE 7,5:P
RINT"Juli 1987":LOCATE 9,7:PRINT"P
art 2" <49DB>
60 FOR a=0 TO 290:FOR b=2 TO 24 ST
EP 2:ORIGIN 0,0:IF TEST(a,b)=0 THE
N 80 <373A>
70 ORIGIN 25,25:c=a*2:PLOT a*2,b*2
+c/2.2+2:PLOT a*2,b*2+c/2.2,1 <51B7>
80 NEXT:NEXT <0870>
90 LOCATE 1,25:PRINT SPACE$(22):IN
K 3,2:PEN 3:LOCATE 26,16:PRINT"Cap
riccio":LOCATE 26,18:PRINT"in a-mo
ll":LOCATE 29,20:PRINT"von":LOCATE
24,22:PRINT"Mauro Giuliani":LOCAT
E 26,24:PRINT"(1781-1821)" <7F67>
100 RESTORE 360:DIM ml(519):FOR a=
1 TO 514:READ b:ml(a)=b:NEXT <3AD4>
110 dau=100:aud=350 <1925>
120 ENV 1,15,-1,12:ENT-1,1,-2,6,1,
4,6,1,-2,6:ENV 2,15,-1,8:ENV 3,1,1
5,26 <3E43>
130 EVERY 5,1 GOSUB 530 <0D9B>
140 LOCATE 4,9:PRINT"Originalversi
on ?" <1EF5>

```

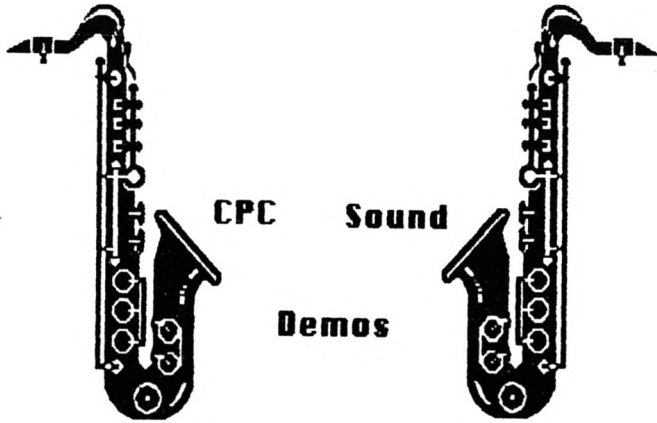
LISTING

```

150 a$=INKEY$:IF a$=""THEN 150 ELS                                     <86AB>
E IF a$="j"OR a$="J"OR a$="y"OR a$                                     400 DATA 478,319,190,319,159,319,1
="Y"THEN LOCATE 4,9:PRINT SPACE$(1                                     79,319,190,319,638,319,379,319,426
8);:GOTO 240                                                         <520C>          ,319,478,319,190,319,159,319,179,3
160 LOCATE 4,9:INPUT"Laenge (10-3                                     19,190,319,638,319,379,319,426,319
0):";la:IF la>9999 THEN 40 ELSE IF                                     <86BD>
la<=9 OR la=31 THEN 160 ELSE LOC                                     410 DATA 451,284,190,284,142,284,1
ATE 4,9:PRINT SPACE$(20);                                           <58AC>          59,284,179,284,451,284,426,284,478
170 SOUND 135,0,1                                                    <0C72>          ,284,506,319,213,319,159,319,179,3
180 fabi=0:FOR b=1 TO 514                                             <1AD3>          19,190,319,506,319,478,319,506,319
190 SOUND 1,ml(b),la,0,2,2:SOUND 4                                     <86DB>
,ml(b)+2,la,0,2,2                                                    <3CE4>          420 DATA 568,358,239,358,179,358,1
200 fabi=fabi+1:IF fabi=4 THEN 210                                     90,358,213,358,536,358,506,358,568
ELSE 220                                                             <2AC3>          ,358,602,379,253,379,190,379,213,3
210 SOUND 2,ml(b)*2,la*4,0,3,3:fab                                     79,239,379,602,379,568,379,638,379
i=0                                                                    <2DD9>          <8613>
220 NEXT:GOTO 110                                                    <0B91>          430 DATA 716,358,284,358,716,358,2
230 '                                                                  <07D6>          84,358,716,426,284,426,716,426,284
240 SOUND 135,0,1:FOR b=1 TO 514                                     <1868>          ,426,758,379,301,379,253,319,190,2
250 SOUND 1,ml(b),dau,0,1,1:GOSUB                                     <86AE>          53,150,190,127,150,190,127,150,190
350                                                                    <255E>
260 SOUND 2,ml(b+1),dau,0,1,1:GOSU                                     440 DATA 758,379,284,379,239,284,1
B 350                                                                  <2739>          90,239,142,190,119,142,190,119,142
270 SOUND 4,ml(b+2),dau,0,1,1:GOSU                                     <2756>          ,190,758,379,301,379,253,319,190,2
B 350                                                                  <27F5>          53,150,190,127,150,190,127,150,190
280 SOUND 1,ml(b+3),dau,0,1,1:GOSU                                     <864C>
B 350                                                                  <27D5>          450 DATA 758,379,284,379,239,284,1
290 SOUND 2,ml(b+4),dau,0,1,1:GOSU                                     <2702>          90,239,142,190,119,142,190,119,142
B 350                                                                  <1085>          ,190,758,301,253,190,758,284,239,1
300 SOUND 4,ml(b+5),dau,0,1,1:GOSU                                     <0644>          90,758,301,253,190,758,284,239,190
B 350                                                                  <164E>          460 DATA 758,301,253,190,426,301,2
310 b=b+5                                                            <485E>          53,190,478,284,239,190,506,301,213
320 NEXT                                                            <2C78>          ,190,568,284,758,284,478,284,506,2
330 LOCATE 8,9:PRINT"Nochmal ?"                                       <86DD>          84,568,284,758,284,478,284,506,284
340 a$=INKEY$:IF a$=""THEN 340 ELS                                     470 DATA 568,190,201,190,213,190,2
E IF a$="j"OR a$="J"OR a$="y"OR a$                                     39,190,253,190,284,190,301,190,284
="Y"THEN 110 ELSE 500                                                 <864D>          ,190,426,213,568,213,358,213,379,2
350 aud=aud-25:FOR f=1 TO aud:NEXT                                     13,426,213,568,213,358,213,379,213
f:RETURN                                                                <8619>
360 DATA 568,190,201,190,239,190,2                                     480 DATA 426,284,213,284,179,284,1
53,190,284,190,758,190,478,190,506                                     90,284,213,284,179,284,190,284,213
,190,568,190,201,190,239,190,253,1                                     <86E8>          ,284,758,284,239,284,179,284,190,2
90,284,190,758,190,478,190,568,190                                     <86D9>          84,758,284,239,284,179,284,190,284
<86D5>
370 DATA 602,190,201,190,213,190,2                                     490 DATA 758,284,253,284,179,284,1
39,190,253,190,758,190,426,190,478                                     90,284,758,301,253,301,179,319,190
,190,506,190,201,190,213,190,239,1                                     <86E9>          ,319,568,190,201,190,239,190,284,1
90,253,190,758,190,602,190,758,190                                     90,758,190,201,190,213,190,253,190
<86C6>
380 DATA 568,190,201,190,239,190,2                                     <86E8>          500 DATA 568,190,201,190,239,190,2
53,190,239,190,758,190,478,190,506                                     53,190,284,190,379,190,478,190,506
,190,568,190,201,190,239,190,253,1                                     <8623>          ,190,568,190,201,190,239,190,253,1
90,284,190,758,190,478,190,568,190                                     90,284,190,379,190,478,190,506,190
<86D9>
390 DATA 638,319,253,319,179,319,1                                     510 DATA 568,190,201,190,239,190,2
90,319,213,319,638,319,426,319,478                                     53,190,284,190,379,190,478,190,506
,319,506,319,213,319,179,319,190,3                                     <8623>
19,213,319,638,319,506,319,638,319

```

```
,190,568,284,758,284,478,284,506,2
84,568,284,758,284,478,284,506,284
,568,568,568,568,716,758,758 <A26C>
520 MODE 1:BORDER 1:INK 0,1:PAPER
0:INK 1,24:PEN 1:END <1D74>
530 Q=INT(RND*25)+1:INK 1,Q:RETURN
<1F9E>
```



```
10 '***** <2397>
12 '* MUSIKDEMO 3 * <232B>
14 '* SPANISCHE ROMANZE * <2376>
16 '* VON * <2359>
18 '* MANFRED LIPPOWSKI * <2357>
20 '* FUER * <2309>
22 '* SCHNEIDER CPC-WELT * <23F1>
24 '* CPC 464/664/6128 JE* <2380>
26 '***** <23B7>
30 PEN 2:LOCATE 3,3:PRINT CHR$(164
)+ " Manfred Lipowski":LOCATE 7,5:P
RINT "AUGUST 1987":LOCATE 9,7:PRINT
"Part 3" <4B7F>
40 FOR a=0 TO 290:FOR b=2 TO 24 ST
EP 2:ORIGIN 0,0:IF TEST(a,b)=0 THE
N 60 <3773>
50 ORIGIN 25,25:c=a*2:PLOT a*2,b*2
+c/2.2+2:PLOT a*2,b*2+c/2.2,1 <518F>
60 NEXT:NEXT <0848>
70 LOCATE 1,25:PRINT SPACE$(22):IN
K 3,2:PEN 3:LOCATE 26,16:PRINT "Spa
nische":LOCATE 27,18:PRINT "Romanze
":LOCATE 28,20:PRINT "Autor":LOCATE
28,22:PRINT "Anonym":PEN 1:LOCATE
26,24:PRINT "Bitte Warten" <7B18>
80 ' <07AB>
90 la=20:'Kann bei Belieben abgeae
ndert werden,sollte jedo
ch 10 bzw 30 nicht unt
er-ueberschreiten Variab
le >la< ist zustaendig
fuer die Soundlaenge. <C586>
100 ENV 1,15,-1,23:ENV 2,15,-1,10:
ENV 3,15,-1,7 <27BA>
110 RESTORE 280:DIM ML(272),LM(272
```

```
,FK(272):FOR a=1 TO 272:READ b,c,
d:ML(A)=B:LM(A)=D:FK(A)=C:NEXT A <8035>
120 LOCATE 26,24:PRINT SPACE$(12);
<148C>
130 'Parser: (c) M.Lipowski 87 <20E4>
140 EVERY 5,0 GOSUB 240 <0D54>
150 LOCATE 6,9:PRINT "Optimal >20<
";:LOCATE 2,11:INPUT "Laenge (10-30
)":":la:IF la<10 OR la>30 THEN 150 <51CD>
160 LOCATE 2,9:PRINT SPACE$(19);:L
OCATE 2,11:PRINT SPACE$(23);:FOR A
=1 TO 272 <2D03>
170 IF ml(a)<>0 AND lm(a)<>0 AND f
k(a)<>0 THEN 250 <3348>
180 IF ml(a+3)<>0 THEN 260 <198A>
190 SOUND 1,ml(a),la*9,0,1,1:SOUND
2,ml(a)+2,la,15:SOUND 4,lm(a),la*
3,0,2,2:SOUND 4,lm(a+3),la*3,0,2,2
:SOUND 4,lm(a+6),la*3,0,2,2 <9691>
200 FOR a1=1 TO 8:SOUND 2,fk(a+a1)
,la,0,3,3:NEXT a1 <3771>
210 a=a+8 <10B3>
220 NEXT:LOCATE 6,9:PRINT "Nochmal
? j/n"; <1DC0>
230 a$=INKEY$:IF a$="" THEN 230 ELS
E IF a$="j" OR a$="J" THEN 150 ELSE
MODE 1:INK 1,24:BORDER 1:INK 0,1:P
APER 0:PEN 1:END <4B8F>
240 INK 1,RND*25+1:RETURN <1170>
250 SOUND 1,ml(a),la*8,0,1,1:SOUND
4,lm(a),la*8,0,1,2:SOUND 2,fk(a),
la*8,0,1,3:GOTO 220 <6009>
260 SOUND 1,ml(a),la*3,0,2,1:SOUND
2,fk(a),la,0,3,3:SOUND 4,lm(a),la
*3,0,2,2:SOUND 2,fk(a+1),la,0,3,3:
SOUND 2,fk(a+2),la,0,3,3:a=a+2:GOT
O 220 <A48E>
270 'Daten fuers Musikstueck:1. Za
hl: BASS/2. ZAHL: BEGLEITUNG/3. ZA
HL: OBERSTIMME <5375>
280 DATA 758,0,127,0,253,0,0,319,0
,0,0,127,0,253,0,0,319,0,0,0,127,0
,253,0,0,319,0,758,0,127,0,253,0,0
,319,0,0,0,142,0,253,0,0,319,0,0,0
,159,0,253,0,0,319,0 <9AF2>
290 DATA 758,0,159,0,253,0,0,319,0
,0,0,169,0,253,0,0,319,0,0,0,190,0
,253,0,0,319,0,758,0,190,0,253,0,0
,319,0,0,0,159,0,253,0,0,319,0,0,0
,127,0,253,0,0,319,0 <9A0A>
300 DATA 758,0,95,0,253,0,0,319,0,
0,0,95,0,253,0,0,319,0,0,0,95,0,25
3,0,0,319,0,758,0,95,0,253,0,0,319
,0,0,0,106,0,253,0,0,319,0,0,0,119
,0,253,0,0,319,0 <9635>
310 DATA 568,0,119,0,190,0,0,239,0
,0,0,127,0,190,0,0,239,0,0,0,142,0
,190,0,0,239,0,568,0,142,0,190,0,0
```

```
,239,0,0,0,127,0,190,0,0,239,0,0,0
,119,0,190,0,0,239,0 <9AD3>
320 DATA 506,0,127,0,169,0,0,201,0
,0,0,119,0,169,0,0,201,0,0,0,127,0
,169,0,0,201,0,506,0,100,0,169,0,0
,201,0,0,0,119,0,169,0,0,201,0,0,0
,127,0,169,0,0,201,0 <9AF9>
330 DATA 758,0,127,0,253,0,0,319,0
,0,0,142,0,253,0,0,319,0,0,0,159,0
,253,0,0,319,0,758,0,159,0,253,0,0
,319,0,0,0,169,0,253,0,0,319,0,0,0
,190,0,253,0,0,319,0 <9A7E>
340 DATA 506,0,169,0,253,0,0,284,0
,0,0,169,0,253,0,0,284,0,0,0,169,0
,253,0,0,284,0,506,0,169,0,253,0,0
,284,0,0,0,159,0,253,0,0,284,0,0,0
,169,0,253,0,0,284,0 <9AD0>
350 DATA 379,0,190,0,253,0,0,319,0
,506,0,190,0,253,0,0,319,0,638,0,1
90,0,253,0,0,319,0,758,319,190 <6072>
360 DATA 758,0,150,0,253,0,0,301,0
,0,0,150,0,253,0,0,301,0,0,0,150,0
,253,0,0,301,0,758,0,150,0,253,0,0
,301,0,0,0,169,0,253,0,0,301,0,0,0
,190,0,253,0,0,301,0 <9A38>
370 DATA 506,0,190,0,284,0,0,338,0
,0,0,201,0,284,0,0,338,0,0,0,201,0
,284,0,0,338,0,506,0,201,0,284,0,0
,338,0,0,0,213,0,284,0,0,338,0,0,0
,201,0,284,0,0,338,0 <9AD0>
380 DATA 506,0,113,0,169,0,0,201,0
,0,0,113,0,169,0,0,201,0,0,0,113,0
,169,0,0,201,0,506,0,113,0,169,0,0
,201,0,0,0,100,0,169,0,0,201,0,0,0
,113,0,169,0,0,201,0 <9A4A>
390 DATA 758,0,113,0,150,0,0,190,0
,0,0,127,0,150,0,0,190,0,0,0,127,0
,150,0,0,190,0,758,0,127,0,150,0,0
,190,0,0,0,113,0,150,0,0,190,0,0,0
,100,0,150,0,0,190,0 <9AD0>
400 DATA 758,0,95,0,150,0,0,190,0,
0,0,95,0,150,0,0,190,0,0,0,95,0,15
0,0,0,190,0,758,0,95,0,150,0,0,190
,0,0,0,100,0,150,0,0,190,0,0,0,106
,0,150,0,0,190,0 <9618>
410 DATA 506,0,113,0,190,0,0,225,0
,0,0,113,0,190,0,0,225,0,0,0,113,0
,190,0,0,225,0,506,0,113,0,190,0,0
,225,0,0,0,127,0,190,0,0,225,0,0,0
,142,0,190,0,0,225,0 <9AA7>
420 DATA 758,0,150,0,253,0,0,301,0
,0,0,150,0,253,0,0,301,0,0,0,150,0
,253,0,0,301,0,506,0,150,0,201,0,0
,284,0,0,0,142,0,201,0,0,284,0,0,0
,169,0,201,0,0,284,0 <9AE0>
430 DATA 379,0,190,0,253,0,0,301,0
,506,0,190,0,253,0,0,301,0,602,0,1
90,0,253,0,0,301,0,758,301,190 <6074>
```

SPEED UP

Wer hat sich nicht schon manchmal geärgert, daß die CPC-Typen hinsichtlich der Bildschirmausgabe nicht zu den schnellsten Rechnern gehören? Unser Autor Michael Folz hat nun ein Programm geschrieben, das die Bildschirmausgabe auf allen CPC im Mode erheblich beschleunigt.

Liest man das ROM-Listing durch, so fällt einem auf, daß die Amstrad-Programmierer sich hier einfach zu wenig Gedanken gemacht haben: Um ein Zeichen im Mode 2 auszugeben, wird die gleiche umständliche Routine benutzt wie auch im Mode 0 oder 1!

Was dem Programmierer dann noch übrig bleibt, ist die Erstellung einer neuen, schnelleren Ausgaberroutine speziell für den Mode 2.

Alle anderen Routinen, die uns bekannt sind, brachten Beschleunigung um etwa 20 bis 30 %, zum anderen legten sie dem Benutzer Beschränkungen auf, wie z.B. der nur noch begrenzt mögliche Einsatz von Windows, die Notwendigkeit, die Routine vor Mode-Wechseln auszuschalten usw. usw.

Speed Up ist da ganz anders. Der Beschleunigungsfaktor liegt, in der Theorie, bei etwa 65 bis 70 % (!), wovon in der Praxis jedoch noch die Zeit für Bildschirmscrolling usw. abgeht, so daß man im Endeffekt auf eine um durchschnittlich etwa 50 % schnellere Bildschirmausgabe im Mode 2 kommt.

Beschränkungen bezüglich der Windows gibt es keine, Speed Up funktioniert hier ganz wie die normale Systemausgabe auch. Auch die transparente Darstellung von Zeichen (einschalten mit CHR\$(22) CHR\$(1)) ist ohne weiteres möglich. Wechselt man in einen anderen Mode als den Mode 2, deaktiviert Speed Up sich automatisch.

Das Programm ist mit anderen Routinen, die die Bildschirmausgabe patchen, durchaus kompatibel, jedoch sollten dann diese Routinen vor Speed Up installiert werden.

JE

```
1 '***** <24B1>
2 '* SPEED-UP * <2440>
3 '* VON * <24F2>
4 '* MICHAEL FOLZ * <2412>
5 '* FUER * <2424>
6 '* SCHNEIDER CPC-WELT * <24EF>
7 '* CPC 464/664/6128 JE* <2453>
8 '***** <24BF>
10 MEMORY &A5FF <09B5>
650 a=&A600:e=&A6C1:zb=1000:e=e+1 <2C69>
660 FOR i=a TO e:IF i=e THEN SAVE"
SPEED-UP.BIN",B,&A600,&C1:CLS:PRIN
T"STARTEN MIT >SYMBOL AFTER 256":P
RINT">MEMORY &A5FF":PRINT"LOAD >SP
EED-UP.BIN":PRINT"CALL &A600":CALL
&A600:END <9B68>
670 READ d$:POKE i,VAL("&"+d$) <1DEA>
730 IF i<e THEN NEXT i <1521>
```

```

1001 DATA 21,D3,BD,E5,11,BB,A6,01 <1E76>
1002 DATA 03,00,C5,ED,B0,EB,C1,D1 <1ED3>
1003 DATA ED,B0,3A,06,00,FE,91,C8 <1ECF>
1004 DATA DD,21,55,A6,1E,06,FE,80 <1EA9>
1005 DATA 28,06,DD,21,6D,A6,1E,02 <1EF7>
1006 DATA DD,56,01,01,4C,00,21,75 <1E4F>
1007 DATA A6,DD,7E,00,ED,B1,20,11 <1ECD>
1008 DATA 7E,BA,20,F5,DD,7E,03,77 <1E8D>
1009 DATA DD,7E,02,2B,77,23,23,18 <1E05>
1010 DATA E8,1D,C8,DD,23,DD,23,DD <1EC5>
1011 DATA 23,DD,23,18,D3,C3,B7,C8 <1E88>
1012 DATA B1,29,B7,88,B2,C4,B7,C9 <1EDA>
1013 DATA B1,C6,B7,CB,B1,D4,12,D3 <1E0A>
1014 DATA 12,1F,0C,13,0C,D4,12,D0 <1E5F>
1015 DATA 12,1F,0C,1B,0C,47,3A,C3 <1E74>
1016 DATA B7,FE,02,78,C2,BB,A6,E5 <1EB5>
1017 DATA CD,A2,BB,21,B2,A6,36,00 <1EF2>
1018 DATA 28,02,36,B6,E1,4C,26,00 <1E5E>
1019 DATA 5D,54,29,29,19,29,29,29 <1E0A>
1020 DATA 29,59,19,ED,5B,C4,B7,19 <1ED9>
1021 DATA 7C,E6,07,67,3A,C6,B7,84 <1EAD>
1022 DATA 67,EB,78,CD,D4,12,EB,06 <1E95>
1023 DATA 08,1A,B6,77,13,CD,1F,0C <1E4B>
1024 DATA 10,F7,C9,00,00,00,C3,75 <1ECC>
1025 DATA A6,28 <0CE6>

```

```

.0 -" <5FA4>
30 PRINT STRING$(80,"=") <1072>
40 PRINT:PRINT <0853>
50 LOCATE 1,7:INPUT"Welches Lauf
werk benutzen Sie [A/B]
:" ,DR$ <48DF>
60 IF DR$="" THEN 50 <120F>
70 IF UPPER$(DR$)<>"B" THEN DR=0 EL
SE DR=1 <248C>
80 LOCATE 1,9:INPUT"Welches Format
hat die Diskette <C>P/M oder <D>a
ten :",FORM$ <4ADF>
90 IF FORM$="" THEN 80 <14DB>
100 IF UPPER$(FORM$)="C" THEN TR=2:
SEC=65:MSEC=68:GOTO 120 <3722>
110 TR=0:SEC=193:MSEC=196 <21F8>
120 LOCATE 1,11:INPUT"Welche Use
rnummer hat die Diskette (
0-15) :",USER$ <4B01>
130 IF USER$="" THEN 120 <1448>
140 IF USER>15 THEN 120 <141B>
150 LOCATE 1,23:PRINT"Bitte legen
Sie die Diskette mit dem zu retten
den Programm ein .Dann <T A S T E>
":CALL &BB18:CLS <659D>
160 MEMORY &A000-1:FOR adr=&A000 T
O &A01C:READ byte:POKE adr,byte:NE
XT ADR <3C66>
170 POKE &A021,TR:POKE &A020,DR:PO
KE &A022,SEC:ADR=&A030:CALL &A000 <36C2>
180 IF PEEK(ADR)=&E5 THEN 200 <1898>
190 ADR=ADR+&20:IF ADR>&A230 THEN
260 ELSE 180 <2B93>
200 IF PEEK(ADR+1)<>&E5 THEN FOR L
=ADR+1 TO ADR+8:PRINT CHR$(PEEK(L)
);:NEXT:ELSE 190 <45F0>
210 PRINT"."+CHR$(PEEK(ADR+9))+CHR
$(PEEK(ADR+10))+CHR$(PEEK(ADR+11))
; <3F33>
220 PRINT" Programm retten [J/N]
? "; <24AD>
230 A$=UPPER$(INKEY$):IF A$="" THEN
230 <1D6B>
240 IF A$="J" THEN POKE ADR,USER::P
RINT"J" ELSE PRINT"N" <2AC7>
250 GOTO 190 <09D7>
260 POKE &A01C,&85:CALL &A000:POKE
&A01C,&84 <1B48>
270 SEC=SEC+1:IF SEC>MSEC THEN CLS
:/USER,user:CAT:ELSE 170 <3DA8>
280 DATA &21,&1C,&A0,&CD,&D4,&BC,&
22,&1D <269A>
290 DATA &A0,&79,&32,&1F,&A0,&21,&
20,&A0 <26F3>
300 DATA &5E,&23,&56,&23,&4E,&21,&
30,&A0 <269B>
310 DATA &DF,&1D,&A0,&C0,&84 <1A96>

```

ANTI-ERA

Wem ist das nicht schon einmal passiert? Sie wollten Ihre Diskette aufräumen und alle überflüssigen Programme löschen, benutzten dazu den Amsdos-Befehl ERA und löschten versehentlich ein Programm, das Sie noch benötigen!

Ihnen noch nicht? Dann können Sie von Glück sagen. Für alle anderen hier ein Programm, das solche Files wieder rettet. Voraussetzung ist allerdings, daß Sie zwischenzeitlich keine neuen Programme abgespeichert haben.

Nach dem Start dieses Programms müssen Sie angeben, welches Laufwerk benutzt werden soll, welche Usernummer benutzt wurde (normalerweise 0) und in welchem Format (Data- oder System-Format) die Diskette ursprünglich formatiert wurde. Das Programm läuft auf allen CPC's mit Diskettenstation.

```

1 '***** <2506>
2 '* ANTI-ERA * <25BB>
3 '* VON * <25F6>
4 '* CHRISTIAN PRELLA * <25A9>
5 '* FUER * <2558>
6 '* SCHNEIDER CPC-WELT * <2539>
7 '* CPC 464/664/6128 JE* <255F>
8 '***** <2514>
10 INK 0,0:BORDER 0:INK 1,18 <1255>
20 MODE 2:CLS:PRINT:PRINT"- ANTI-E
RA stellt gelöschte Programme
wieder her by CH-Soft 1987 V3

```

POLYGRAF

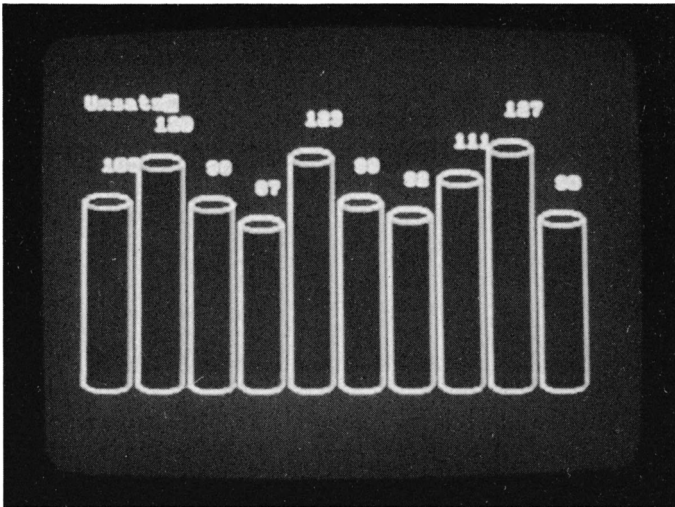
Dieses von unserem Autor Werner Arndt geschriebene Listing ist ein vielseitiges Grafikprogramm für den CPC 664/6128 zur Darstellung von Diagrammen, wobei zwischen vielen Darstellungsarten (Balken, Linien, Torte, Säulen usw.) gewählt werden kann. Die Darstellungen lassen sich auch in bequemer Weise beschriften.

Nach dem Programmstart erscheint ein Menü mit fünf Auswahlmöglichkeiten. Will man die erstellte Grafik später ausdrucken (auf Schneider NLQ 401 oder DMP 3000), so wird mit der Nummer 1 die Hardcopyroutine in einen gesicherten Speicherbereich geladen. Im späteren Programm wird der Menüpunkt 1 nicht mehr erneut aufgerufen, da sonst eine Memory full Meldung erscheinen würde. Sollte dies jedoch einmal passieren, startet man wieder mit run.

Der Menüpunkt 2 dient der Schriftauswahl. Es kann in den drei bekannten Schriftmodi 0, 1, 2 gearbeitet werden.

Die Nummer 3 erlaubt es, eine bereits abgespeicherte Grafik zu laden. Dabei ist zu beachten, daß diese auch mit dem gleichen Modus wieder geladen wird, unter dem sie gespeichert wurde. Es empfiehlt sich, in der Bezeichnung der abgespeicherten Grafik den Modus mit zu vermerken (z.B. g3mod0 = Grafik im Modus 0 abgespeichert).

Will man eine erstellte Grafik speichern, so muß diese zuvor unter Nummer 4 eine Bezeichnung erhalten. Hat man dies einmal vergessen, so wird die Grafik trotzdem abgespeichert und zwar unter der Bezeichnung Grafik.



Hat man eine bereits abgespeicherte Grafik geladen und diese mit dem Cursor anders beschriftet, so wird sie wieder unter dem selben Namen gespeichert (mit Pfeiltaste Cursor entfernen und >s< drücken).

Als letzter Menüpunkt erscheint die Nummer 5. Je nach Bedarf und Art der grafisch darzustellenden Werte kann man aus neun Grafikarten die geeignete auswählen. Soll keine automatische Darstellung der

Werte über oder in den Grafiken erfolgen, so ist >n< einzugeben. Insbesondere bei langen Zahlen und einer hohen Anzahl kann es zu Überschneidungen kommen. Hier sollten die Werte manuell richtig positioniert werden.

Mit dem Cursor, der oben links erscheint, können jetzt beliebig Text und Sonderzeichen vom Zehnerblock eingegeben werden.

Ist man fertig und hat die Grafik vorher bezeichnet (Nummer 4), so entfernt man den Cursor durch Drücken der Pfeiltaste unter dem Pfundzeichen. Dann wird mit >s< die Grafik abgespeichert.

Nach diesem Vorgang erscheint wieder der Cursor. Will man nun die Grafik ausdrucken, so entfernt man wieder den Cursor mit der Pfeiltaste und drückt die >d< Taste. Der Ausdruck erfolgt nur dann, wenn am Anfang unter Menüpunkt 1 die Hardcopy geladen wurde.

JE

```

10 '***** <24C3>
20 '* POLYGRAF * <240B>
30 '* VON * <2429>
40 '* WERNER ARNDT * <2410>
50 '* FUER * <247E>
60 '* SCHNEIDER CPC-WELT * <245A>
70 '* CPC 664/6128 JE* <245E>
80 '***** <244E>
130 ' <070E>
140 MODE 2: CLEAR: GOSUB 1400 <0E71>
150 ' <0736>
160 '***** <23C2>
170 '* Auswahl der Grafik * <234F>
180 '***** <23EA>
190 ' <0786>
200 MODE 1: schrift=1:a=1 <1B0A>
210 CLS: PRINT CHR$(150); STRING$(38 <233F>
,CHR$(154));CHR$(156)
220 FOR i=0 TO 22 <0FB2>
230 LOCATE 1,2+i <0E07>
240 PRINT CHR$(149) <0C9F>
250 LOCATE 40,2+i:PRINT CHR$(149) <174B>
260 NEXT <06CB>
270 LOCATE 1,24 <0A51>
280 PRINT CHR$(147);STRING$(38,CHR <217A>
$(154));CHR$(153)
290 IF a>5 OR a<1 THEN GOTO 310 <1855>
300 DIM wert(100):DIM x(100) <1BC6>
310 LOCATE 13,4:PRINT"POLYGR <1E1E>
AF"
320 LOCATE 15,23:PRINT CHR$(164)+" <2813>
Werner E.J. Arndt"
330 LOCATE 13,6:PRINT STRING$(15,C <194D>
HR$(134))
340 LOCATE 3,9:PRINT"1. Hardcopy" <1859>
350 LOCATE 3,11:PRINT"2. Schriftmo
    
```


LISTING

```

810 FOR i=1 TO 4000:NEXT:INK 1,26 <18D6> <1EA4>
820 CLS:GOTO 350 <0BDD> 1260 ' <07E6>
830 ' <078A> 1270 '***** <225D>
840 '***** <2315> 1280 '* Auswahl des Schriftmodus* <22AE>
850 '* Bezeichnen der Grafik * <23E3> 1290 '***** <2286>
860 '***** <233D> 1300 ' <0738>
870 ' <07DA> 1310 PRINT"Schrift":PRINT:PRINT:P
880 PRINT"Grafikbezeichnung":PRINT RINT <1657>
<1BF0> 1320 PRINT"gross > 0 <":PRINT <17AA>
890 PRINT"(max. 8 Zeichen)":PRINT <1A5E> 1330 PRINT"mittel > 1 <":PRINT <1799>
900 PRINT"_____":PR 1340 PRINT"klein > 2 <":PRINT:PR
INT <1E31> INT <1950>
910 PRINT"nach Grafik":PRINT <15F9> 1350 INPUT schrift <1042>
920 PRINT"Pfeiltaste druecken,":PR 1360 IF schrift>2 THEN GOTO 1310 <1797>
INT <1E22> 1370 MODE schrift <10DB>
930 PRINT"dann >s< = speichern":PR 1380 CLS:GOTO 360 <0B90>
INT <1EB3> 1390 ' <07EC>
940 PRINT:PRINT:INPUT"> ";bild$:PR 1400 '***** <2261>
INT <1865> 1410 '* Tastennumbelegung * <2201>
950 IF LEN(bild$)>8 THEN PRINT"Ueb 1420 '***** <2289>
er 8 Zeichen!":PRINT <2AE0> 1430 ' <073B>
960 IF LEN(bild$)>8 THEN GOTO 880 <1849> 1440 KEY DEF 1,1,9 <0CA7>
970 CLS:GOTO 1760 <0B29> 1450 KEY DEF 2,1,10 <0DD7>
980 ' <07B5> 1460 KEY DEF 8,1,8 <0C85>
990 ' <07C9> 1470 KEY DEF 0,1,11 <0D04>
1000 ' Laden einer Grafik <1A78> 1480 KEY DEF 10,1,249 <0EBE>
1010 ' <07F1> 1490 KEY DEF 11,1,248 <0E4D>
1020 ' <0706> 1500 KEY DEF 3,1,250 <0D7C>
1030 LOCATE 1,10:CLS <0C3A> 1510 KEY DEF 20,1,240 <0EC9>
1040 PRINT"Bezeichnung":PRINT <1514> 1520 KEY DEF 12,1,241 <0EED>
1050 PRINT"der":PRINT <0DD8> 1530 KEY DEF 4,1,242 <0D11>
1060 INPUT"Grafik ";bild$:PRINT:P 1540 KEY DEF 13,1,243 <0E9F>
RINT STRING$(16,154) <25A6> 1550 KEY DEF 14,1,166 <0EFE>
1070 IF LEN(bild$)>8 OR LEN(bild$) 1560 KEY DEF 5,1,231 <0D9A>
<1 THEN 1030 <25A3> 1570 SYMBOL AFTER 90 <096A>
1080 PRINT:PRINT"In welchem Modus" 1580 SYMBOL 91,&X1011010,&X111100,
:PRINT <1C3E> &X1100110,&X1100110,&X1111110,&X11
1090 PRINT"wurde die Grafik":PRINT 00110,&X1100110,&X0 <284C>
<1A9E> 1590 SYMBOL 92,&X10111010,&X110110
1100 PRINT"abgespeichert?":PRINT <19BF> 0,&X11000110,&X11000110,&X11000110
1110 PRINT:INPUT"Mode: ";schrift <1BAA> ,&X1101100,&X111000,&X0 <2866>
1120 IF schrift=0 OR schrift=1 OR 1600 SYMBOL 93,&X1100110,&X0,&X110
schrift=2 THEN 1130 ELSE 1030 <3585> 0110,&X1100110,&X1100110,&X1100110
1130 MODE schrift <10FA> ,&X111100,&X0 <2839>
1140 CLS:ON ERROR GOTO 1030 <0D59> 1610 SYMBOL 123,&X1001000,&X0,&X11
1150 LOAD bild$ <0D02> 11000,&X1100,&X1111100,&X11001100,
1160 GOTO 5700:'-- Beschriftung -- &X1110110,&X0 <285C>
<1EF1> 1620 SYMBOL 124,&X100100,&X0,&X111
1170 ' <0732> 100,&X1100110,&X1100110,&X1100110,
1180 '***** <22A8> &X111100,&X0 <28D6>
1190 '* Abspeichern einer Grafik* <2295> 1630 SYMBOL 125,&X0,&X1000100,&X0,
1200 '***** <22D0> &X1100110,&X1100110,&X1100110,&X11
1210 ' <0782> 1111,&X0 <2850>
1220 IF bild$=""THEN bild$="grafik 1640 SYMBOL 126,&X111000,&X1101100
" <21F7> ,&X1101100,&X1101100,&X1100110,&X1
1230 SAVE bild$,B,49152,16383 <1DA3> 110110,&X1101100,&X11000000 <28DF>
1240 bild$="" <0F42> 1650 KEY DEF 22,1,124,92 <1129>
1250 GOTO 5700:'-- Beschriftung -- 1660 KEY DEF 19,1,125,93 <11B5>

```

```

1670 KEY DEF 17,1,123,91 <1189>
1680 KEY DEF 26,1,126,96 <1165>
1690 KEY DEF 5,1,231 <0D9F>
1700 RETURN <0674>
1710 ' <076C>
1720 '***** <22E2>
1730 '* Auswahl der Grafiken * <227A>
1740 '***** <220B>
1750 ' <07BC>
1760 PRINT"Art der Grafik:" <1703>
1770 PRINT STRING$(20,154):PRINT <1191>
1780 PRINT"1.Saeule":PRINT <1298>
1790 PRINT"2.Saeule (3D)":PRINT <170E>
1800 PRINT"3.Bandgrafik":PRINT <1678>
1810 PRINT"4.Zylinder":PRINT <1471>
1820 PRINT"5.Zylinder (voll)":PRIN
T <1BA7>
1830 PRINT"6.Kreisgrafik":PRINT <1721>
1840 PRINT"7.Kreis + Segment":PRIN
T <1B36>
1850 PRINT"8.Blockgrafik (3D)":PRI
NT <1C67>
1860 PRINT"9.Kurvengrafik":PRINT:P
RINT:PRINT <1CFD>
1870 INPUT b <0AB5>
1880 IF b>9 OR b<1 THEN GOTO 1870 <189C>
1890 CLS <06F5>
1900 ON b GOTO 1920,2150,2770,2440
,3010,3350,4010,4730,5030 <2E87>
1910 ' <07FE>
1920 '***** <2273>
1930 '* Saeule * <22E8>
1940 '***** <229B>
1950 ' <074D>
1960 GOSUB 5380:'holen der Werte <1B6C>
1970 ' <0775>
1980 '-- Zeichnen der Grundlinie
-- <2557>
1990 ' <079D>
2000 CLS:MOVE 0,55:DRAW 640,55 <1472>
2010 ' <07C5>
2020 '-- Zeichnen der Werte -- <1F23>
2030 ' <07ED>
2040 GOSUB 5910:'Wertzuweisung <19D4>
2050 FOR i=1 TO anzahl <16B1>
2060 MOVE a,55:DRAW a,(d*wert(i)/m
a)+c <338F>
2070 DRAW a+x,(d*wert(i)/ma)+c <2F23>
2080 DRAW a+x,55 <12DE>
2090 IF beschrift$="n"THEN GOTO 21
20 <1B19>
2100 MOVE a+4,(d*wert(i)/ma+c)+20 <2F39>
2110 PRINT wert(i); <1492>
2120 a=a+x <13C7>
2130 NEXT i <0A1C>
2140 GOTO 5700:'-- Beschriftung --
<1E9D>
2150 ' <07DF>
2160 GOSUB 5380:'holen der Werte <1BFC>
2170 ' <0707>
2180 '-- Zeichnen der Grundlinie -
- <24B1>
2190 ' <072E>
2200 CLS:MOVE 0,55:DRAW 608,55 <14FA>
2210 ' <0756>
2220 '-- Zeichnen der Werte -- <1FB5>
2230 ' <077E>
2240 GOSUB 5910:'Wertzuweisung <1965>
2250 x=608/anzahl <170E>
2260 b=x/10 <1123>
2270 FOR i=1 TO anzahl <1669>
2280 MOVE a,55:DRAW a,(d*wert(i)/m
a)+c <3347>
2290 DRAW a+x,(d*wert(i)/ma)+c <2FDA>
2300 DRAW a+x,55 <1296>
2310 MOVE a,(d*wert(i)/ma)+c <2AD0>
2320 DRAW a+b,((d*wert(i)/ma)+c)+b
:IF wert(i+1)<=wert(i)THEN DRAW a+
(x)+b,((d*wert(i)/ma)+c)+b ELSE DR
AW a+x,((d*wert(i)/ma)+c)+b <C17B>
2330 IF wert(i+1)<=wert(i)THEN DRA
W a+x,(d*wert(i)/ma)+c:MOVE a+(x)+
b,((d*wert(i)/ma)+c)+b <8734>
2340 IF wert(i+1)<=wert(i)THEN DRA
W a+x+b,((d*wert(i+1))/ma)+c+b ELS
E DRAW a+x,55 <6BDD>
2350 IF beschrift$="n"THEN GOTO 23
70 <1BF5>
2360 MOVE a+b+1,((d*wert(i)/ma)+c)
+b+20:PRINT wert(i); <4B78>
2370 IF i=anzahl THEN GOTO 2400 <192C>
2380 a=a+x <13D0>
2390 NEXT i <0A25>
2400 MOVE a+x+b,55+b <1C36>
2410 DRAW a+x,55 <1274>
2420 ' <07FC>
2430 GOTO 5700:'-- Beschriftung --
<1EE2>
2440 ' <0723>
2450 GOSUB 5380 <0978>
2460 ' <074B>
2470 '-- Zeichnen der Werte -- <1FAA>
2480 ' <0773>
2490 CLS:DEG <0873>
2500 GOSUB 5910:'Wertzuweisung <196E>
2510 z=x-10 <1158>
2520 b=4 <0B1D>
2530 g=z/2-23 <139E>
2540 IF anzahl>4 THEN g=g+16 <1EEC>
2550 FOR i=1 TO anzahl:'-- Schleif
e fuer jeden einzelnen Wert --
<45AE>
2560 IF schrift=2 THEN b=1 <19CA>
2570 MOVE a,55:DRAW a,(d*wert(i)/m
a)+c <338D>
2580 ORIGIN a,(d*wert(i)/ma)+c <2A6F>

```

LISTING

```

2590 MOVE g,z/6 <1115>
2600 FOR f=1 TO 360 STEP 10:'-- Sc
hleife fuer obere Ellipse -- <3BBB>
2610 DRAW SIN(f)*z/2+z/2,COS(f)*z/
6 <2C41>
2620 NEXT f <0AE7>
2630 ORIGIN a,55 <0D99>
2640 MOVE g,z/6 <1179>
2650 FOR k=1 TO 360 STEP 10:'-- Sc
hleife fuer untere Ellipse -- <3CEE>
2660 DRAW SIN(k)*z/2+z/2,COS(k)*z/
6 <2CC2>
2670 NEXT k <0A5E>
2680 ORIGIN 0,0 <0982>
2690 MOVE a+z,(d*wert(i)/ma)+c:DRA
W a+z,55 <3DBB>
2700 IF beschrift$="n"THEN GOTO 27
30 <1BF8>
2710 MOVE a+b,(d*wert(i)/ma+c)+50 <321E>
2720 PRINT wert(i); <1459>
2730 a=a+z+10 <1647>
2740 NEXT i <0AE2>
2750 ' <0790>
2760 GOTO 5700:'-- Beschriftung --
<1E76>
2770 ' <07B8>
2780 GOSUB 5380:'holen der Werte <1BD7>
2790 ' <07E0>
2800 'Zeichnen der Grundlinie <1ED0>
2810 ' <0708>
2820 CLS:MOVE 0,55:DRAW 640,55 <14DE>
2830 ' <0732>
2840 '--- Zeichnen der Werte ---
<2391>
2850 ' <075A>
2860 GOSUB 5910:'Wertzuweisung <193F>
2870 b=4 <0BDB>
2880 FOR i=1 TO anzahl <162F>
2890 IF i=anzahl+1 THEN GOTO 5700 <1B57>
2900 IF schrift=2 THEN b=1 <1974>
2910 MOVE a,(d*wert(i)/ma)+c <2A81>
2920 DRAW a+x,(d*wert(i)/ma)+c <2FCA>
2930 IF i=anzahl THEN GOTO 2950 <19C5>
2940 DRAW a+x,(d*wert(i+1))/ma+c <31B8>
2950 IF beschrift$="n"THEN GOTO 29
80 <1BC1>
2960 MOVE a+b,(d*wert(i)/ma+c)+20 <329C>
2970 PRINT wert(i); <144E>
2980 a=a+x <1381>
2990 NEXT i <0AD7>
3000 GOTO 5700:'-- Beschriftung --
<1E57>
3010 ' <0799>
3020 GOSUB 5380 <09EE>
3030 ' <07C1>
3040 '--- Zeichnen der Werte --- <21AB>
3050 ' <07E9>
3060 GOSUB 5910:'Wertzuweisung <19D0>
3070 CLS:DEG <08FD>
3080 z=x-10 <11CF>
3090 g=z/2-23 <1301>
3100 b=4 <0BA8>
3110 IF anzahl>4 THEN g=g+16 <1E62>
3120 FOR i=1 TO anzahl:'-- Schleif
e fuer jeden einzelnen Wert --
<4524>
3130 IF schrift=2 THEN b=1 <1941>
3140 MOVE a,55:DRAW a,(d*wert(i)/m
a)+c <3303>
3150 ORIGIN a,(d*wert(i)/ma)+c <2AE5>
3160 MOVE g,z/6 <118B>
3170 FOR f=0 TO 365 STEP 10:'-- Sc
hleife fuer obere Ellipse -- <3BCE>
3180 DRAW SIN(f)*z/2+z/2,COS(f)*z/
6 <2CB7>
3190 NEXT f <0A5C>
3200 ORIGIN a,55 <0D0E>
3210 MOVE z,0 <0C8F>
3220 FOR k=90 TO 270 STEP 10:'--Sc
hleife fuer untere Ellipse -- <3CBC>
3230 DRAW SIN(k)*z/2+z/2,COS(k)*z/
6 <2C39>
3240 NEXT k <0AD4>
3250 ORIGIN 0,0 <09F8>
3260 MOVE a+z,(d*wert(i)/ma)+c:DRA
W a+z,55 <3D30>
3270 IF beschrift$="n"THEN GOTO 33
10 <1B91>
3280 MOVE a+b,(d*wert(i)/ma+c)+50 <3294>
3290 PRINT wert(i); <14CF>
3300 IF wert(i)<ma/100 THEN GOTO 3
320 <2157>
3310 MOVE a+(SIN(180)*z/2+z/2),(CO
S(180)*z/6)+57:FILL 3 <3752>
3320 a=a+z+10 <16E5>
3330 NEXT i <0A82>
3340 GOTO 5700:'-- Beschriftung --
<1E01>
3350 ' <0744>
3360 GOSUB 5380:'-- Werte holen --
<1DB0>
3370 t=0 <0BD6>
3380 ' <0780>
3390 '***** <22F6>
3400 '*Summe der Werte berechnen* <222A>
3410 '***** <221D>
3420 ' <07D0>
3430 FOR c=1 TO anzahl <16FD>
3440 t=t+wert(c) <1C37>
3450 NEXT c <0A59>
3460 ' <071F>
3470 '***** <2295>
3480 '* Umrechnungsformel * <2275>
3490 '***** <22BD>
3500 ' <076F>
3510 DIM x(anzahl) <15A3>

```

```

3520 FOR i=1 TO anzahl <1632>
3530 x(i)=(wert(i)*360)/t <284C>
3540 NEXT <0677>
3550 CLS <06F4>
3560 ' <07E7>
3570 '***** <225E>
3580 '* Zeichnen eines Kreises * <226E>
3590 '***** <2287>
3600 ' <0739>
3610 a=150 <0C7D>
3620 b=150 <0CB1>
3630 DEG <06B0>
3640 ORIGIN 300,200 <0C1C>
3650 MOVE 0,b <0C3A>
3660 FOR i=0 TO 360 STEP 10 <1398>
3670 DRAW a*SIN(i),b*COS(i) <21D1>
3680 NEXT i <0A3E>
3690 ORIGIN 300,180 <0C30>
3700 MOVE 150,0 <0AB6>
3710 FOR i=90 TO 270 STEP 10 <1439>
3720 DRAW a*SIN(i),b*COS(i) <2134>
3730 NEXT i <0AA2>
3740 ORIGIN 0,1 <09D2>
3750 ' <0764>
3760 '***** <22DA>
3770 '* Kreissegmente zeichnen * <22D4>
3780 '***** <2202>
3790 ' <07B4>
3800 TAG:a=a-4 <1218>
3810 MOVE 300,200 <0C23>
3820 DRAW 300,350 <0D8D>
3830 MOVE 300,200 <0C4B>
3840 s=0 <0B74>
3850 g=1.3:h=1.2 <1C10>
3860 FOR i=1 TO anzahl <16DC>
3870 xx=300:yy=200 <1764>
3880 MOVE xx,yy <11F7>
3890 DRAW 300+SIN(x(i)+s)*a,200+CO
S(x(i)+s)*b <3E1C>
3900 IF 200+COS(x(i)+s)*b<170 THEN
DRAW 300+SIN(x(i)+s)*a,200+COS(x(
i)+s)*b-18 <6163>
3910 IF 300+SIN(x(i)+s)*a<300 THEN
g=1.65 <32BF>
3920 IF 300+SIN(x(i)+s)*a<300 AND
schrift=1 THEN g=1.95 <3FB3>
3930 IF 300+SIN(x(i)+s)*a<300 AND
schrift=0 THEN g=2.6 <3F6E>
3940 IF beschrift$="n"THEN GOTO 39
70 <1B82>
3950 IF schrift=0 THEN xx=320 <1C72>
3960 MOVE xx+(SIN((x(i)/2)+s)*a)*g
,yy+(COS((x(i)/2)+s)*b)*h:PRINT we
rt(i); <698F>
3970 s=s+x(i) <19AB>
3980 NEXT i <0A97>
3990 TAGOFF <0681>
4000 GOTO 5700:'-- Beschriftung --
4010 ' <1E2B>
4020 GOSUB 5380:'-- Werte holen -- <076D>
4030 ' <1DD9>
4040 '***** <0795>
4050 '* Errechnung der Summe * <220B>
4060 '***** <2246>
4070 ' <2234>
4080 s=0 <07E5>
4090 FOR c=1 TO anzahl <0B54>
4100 s=s+wert(c) <1627>
4110 NEXT c <1C48>
4120 ' <0A85>
4130 '***** <074B>
4140 '* Umrechnung der Werte * <22C1>
4150 '***** <224E>
4160 ' <22E9>
4170 DIM x(anzahl) <079B>
4180 FOR i=1 TO anzahl <15CF>
4190 x(i)=(wert(i)*360)/s <165C>
4200 NEXT <2872>
4210 CLS <06A3>
4220 GOTO 4360 <061F>
4230 ' <0983>
4240 '***** <0726>
4250 '* Zeichnen eines Kreises * <229C>
4260 '***** <22AC>
4270 ' <22C4>
4280 ORIGIN 300,200 <0776>
4290 MOVE a*SIN(x(1)),b*COS(x(1)) <0C1F>
4300 FOR i=x(1)TO 360 STEP 4 <2705>
4310 DRAW a*SIN(i),b*COS(i) <1836>
4320 NEXT <21D2>
4330 ORIGIN 0,1 <0692>
4340 GOTO 4580 <0971>
4350 ' <0951>
4360 '***** <0717>
4370 '* Kreissegmente zeichnen * <228E>
4380 '***** <2288>
4390 ' <22B6>
4400 a=150:b=150:h=1.2:g=1.8 <0768>
4410 DEG:TAG <2C21>
4420 MOVE 300,200 <08E5>
4430 DRAW 300,350 <0CEA>
4440 MOVE 300,200 <0D53>
4450 s=0 <0C11>
4460 FOR i=1 TO anzahl <0B39>
4470 MOVE 300,200 <168D>
4480 DRAW 300+SIN(x(i)+s)*a,200+CO
S(x(i)+s)*b <0C4D>
4490 xx=280:yy=200 <3EBA>
4500 IF schrift=0 THEN xx=250 <1707>
4510 IF i=1 THEN xx=330 <1B5A>
4520 IF beschrift$="n"THEN GOTO 45
40 <1689>
4530 MOVE xx+(SIN((x(i)/2)+s)*a)*g
,yy+(COS((x(i)/2)+s)*b)*h:PRINT we

```

LISTING

```

rt(i); <6905>
4540 s=s+x(i) <1922>
4550 NEXT i <0A0D>
4560 GOTO 4240 <0963>
4570 ' <07CF>
4580 '***** <2246>
4590 '* Einzelsegment zeichnen * <2236>
4600 '***** <226E>
4610 ' <0721>
4620 MOVE 300+SIN(x(1)/2)*35,200+C
OS(x(1)/2)*35 <2EB2>
4630 DRAW 300+SIN(x(1)/2)*35,350+C
OS(x(1)/2)*35 <2F3F>
4640 ORIGIN 300+SIN(x(1)/2)*35,200
+COS(x(1)/2)*35 <2E36>
4650 MOVE 0,118+COS(x(1)/2)*35 <1B0A>
4660 FOR i=0 TO x(1) <1461>
4670 DRAW a*SIN(i),b*COS(i) <21A5>
4680 NEXT <0665>
4690 DRAW 0,0 <09C3>
4700 ORIGIN 0,1 <0956>
4710 GOTO 5700:'-- Beschriftung --
<1EBB>
4720 ' <07FD>
4730 ' <0711>
4740 GOSUB 5380:'-- Werte holen --
<1D7C>
4750 ' <0738>
4760 '-- Zeichnen der Werte -- <1F97>
4770 ' <0760>
4780 GOSUB 5910:'Wertzuweisung <1947>
4790 CLS <06A9>
4800 x=608/anzahl <1705>
4810 b=x/10 <1119>
4820 f=b+10 <1152>
4830 g=1 <0B76>
4840 IF schrift=1 THEN g=2 <19F4>
4850 IF schrift=0 THEN g=4 <190E>
4860 FOR i=1 TO anzahl <16AF>
4870 MOVE a+f,55 <1209>
4880 DRAW a+f,(d*wert(i)/ma)+c <2F34>
4890 DRAW a+x,(d*wert(i)/ma)+c <2F36>
4900 MOVE a+f,(d*wert(i)/ma)+c <2F93>
4910 DRAW a+b+f,((d*wert(i)/ma)+c)
+b:DRAW a+x+b,((d*wert(i)/ma)+c)+b
<72AC>
4920 DRAW a+x,(d*wert(i)/ma)+c <2F72>
4930 DRAW a+x,55 <122E>
4940 MOVE a+x+b,((d*wert(i)/ma)+c)
+b <3BA1>
4950 DRAW a+x+b,55+b <1C11>
4960 DRAW a+x,55 <126A>
4970 DRAW a+f,55 <1248>
4980 IF beschrift$="n"THEN GOTO 50
00 <1BE6>
4990 MOVE a+b+g,((d*wert(i)/ma)+c)
+b+20:PRINT wert(i); <4EFA>
5000 a=a+x <1351>
5010 NEXT i <0AA7>
5020 GOTO 5700:'-- Beschriftung --
<1E27>
5030 ' <0769>
5040 '***** <22DF>
5050 '* Kurvengrafik * <2261>
5060 '***** <2207>
5070 ' <07B9>
5080 GOSUB 5380:'-- Werte holen --
<1D26>
5090 ' <07E1>
5100 'Zeichnen der Koordinaten <1F2A>
5110 ' <0709>
5120 CLS:MOVE 50,55 <0D76>
5130 DRAW 640,55 <0C52>
5140 MOVE 50,55 <0B64>
5150 DRAW 50,400 <0CA5>
5160 ' <076F>
5170 '-- Zeichnen der Werte -- <1FCC>
5180 ' <0797>
5190 GOSUB 5910:'Wertzuweisung <197C>
5200 x=588/anzahl <1776>
5210 a=50 <0C70>
5220 FOR i=1 TO anzahl <1680>
5230 MOVE a,(d*wert(i)/ma)+c <2AAA>
5240 IF i=anzahl THEN GOTO 5330 <1963>
5250 DRAW a+x,(d*wert(i+1)/ma)+c <31E3>
5260 IF beschrift$="n"THEN GOTO 52
80 <1BDA>
5270 MOVE a+1,(d*wert(i)/ma)+c+20:
PRINT wert(i); <3FDD>
5280 PLOT a,53:PLOT a,52:PLOT a,51
:PLOT a,56 <2868>
5290 PLOT 48,(d*wert(i)/ma)+c:PLOT
47,(d*wert(i)/ma)+c:PLOT 46,(d*we
rt(i)/ma)+c <70A9>
5300 IF i=anzahl THEN GOTO 5700 <196D>
5310 a=a+x <13BE>
5320 NEXT i <0A15>
5330 IF beschrift$="n"THEN GOTO 53
50 <1B91>
5340 MOVE a+1,(d*wert(i)/ma)+c+20:
PRINT wert(i); <3F68>
5350 PLOT a,53:PLOT a,52:PLOT a,51
:PLOT a,56 <28F4>
5360 PLOT 48,(d*wert(i)/ma)+c:PLOT
47,(d*wert(i)/ma)+c:PLOT 46,(d*we
rt(i)/ma)+c <7034>
5370 GOTO 5700:'-- Beschriftung --
<1EE4>
5380 ' <0728>
5390 '***** <229E>
5400 '*Eingabe Anzahl der Werte * <2280>
5410 '***** <22C6>
5420 ' <0778>
5430 ON ERROR GOTO 5460 <0BA6>
5440 ERASE wert <0D82>
5450 IF b=6 OR b=7 THEN ERASE x <1964>

```

```

5460 PRINT"automatische":PRINT <166C>
5470 PRINT"Beschriftung ?":PRINT:P
RINT <1AAE>
5480 PRINT"nein = n ,":PRINT <1403>
5490 PRINT"sonst >ENTER<" <1617>
5500 PRINT:PRINT:INPUT;beschrift$:
CLS <1982>
5510 INPUT"Anzahl Werte: ";anzahl <1F55>
5520 PRINT STRING$(15, "_");:PRINT <139A>
5530 IF anzahl<1 THEN GOTO 5510 <16DE>
5540 DIM wert(anzahl+1) <1A1C>
5550 FOR i=1 TO anzahl <1616>
5560 PRINT <0683>
5570 ' <07A3>
5580 '-- Eingabe der Werte -- <1E7D>
5590 ' <07CB>
5600 PRINT i;:INPUT". Wert : ";wer
t(i) <2658>
5610 NEXT <06AB>
5620 ' <0707>
5630 'Berechnung des Maximalwertes
<2376>
5640 ' <0731>
5650 ma=0 <0CB2>
5660 FOR i=1 TO anzahl:ma=MAX(ma,w
ert(i)) <3450>
5670 NEXT i <0AD3>
5680 RETURN <069D>
5690 ' <0795>
5700 '***** <220B>
5710 '* Beschriftung der Grafik * <22BF>
5720 '***** <2232>
5730 ' <07E5>
5740 CURSOR 1:TAGOFF <0967>
5750 a$=INKEY$ <0C34>
5760 IF a$="^"THEN GOTO 5850 <137C>
5770 PRINT a$; <0BC3>
5780 GOTO 5750 <0937>
5790 ' <075C>
5800 '***** <22D2>
5810 '* neu mit >ENTER< drucken * <22DC>
5820 '* mit >d< Abspeichern >s< * <22E3>
5830 '***** <220E>
5840 ' <07C0>
5850 CURSOR 0 <0710>
5860 a$=INKEY$:IF a$=""THEN 5860 <19E7>
5870 IF a$="d"THEN CALL &9000 <13AA>
5880 IF a$="s"THEN GOTO 1170 <134F>
5890 CLS <0645>
5900 GOTO 350 <0913>
5910 'Grundlinie/Hoehe/Wertbreite <2214>
5920 TAG <069A>
5930 a=0 <0BAF>
5940 c=55:'Hoehe der Grundlinie <2390>
5950 d=285:'Gesamthoehe <1BB9>
5960 x=638/anzahl:'Wertbreite <2449>
5970 RETURN <06E2>
5980 END <0632>

```

GESUCHT: ANLEITUNG ZU "PUZZLE" UND LISTING ZU "SCHIFFE VERSENKEN"

Durch einen simplen Irrtum fehlte in Heft 12/87 das Listing zur abgedruckten Spielbeschreibung des Schulklassikers „Schiffe versenken“. Außerdem wurde das Programm „Puzzle“ veröffentlicht, ohne daß eine Anleitung zu finden war. Die Seeschlacht für den CPC werden wir so bald als möglich nachliefern, zum Bilderlegen hier eine Kurzanleitung:

Das Listing 1 ist an sich lauffähig. Also bitte abtippen, speichern und dann starten. Das Listing 2 bitte zum Listing 1 „hinzumergen“ und dann erneut abspeichern. Genauso mit dem Listing 3 verfahren. Der Sinn des Spieles besteht darin, die Puzzle-Teile durch Tauschen und Drehen so zu legen, daß die an den Grenzen der Einzelteile abgebildeten Symbole zu den Symbolen der anderen Teile passen. Zum Tauschen wählen Sie mit den Cursor-Tasten einen Teil aus und markieren ihn mit Copy. Dann verfahren Sie mit dem zweiten Teil ebenso. Die Puzzles werden dann getauscht. Zum Drehen fahren Sie den Teil wieder mit den Cursortasten an und können ihn dann mit der „R“-Taste rechts-, und mit der „L“-Taste links-herum drehen.

HARDCOPY EINBINDEN

Ich würde gerne die Superhardcopy von 2/87 in die Listings „Ostfriesisches Abitur“ und „Bio-

master“ einbinden, komme aber nicht klar. Beim Laden von SH-Copy muß ich erst über das Inhaltsverzeichnis gehen und bekomme dann auch die Matrixveränderung mit drauf. Wäre es Ihnen möglich, bei den mitgelieferten Listingausschnitten die Befehle einzufügen? Für Ihre Bemühungen „Thanks“.

Fritz Warth,
6800 Mannheim 31

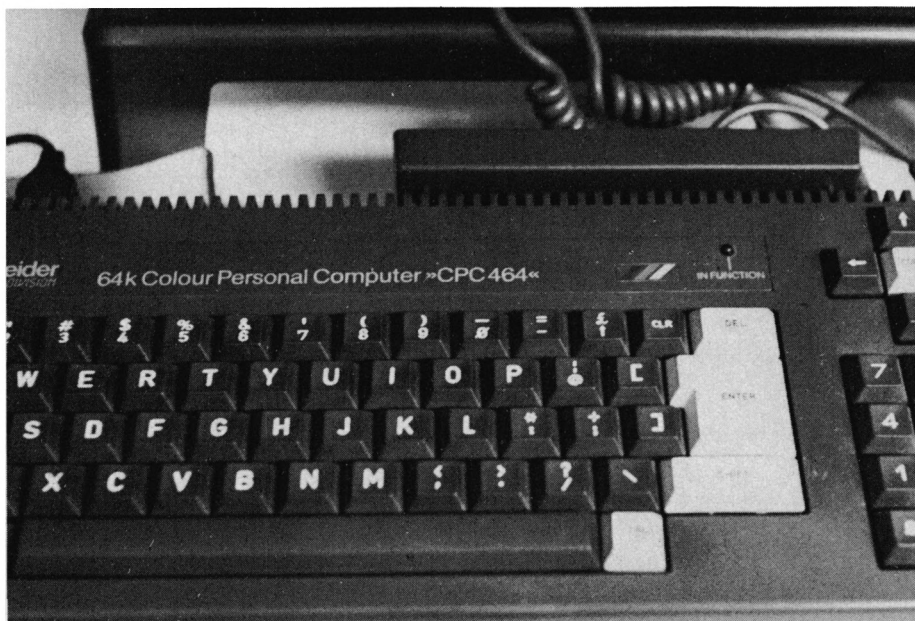
Beim Programm „Biomaster“ ist dies relativ einfach. Zuerst müssen Sie das Hardcopy-Programm laufen lassen (ab Zeile 3000 genügt) und dann mit NEW den Basic-Teil löschen. Nun haben Sie eine Reihe von Basic-Befehlen zur Verfügung. Unter anderem auch den Befehl „/HARDCOPY“. Wenn Sie nun in den Biomaster laufen lassen, wird in Zeile 475 dieser RSX-Befehl aufgerufen und die Hardcopy wird ausgegeben.

Beim Ostfriesenrechner wird vom Programm vorausgesetzt, daß die Hardcopy-Routine als Binärfile vorliegt (Zeile 3870) und wird dann in der Zeile 4580 durch den CALL-Befehl aufgerufen. Wollen Sie jedoch unsere Hardcopy-Routine verwenden, dann müssen Sie dafür Anpassungen vornehmen. Bitte entfernen Sie in Zeile 3860 und 3870 die REM-Bemerkungen nicht und ersetzen Sie in Zeile 4580 den CALL-Befehl durch „...THEN /HARDCOPY“. Speichern Sie neu ab und starten Sie dann das Hardcopy-Programm. Danach löschen Sie mit „NEW“ und lassen den Ostfriesenrechner ablaufen. Er kann nun auf den RSX-Befehl zurückgreifen.

Basic ist ganz einfach

Wie sag ich's meinem Programm

In diesem Heft nun die Fortsetzung unseres Basic-Einsteigerkurses. Diesmal geht es um Eingaberoutinen und Tastaturabfragen, die wohl bei keinem Programm fehlen. Da aber immer wieder Unklarheiten bestehen, wie solche Probleme am besten gelöst werden, wollen wir uns damit einmal etwas intensiver beschäftigen.



In der Programmküche dreht sich heute alles um die Tastatur

KEIN PROGRAMM OHNE EINGABEN!

Ihre Reaktion auf unsere neue Serie für Einsteiger im letzten Heft war groß. Größer, als wir erwartet haben. Überwiegend erreichten uns positive Antworten. Die wenigen kritischen Stimmen betonten immer wieder, daß diese Serie für sie nicht neu sei. Also, nochmals zur Klarstellung: Diese Serie ist für Einsteiger gedacht, die ihren CPC erst kurz besitzen und nach der Phase des Spielens nun selbst einmal ein Programm entwickeln wollen, mit den Erläuterungen des Handbuchs von Schneider aber nicht zufrieden sind, weil einige Probleme einfach zu knapp abgehandelt werden.

An diesem Punkt wollen wir einsteigen und bestimmte Probleme im Zusammenhang und ausführlich behandeln. Wie Sie dazu aus der Einleitung entnehmen können, geht

es diesmal um Eingaben und Tastaturabfragen.

Welches Programm kommt wohl um Eingabe und Tastaturabfragen herum? Wohl keins, denn irgendwie müssen Sie ja Ihrem Programm klarmachen, was Sie von ihm wollen!

Als erstes stellen wir uns einmal vor, Sie wollen von einem Menü aus, in dem Sie die Auswahl von 5 Menüpunkten vorgesehen haben, in die entsprechenden Unterprogramme verzweigen. Das Menü dazu soll so aussehen:

```

10 mode 2
20 print „Sie haben die Auswahl
zwischen 5 Menüpunkten“
30 print
40 print „Menüpunkt 1“
50 print „Menüpunkt 2“
60 print „Menüpunkt 3“
70 print „Menüpunkt 4“
80 print „Menüpunkt 5“
90 print
100 print „Ihre Wahl?“
    
```

Soweit das Menü, das dem Benutzer vorgeblendet wird. Jetzt müssen Sie vom Programm her auswerten können, welches Unterprogramm gewählt werden soll.

SIE HABEN NICHT NUR EINE MÖGLICHKEIT

1. Möglichkeit: Sie sehen eine if..then-Bedingung vor. Die entsprechenden weiteren Schritte müßten dann also so aussehen:

```

110 input a
120 if a=1 then goto 1000
130 if a=2 then goto 2000
140 if a=3 then goto 3000
150 if a=4 then goto 4000
160 if a=5 then goto 5000
170 print „Falsche Eingabe“: for
i=1 to 1000: next i: goto 10
1000 print „Jetzt bin ich im Unter-
programm 2“:end
3000 print „Jetzt bin ich im Unter-
programm 3“:end
4000 print „Jetzt bin ich im Unter-
programm 4“:end
5000 print „Jetzt bin ich im Unter-
programm 5“:end
    
```

Nun wollen wir uns das Programm einmal genauer ansehen: In Zeile 110 erwartet Ihr CPC eine Eingabe. Und zwar die Eingabe einer Ziffer, die durch Drücken der Enter-Taste abgeschlossen werden muß. Die nachfolgenden Programmzeilen (120 bis 160) verzweigen jetzt in das entsprechende Unterprogramm, das wir der Einfachheit halber hier nur mit der Bildschirmmeldung „Jetzt bin ich im Unterprogramm“ durch den Endbefehl des Programms abgebrochen haben. In diesen Zeilen müßte dann das von Ihnen geschriebene Unterprogramm ablaufen. Die Zeile 170 ist nur für eine erneute Abfrage vorgesehen, falls andere Zahlen als 1 bis 5 eingegeben worden sind.

DIE EINGABE MUSS MIT DEM VARIABLENTYP ÜBEREINSTIMMEN

Falls nun aber vom Benutzer des Programms keine Zahl sondern ein Buchstabe (ein String), gleichgültig,

ob versehentlich oder absichtlich, eingegeben wurde, bricht das Programm mit der Fehlermeldung „?Redo from start“ ab.

Sie wissen nicht, warum? Dann wollen wir versuchen, es Ihnen klarzumachen. Durch die Angabe der Variablen „a“ hinter dem Inputbefehl signalisieren wir dem Computer, daß als Eingabe eine Zahl erwartet wird. Wollen wir jedoch auch Buchstaben für die Eingabe zulassen, so müssen wir auch dem CPC dies mitteilen, indem wir als Variable nicht a vorsehen, sondern a\$. Dann allerdings wird die Variable als String angesehen und muß entsprechend weiterverarbeitet werden. Die Zeile 120 müßte dann so aussehen: 120 if a\$="1" then...

Sehen Sie den Unterschied? Wenn eine Zahl erwartet wird, reicht als Bedingung der Vergleich mit dem absoluten Wert. Bei einer Stringvariablen muß mit dem „Inhalt“ verglichen werden, daher die „1“ in Anführungszeichen.

TASTATURABFRAGEN LEICHT GEMACHT

2. Möglichkeit: Sie löschen die Zeilen 110 bis 160 wieder, indem Sie im Direktmodus eingeben: delete 110-160. Jetzt bitte die nachfolgenden neuen Zeilen eingeben, wobei wir unterstellen, daß das Restprogramm noch im Speicher ist.

```
110 call&bb06
120 if inkey(66)=0 then goto 1000
130 if inkey(64)=0 then goto 2000
140 if inkey(65)=0 then goto 3000
150 if inkey(57)=0 then goto 4000
160 if inkey(56)=0 then goto 5000
```

In der Zeile 110 steht für reine Basicprogrammierer ein etwas ungewöhnlicher, aber sehr nützlicher Befehl: Durch die Anweisung call &bb06 bleibt der CPC solange stehen und tut für die Programmausführung nichts, solange keine Taste gedrückt wird. Wenn aber dann eine Taste gedrückt wird, arbeitet das Programm weiter, indem der nächste im Listing stehende Befehl abgearbeitet wird. Diesen Befehl call&bb06 wollen wir an dieser Stelle nicht weiter erläutern, dazu

verfolgen Sie besser unseren Maschinensprache-Kurs.

In unserem Fall wird dann in den Zeilen 120 bis 160 geprüft, welche Taste gedrückt wurde. Vergleichen Sie bitte dazu Ihr Handbuch und sehen sich die Seite an, auf der die Tastennummern Ihres CPC's erläutert werden. Die Taste (inkey) Nr. 66 ist die Taste oben links neben der ESC-Taste, eben die Taste mit der Eins darauf. Die Taste Nr. 64 ist die Taste mit der Zwei usw.

FÜNFZIG MAL IN EINER SEKUNDE

Die Tastatur wird intern 50 mal in der Sekunde abgefragt, welchen Zustand die einzelnen Tasten haben, ob also eine Taste gedrückt wurde oder nicht. Mit dem inkey-Befehl haben Sie die Möglichkeit, diese Information in einem BASIC-Programm abzufragen. Wenn eine Taste nicht gedrückt wurde, liefert der inkey-Befehl bei der entsprechenden Tastennummer den Wert -1. Wurde die abgefragte Taste jedoch gedrückt, wird als Ergebnis der inkey-Abfrage der Wert 0 ausgegeben.

Da Sie jedoch noch andere Möglichkeiten haben, Ihre Tasten zu „bearbeiten“, werden entsprechend den gedrückten Tastenkombinationen (zusammen mit „Shift“, zusammen mit „CTRL“, zusammen mit „Shift“ und „CTRL“) verschiedene Werte bei der inkey-Abfrage als Ergebnis vom Computer geliefert. Einzelheiten entnehmen Sie dazu bitte der nachfolgenden Tabelle:

Wert, der geliefert wird	Shift gedrückt	CTRL gedrückt	Taste(Nr.) gedrückt
-1	--	--	nein
0	nein	ja	nein
32	ja	nein	ja
128	nein	ja	ja
160	ja	ja	ja

Nochmals zur Erläuterung: Wenn Sie also wissen wollen, ob die Taste „Eins“ in der oberen Zahlenreihe gedrückt worden ist – und zwar allein – oder zusammen mit Shift oder CTRL – heißt die jeweilige Abfrage:

```
if inkey(64)=0 then ... (Taste allein?)
if inkey(64)=32 then ... (mit Shift?)
if inkey(64)=128 then... (mit CTRL?)
if inkey(64)=160 then... (mit Shift und CTRL?)
```

Alles klar? So können Sie also leicht feststellen, welche Taste gedrückt wurde und in Ihrem Programm entsprechend weiter auswerten.

GEFRAGT IST DER INHALT

Eine andere Möglichkeit ist die Abfrage mit inkey\$. Bei diesem Befehl wird nicht der Zustand der Taste abgefragt, sondern der „Inhalt“. Ein Beispiel:

```
10 mode 2
20 print „Wählen Sie Ja oder Nein
<j/n>“
30 jn$=inkey$
40 if jn$="j" then print „Sie haben die <j>-Taste gedrückt!“:end
50 if jn$="n" then print „Sie haben die <n>-Taste gedrückt!“:end
60 goto 30
```

Hier wird also gefragt, ob die j- oder die n-Taste gedrückt wurde. Wird irgendeine andere Taste gedrückt, so erfolgt durch den goto 30-Befehl in Zeile 60 der Rücksprung zur Abfrage in Zeile 30. Ebenso verfährt das Programm, wenn Sie keine Taste drücken, der Wert der Variablen jn\$ also leer ist. Und wenn Sie ein großes „J“ eingeben, ebenfalls; denn auch dann treffen ja die Bedingungen in den Zeilen 40 und 50 nicht zu.

WAHLWEISE GROSS- ODER KLEINSCHREIBWEISE

Wenn Sie nun aber wahlweise ein großes „J“ und ein kleines „j“ bzw. ein großes „N“ oder ein kleines „n“ zulassen wollen, dann können Sie die Zeile 30 folgendermaßen abändern:

```
30 jn$=lower$(inkey$)
```

Mit diesem lower\$-Befehl werden alle Tasteneingaben, ganz gleich, ob Sie Klein- oder Großbuchstaben eingegeben haben, umgesetzt in die kleine Schreibweise. Dies sollten Sie bei eigenen Programmen immer vor-

sehen, denn Sie können ja beim Anwender nicht voraussetzen, daß er weiß, ob Sie im Programm die Groß- oder Kleinschreibweise gewählt haben. Und Sie wollen ja sicher gute Programme schreiben, in denen Bedienungsfehler von vornherein ausgeschaltet sind, nicht wahr?

Statt des lower\$-Befehls, der die Eingaben in die Kleinschreibweise umwandelt, können Sie natürlich auch in Großbuchstaben umwandeln. Dann heißt der entsprechende Befehl:

```
30 jn$=upper$(inkey$)
```

Damit werden also die Eingaben, ganz gleich, wie sie eingegeben werden, in Großbuchstaben umgewandelt. Es müssen dann natürlich die Zeilen 40 und 50 angepaßt werden, denn in diesen Zeilen muß ja nun auf Großbuchstaben abgefragt werden. Also:

```
40 if jn$=„J“ then ...
```

WAS SOLL DENN EINGEGEBEN WERDEN?

So, nun genug der Tastaturabfrage. Jetzt wollen wir uns nochmals mit dem Input-Befehl befassen. Damit der Anwender auch weiß, was bei einer Input-Abfrage von ihm erwartet wird, sollten Sie den dazugehörigen Text mit eingeben.

Also in etwa so:

```
10 mode 2
20 print „Wie alt sind Sie?“
30 input alt
40 ,weiter im Programm
```

Nun weiß der Benutzer genau Bescheid. Denn auf dem Schirm erscheint die Frage nach dem Alter, die dann einzugeben ist. Sie können dies allerdings auch noch eleganter lösen, in dem Sie den Abfragetext mit in den Input-Befehl einbeziehen:

```
10 mode 2
20 input „Wie alt sind Sie“;alt
```

Sehen Sie den Unterschied? Eine Zeile wurde gespart, und diese Lösung sieht doch wirklich besser aus. Mit diesem Beispiel haben Sie die direkte Zuordnung zwischen dem Abfragetext und der nachfolgenden Variablen. Ersetzen Sie nun einmal in der Zeile 20 das Semikolon

durch ein Komma und starten das Programm. Sie werden feststellen, daß bei einem Semikolon als Trennzeichen zwischen dem Text und der nachfolgenden Variablen ein Fragezeichen ausgegeben wird, bei einem Komma jedoch nicht. Welches Trennzeichen Sie verwenden wollen, ist Geschmackssache, auf den Programmablauf hat dies keine Auswirkungen.

Am Anfang dieses Artikels haben wir den Hinweis gebracht, daß bei der Eingabe eines Buchstabens (eines Strings), wenn eine Zahl vom Programm erwartet wird, die Fehlermeldung „?Redo from Start“ ausgegeben wird. Um diese Fehlermeldung auszuschließen, können Sie folgendes programmieren:

```
10 mode 2
20 Input „Wie alt sind Sie“;alt$:
alt=val(alt$):if alt<1 or alt>100
then goto 10
```

Nun wird ein kleiner Trick angewandt. Die Altersangabe wird nicht in einer Zahlenvariablen abgelegt, sondern in dem String alt\$. Anschließend wird das erste Zeichen dieses Strings durch den VAL-Befehl in den entsprechenden numerischen Wert umgewandelt. Ist das erste Zeichen des numerischen Strings (also alt\$) keine Zahl, sondern ein Buchstabe, so wird durch die VAL-Umwandlung der Wert der Zahlenvariablen Null. In der anschließenden If-Bedingung haben wir nun festgelegt, daß wieder zur Zeile 10 gesprungen werden soll, wenn alt kleiner 1 oder größer 100 ist.

ZWEI FLIEGEN MIT EINER KLAPPE

Damit haben Sie nun „zwei Fliegen mit einer Klappe geschlagen“:

1. Wenn Sie keine Zahl, sondern einen Buchstaben eingeben, bringt das Programm keine Fehlermeldung, sondern weist die Eingabe einfach zurück und fragt nochmals ab.

2. Wenn Sie eine Zahl eingeben, die kleiner als null oder größer als 100 ist, wird die Eingabe ebenfalls nicht angenommen.

MEHRERE VARIABLEN AUF EINMAL

Nun gibt es aber auch manchmal die Notwendigkeit, mit einem Input-Befehl mehr als eine Variable abzufragen. So zum Beispiel:

```
10 mode 2
20 input „Welchen Monat und welchen Tag haben wir heute?“;
mt$,tg
```

Sie sehen in diesem Beispiel, daß in der Zeile 20 zwei Variablen stehen, nämlich mt\$ (für den Monat) und tg für den Tag. Der CPC erwartet nun in diesem Fall zwei Eingaben, die durch ein Komma getrennt einzugeben sind. Also: Mai,12<Enter>.

Daraus können Sie entnehmen, daß eine Eingabe nur dann ein Komma enthalten darf, wenn der CPC zwei Eingaben erwartet. Was machen Sie nun aber, wenn Sie auf einen Input-Befehl hin einen Text eingeben wollen (oder müssen), der ein solches Komma enthält? In einem solchen Fall besteht nur eine Möglichkeit: Sie nehmen nicht den Input-Befehl, sondern verwenden Line input. Beispiel:

```
10 mode 2
20 line input „Welche Tage gibt es“;
a$
```

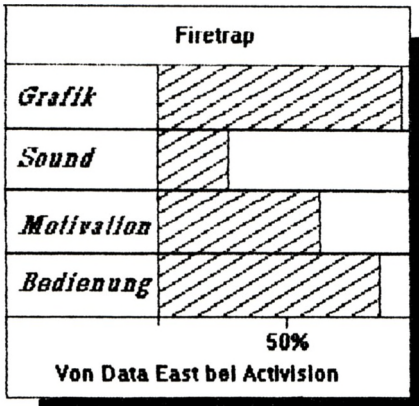
In diesem Fall akzeptiert der Rechner auch eine Aufzählung, die jeweils mit Kommata getrennt wurden. Der String a\$ enthält dann alle Eingaben, auch wenn zwischen durch ein Komma eingegeben wurde. Die Kommata sind Bestandteil des Strings a\$.

Programmier-Küche



Der CPC brennt!

Ein Alptraum vieler Menschen ist es, in einem brennenden Hochhaus von den Flammen eingeschlossen zu werden. Ob die Hilfe noch rechtzeitig kommt? Beeilen Sie sich, denn der Retter sind Sie!



In einem Hochhaus einer amerikanischen Großstadt schlagen in vielen Stockwerken die Flammen aus den Fenstern. Überall ist Feuer und Rauch. Durch die verqualmten Treppenhäuser ist keine Flucht möglich. Die Fahrstühle funktionieren nicht mehr und die Feuerwehr hat keine Leitern, die so lang sind, daß die von den Flammen eingeschlossene Frau im obersten Stockwerk noch gerettet werden könnte.

WO BLEIBT DIE RETTUNG?

Muß sie verbrennen oder elendiglich ersticken? Keiner weiß es, keiner kann Hilfe bringen. Wirklich keiner?

Doch zum Glück sind Sie ja da. Als ausgebildeter Fassadenkletterer sehen Sie nur eine Chance, der Eingeschlossenen zu Hilfe zu eilen. Sie müssen an der Fassade emporklettern und versuchen, die Arme zu retten. Ob Sie es schaffen?

Zweifel nagten an Ihrer Seele. Sollen Sie es wirklich riskieren? Doch bald haben Sie Ihre eigene Angst überwinden und treten die riskante Klettertour an.

An dieser Stelle beginnt das Spiel „Firetrap“. Sie müssen nun versuchen, sich an einer Hausecke von Stockwerk zu Stockwerk nach oben zu mühen. Ganz entfernt ruft immer wieder die Eingeschlossene um

Hilfe. Schon ist sie von den Flammen umhüllt. Kommen Sie noch rechtzeitig?

GEFRAGT SIND FASSADENKLETTERER

Bei Ihren Kletterkünsten an der Hausfassade müssen Sie auf sich selbst achten. Denn plötzlich schlagen aus einem Fenster, an dem Sie gerade vorbeiklettern wollen, die Flammen heraus. Da heißt es, geschickt sein und rechtzeitig ausweichen.

Doch das ist nicht die einzige Gefahr, die Ihnen droht. Es fallen auch allerlei Gegenstände herab und wenn Sie nicht aufpassen, werden Sie von ihnen erschlagen.

Doch zum Glück haben Sie zufällig eine Strahlenpistole für Fassadenkletterer dabei. Mit ihr können Sie die Ihnen von oben entgegenkommenden Trümmer zerstrahlen, also einfach in Staub auflösen. Wie praktisch das ist! Fast wie im richtigen Leben.

Aber Spaß beiseite, denn Ihre Lage wird immer bedrohlicher. Immer heftiger wütet das Feuer, immer lauter ruft die Frau im obersten Stockwerk um Hilfe, immer mehr Gegenstände fallen von oben auf Sie herab.

ZURÜCK ZUR ERDE IM SCHWEBEFLUG

Sind Sie geschickt, weichen allen Hindernissen aus und erreichen unversehrt das oberste Stockwerk, dann winkt Ihnen eine zärtliche Belohnung: Die völlig fremde Frau sinkt Ihnen in die Arme und beide schweben Sie wieder zurück zur Erde.

Werden Sie jedoch von den Flammen erfaßt oder von den herunterfallenden Gegenständen erschlagen, dann haben Sie noch einen Versuch.

DIE RETTUNG DAUERT 30 SEKUNDEN!

Beim Testen dieses Spieles haben wir etwa 30 Sekunden gebraucht, um durch geschickte Joystick-Bewegungen den Fassadenkletterer bis in das oberste Stockwerk zu dirigieren. Dann ist das Spiel aus und alles kann wieder von vorne beginnen.

Der Preis für dieses Game (DM 44,95) steht in keinem Verhältnis zu seinen kläglichen 30 Sekunden Spieldauer. Manchmal muß man sich schon fragen, ob die Softwarehäuser kein schlechtes Gewissen dabei haben, den meist jugendlichen Käufern das Geld aus der Tasche zu ziehen. Für ein gutes Spiel mit immer wieder anderen Situationen und abwechslungsreichen Screens zahlt man ja diesen Preis gern. Aber für 30 Sekunden??



GEHEN SIE LIEBER INS KINO

Das Game wird mit dem Joystick gespielt. Durch Druck auf den Feuerknopf wird Ihre Strahlenpistole ausgelöst. Die Grafik ist eher durchschnittlich. Nicht schlecht, aber da nur ein Screen, der zwar scrollt, vorhanden ist, wird's auf die Dauer etwas eintönig. Sound ist überhaupt nicht vorhanden und die Spielmotivation ließ nach fünf Durchgängen schon erheblich nach.

Sparen Sie sich das Geld für dieses Spiel und gehen Sie dafür ins Kino. Sie haben mehr davon! JE

Firetrap:
Data East by Activision

Motivation: 20 %

Grafik: 40 %

Sound: 0 %

Handhabung: 90 %

Der Turm zu Babylon

Programmiersprachen für den CPC

Die Zeiten, als ein Homecomputer nur BASIC verstand, sind schon lange vorbei. Alleine hierzu gibt es mehrere Compiler, die – mit kleinen Nachteilen – Tempo mit ins Spiel bringen. Aber auch andere Programmiersprachen sind mittlerweile verfügbar. Von C bis Turbo Pascal, von Fortran bis FORTH. So schön die große Auswahl auch ist, sie erfordert ein wenig Grundsatzinformationen. Wo der Unterschied liegt und was bereits vor dem Kauf zu beachten ist, können Sie jetzt in der CPC-WELT erfahren.

ASSEMBLER – DIE SCHNELLSTEN PROGRAMME

Bei den allerersten Homecomputern mit Z80-Prozessoren – man denke nur an den Sinclair ZX 81 – war die Rechengeschwindigkeit einfach zu langsam. Hobbyprogrammierer stellten schnell fest, daß es in der Regel am BASIC-Interpreter lag, der nicht rasch genug „übersetzte“. Der Wunsch nach einer maschinennahen Programmierung kam auf und viele, die schon Kenntnisse über den Prozessor hatten, konnten leistungsfähige Programme entwickeln. Assembler wurde die Programmiersprache schlechthin. Trotz schneller Interpreter und höherer Taktzahl des Prozessors ist der Wunsch geblieben, möglichst alles aus dem Rechner herauszuholen; und das geht nach wie vor nur mit der Maschinensprache. Wer sich noch nicht mit dem Thema befaßt hat, dem sei gesagt, daß auch Assembler eine Sprache ist, die der Rechner letztlich

nicht versteht. Auch hier gibt es Befehlswörter, sogenannte Mnemonics, meist Abkürzungen aus dem englischen Computersprachgebrauch. Hiermit werden Anweisungen für die einzelnen Register erstellt; es entsteht ein Quellcode, der auch kommentiert werden kann. Lauffähig wird der Text erst durch die „Assemblierung“, die Umwandlung in einen reinen Zahlencode.

Auch die Disassemblierung – das Auslesen eines Speicherinhaltes – ist möglich, den REM-Kommentar darf man dabei natürlich nicht erwarten, er steht nur im Quellcode. Die Vorteile der Maschinensprache liegen nicht alleine im Tempo. Beim CPC kann durch RSX-Kommandos der BASIC-Befehlssteil ergänzt werden. Gerade beim CPC 464 können so die fehlenden Anweisungen nachträglich eingefügt werden. Nachteile müssen bei der prozessornahen Programmierung allerdings auch in Kauf genommen werden: Der Assembler verbraucht un-

ter Umständen viel Speicherplatz (Ausnahme sind Software-Eproms), je nachdem, wie leistungsfähig das Programm ist. Das Erlernen der Maschinensprache fällt zudem recht schwer, dauert entsprechend lange und der Quellcode ist aufgrund der Mnemonics dann auch für Kenner recht unübersichtlich.

BASIC – PROGRAMMIERSPRACHE FÜR ALLE UND ALLES

Über den Beginners All Purpose Symbolic Instruction Code (Symbolische Allzwecksprache für Anfänger) denkt heute kaum noch jemand nach. Schade drum, denn erstens ist der CPC bereits mit einem hervorragenden Interpreter und umfangreichem Befehls-Sprachschatz ausgestattet, und zweitens gibt es zahlreiche Compiler, die ein BASIC-Programm in einen schnellen Assemblercode übersetzen können. Der Interpreter im CPC stammt aus dem englischen Locomotive-Softwarehaus. Bei der Entwicklung wurden gedankliche Anleihen bei MicroSoft genommen, die in der Programmierlogik einen BASIC-Standard schaffen. So hervorragend das eingebaute ROM auch ist, es weist Fehler auf, etwa die DEC\$-Anweisung beim CPC 464. Die Korrekturen und das Hinzufügen neuer Befehle (FILL, CIRCLE usw.) beim CPC 664/6128 sorgen leider für Inkompatibilität innerhalb der CPC-Reihe. Soweit es um die zahlreich angebotenen Interpreter mit Compiler geht, sind einige Dinge zu beachten: Zunächst verbrauchen diese

„Programme zur Programmierstellung“ Speicherplatz, das entstehende Listing kann also nicht so lang sein, wie es mit dem Locomotive-BASIC der Fall ist. Besonders unangenehm wird dies, wenn sich Compiler, Quellcode und assembliertes Programm den Platz teilen müssen.

Daneben muß aber auch an die Spezifikation der Compiler gedacht werden, denn Ideal-Compiler gibt es nicht. Teilweise sind sie zur Anfertigung von Spielen von Grafikroutinen („Taifun“) ausgelegt, dann fehlt zu meist die Möglichkeit, Realzahlen zu verarbeiten. Sehr ernsthafte Compiler (zum Beispiel „C-BASIC“) machen unter Umständen bei der Grafikprogrammierung Schwierigkeiten. Hier sollte jeder erst einmal darüber nachdenken, welches seine bevorzugten Programmenthemen sind, bevor er den Compiler kauft.

C – NEUE MODE ODER ZUKUNFTSTRÄCHTIGE SPRACHE

Die Rechner mit dem 68000er Prozessor haben sich mittlerweile auf breiter Front durchgesetzt. Und mit ihnen kam eine neue Programmiersprache, die mit einem einfachen „C“ benannt wurde. Aber wer C sagt, der muß auch B sagen, denn bei BCPL beginnt die Geschichte: Basic Combined Programming Language wurde in Cambridge entwickelt und sollte zur Erstellung von Systemsoftware (Betriebssystemen) dienen. Als jedoch das Betriebssystem UNIX auftauchte, arbeiteten die Entwickler auf der Grundlage von BCPL an einer Möglichkeit,

UNIX auf andere Rechner zu übertragen. Es wurde mehr daraus, nämlich das eigenständige C.

Dabei wird immer wieder von einer „höheren“, der menschlichen Kommunikation nahestehenden, Programmiersprache gesprochen und damit werden gegebenenfalls auch falsche Erwartungen geweckt. Gar so einfach ist es doch nicht, und einige Querverweise zu BASIC führen eher dazu, daß ein Quellcode unübersichtlich werden kann. Aber lassen Sie sich nicht abschrecken, Übung macht den Meister und die CPC-WELT will Ihnen ab dieser Ausgabe Gelegenheit dazu geben.

Doch davon einmal abgesehen, überwiegen die Vorteile. Bei C ist es gelungen, sehr maschinen-nah zu programmieren und doch mit relativ strenger Normung system-übergreifend zu bleiben. So widersprüchlich sich dies anhört, es ist ganz einfach: Hardwareabhängige Routinen, etwa die Bildschirmsteuerung, werden in Bibliotheks-routinen abgelegt, die nur für den speziellen Rechner gelten. Das eigentliche Programm greift dann darauf zurück.

Mit der bereits erwähnten Prozessornähe ergibt sich selbstverständlich auch eine Geschwindigkeitssteigerung gegenüber BASIC. Dies führte dazu, daß es bereits etliche Betriebssysteme gibt, die unter C erstellt wurden. Dabei handelt es sich natürlich auch wieder um die Methode Quellcode und Compiler – mit den entsprechenden Nachteilen (Speicherplatzbedarf und relativ umständliche Handhabung). Wer sich über diese Grundlagen hinaus informieren will, dem sei die Rubrik C-

Splitter der CPC-Welt (erstmal in dieser Ausgabe) empfohlen.

COMAL – DER KLARE INTERPRETER AUS DEM NORDEN

Aus Dänemark stammt die COMMon Algorithmic Language, welche hier nur der Vollständigkeit halber erwähnt werden soll. Diese Programmiersprache wurde zur Schulung auf PC's entwickelt und erfreut sich dank ihrer Einfachheit und klaren Struktur relativ großer Beliebtheit im Schulsektor. Der große Durchbruch gelang allerdings nicht. Hier konnte Pascal schon besser überzeugen.

PASCAL – VIELE NORMEN, VIELE UNTERSCHIEDE

Wer Pascal erwähnt, muß auf das Stichwort „Strukturierte Programmierung“ nicht lange warten. Wie keine andere Programmiersprache zwingt der von Professor Niklas Wirth erfundene Interpreter zur Ordnung. Jede Routine erhält einen eigenen Programmkopf, in dem auch die Variablen definiert werden müssen. Der compilierte Quellcode läuft zudem sehr maschinen-nah ab, so daß – gemessen an BASIC-Pascal in der Ausführung eine schnelle Programmiersprache ist.

Probleme gab es mit der Portabilität, also der Übertragung von einem Rechner auf den anderen. Ursprünglich war ein sehr begrenzter und daher auch leicht übertragbarer Befehlssatz vorgesehen. Was aber einmal ausreichen mochte, mag die Fähigkeiten moderner Rechner nicht mehr auszu-

reizen. Den ersten Versionen folgten andere, und heute schließlich muß sich der Interessent zwischen mehreren Dialekten entscheiden. Beim CPC gibt es neben dem HiSoft-Pascal auch das Turbo Pascal. Wer die Wahl hat, der sollte zu letzterem greifen. Hier ist eine gewisse Übertragbarkeit gewährleistet, da Turbo Pascal unter CP/M arbeitet.

FORMULA – DAS LIEBLINGSKIND DER MATHEMATIKER

Nur damit der Blick über die Programmiersprachen möglichst groß wird, sei hier auch FORMULA (FORMULary LAnguage) erwähnt. Speziell für die Bedürfnisse von Mathematikern (Formeleingabe!) entwickelt, erlangte dieser Interpreter keine große Verbreitung. Software für den CPC über dieses Thema ist nur über den Public Domain Vertrieb erhältlich.

FORTH – „POLNISCHES RECHNEN“ WIRD ERFORDERLICH

Wörtlich übersetzt, heißt forth auch „fortführend“, soweit es die Programmiersprache betrifft, werden allerdings erst einmal Umwege gemacht. Schuld daran trägt die umgekehrte Polnische Notation, von der mancher wahrscheinlich erst durch Computer erfahren hat. Dabei wird jeder Operand auf einen Stapelspeicher gelegt, der nach dem LIFO-Prinzip (Last In – First Out) arbeitet. Zwischenergebnisse sind Bestandteil dieses Stapels. Wer sich daran einmal ge-

wöhnt hat, der kommt mit FORTH ebenso gut (oder ebenso schlecht) zurecht wie mit jeder anderen Programmiersprache. Das Haupteinsatzgebiet ist bei kleineren Steuerprogrammen zu sehen. Es gibt jedoch auch Assembler, die unter Forth programmiert wurden, was für die Leistungsfähigkeit des Interpreters spricht.

LOGO – KAUM BEKANNT, OBWOHL FÜR CPC-BESITZER KOSTENLOS

Merkwürdig ist es schon: Da verfügen rund 300000 CPC-Besitzer neben BASIC über eine zweite Programmiersprache, doch davon ist selten etwas zu bemerken, kaum jemand befaßt sich damit. Vielleicht liegt es an der vorgegebenen Zielgruppe. Obwohl die Sprache durchaus leistungsfähig (eine der ersten Möglichkeiten, auch Prozeduren zu gestalten) ist, sollte sie dazu herhalten, Kindern den Umgang mit Computern zu erklären. Da gibt es zum Beispiel eine Turtle-Grafik, mit der sich arbeiten läßt. Die allerersten Versionen zeigten statt eines Kurses tatsächlich eine stilisierte Schildkröte, um die Kleinen bei Laune zu halten. Dies hatte zur Folge, daß kaum ein ernsthafter Programmierer Interesse an Logo bekam. Trotzdem lohnt es sich, diese Programmiersprache besser kennen zu lernen. Die Leser der CPC-WELT haben ab der nächsten Ausgabe Gelegenheit dazu. ■



TRANTOR

Die reinen Ballergames haben Hochkonjunktur. Und dies hat seinen guten Grund: Man braucht sich nicht um Taktik und Strategie zu kümmern, sondern nur wild um sich zu schießen. Das Richtige, um seinen Frust abzureagieren.

Trantor	
Grafik	██████████
Sound	██
Motivation	██████
Bedienung	██████████
50%	
Von Probe bei Activision	

Bei diesem neuen Spiel des englischen Softwarehauses „Probe Software“ können Sie nur bestehen, wenn Sie möglichst viel um sich schießen, denn Sie sind der letzte richtige Einzelkämpfer, also so eine Art CPC-Rambo.

Aber diesmal nicht mit einem Strahlengewehr oder einer Schnellfeuerpistole, sondern viel effektiver: In diesem Game sind Sie mit einem Flammenwerfer ausgerüstet, der alles, worauf Sie schießen, kurz aufglühen und vergehen läßt. Und was Sie zerstrahlt haben, kann Ihnen nicht mehr gefährlich werden!

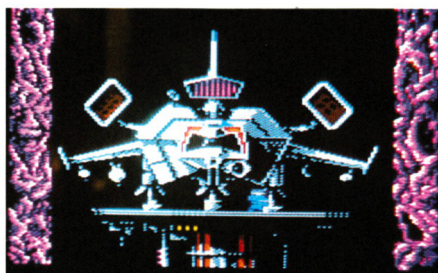
DAS RICHTIGE FÜR JOYSTICKAKROBATEN!

Also genau das Richtige für alle Ballerfreunde, die geschickt mit dem Joystick umgehen können und auf alles, was sich bewegt, feuern wollen.

Nach dem Start des Spiels sehen Sie zum Einstimmen zunächst in einem kurzen Vorspann, wie sich Ihr Raumschiff langsam der Landeplattform des feindlichen Planeten nähert. Während dieser Landephase hat es nicht viel Sinn, auf irgendwelche Tasten zu drücken, den Vorspann können Sie nicht abkürzen!

Geht es dann endlich richtig los, finden Sie sich in einem Gang wieder, in dem einige Dinge herumfliegen, die Ihnen bei Berührung einige „Gesundheitspunkte“ kosten.

Nichts wie abschießen, heißt also die Devise. Doch Sie müssen schon sparsam mit Ihrer Munition sein, denn so ein Flammenwerfer verbraucht schon einiges. Sonst sieht es nachher so aus, daß Sie zwar noch einigermaßen „gesund“ sind, sich aber nicht mehr wehren können, weil Ihre Waffe nicht mehr funktioniert. Und dies sollte Ihnen nach Möglichkeit nicht passieren!



SUCHEN SIE SICH DAS CODEWORT ZUSAMMEN

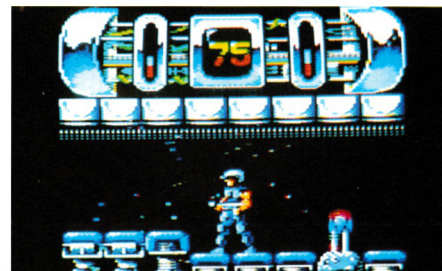
In diesen Gängen stehen nun acht Computer herum, von denen jeder einen Buchstaben enthält. Diese Buchstaben müssen Sie sich merken, in die richtige Reihenfolge bringen und in das Sicherheitsterminal eingeben.

Bei richtiger Eingabe wird der Beam-Code ausgegeben. Nun nichts wie hin zur Beam-Abteilung, dort den Code eingeben, und schon werden Sie nach erfolgreicher Mission transferiert und sind in Sicherheit!

Schaffen Sie es jedoch nicht, weil es mit Ihrer Gesundheit nicht zum besten steht, so lösen Sie sich plötzlich in einem Sternenhagel auf und müssen wieder ganz von vorne beginnen.

FAZIT:

Die Grafik bei diesem Spiel ist recht gut gelungen. Daran läßt sich wieder einmal sehen, wozu der CPC bei sorgfältiger Programmierung fähig ist. Auch die Objekte, die einem in dem Gang da entgegenfliegen, sind abwechslungsreich und zum Teil recht originell gestaltet. Der Sound beschränkt sich während des Spielens auf die Zisch-Geräusche des Flammenwerfers.



Der Titelsound sollte manchmal abgeschaltet werden können. Doch dies ist leider nicht der Fall!

Die Steuerung mit Joystick oder Tastatur ist sehr einfach und reagiert schnell. Sie werden sicher keine Schwierigkeiten haben, Ihr Männlein schnell durch die Gänge zu leiten und während des Laufes auch noch zu schießen. Die Tastenbelegung können Sie sich völlig frei selbst wählen.



Für Ballerfreunde ist die Motivation zum Spielen sicher sehr hoch, da die Gänge recht lang und abwechslungsreich gestaltet sind.

In Deutschland wird das Game von Ariolasoft vertrieben und kostet in der Diskettenversion DM 44,95. Für die Kassette müssen Sie DM 29,95 hinblättern. **JE**

Probe Software by Ariolasoft

Motivation: 75 %

Grafik: 90 %

Sound: 50 %

Handhabung: 100 %

**CLASSIX 1
ALTE HÜTE IN NEUER
VERPACKUNG**

Die Idee einer Sammelpackung ist eigentlich zu begrüßen: Jeder erhält die Chance, zeitlich weit zurückliegende Programme nachzukaufen, auch wenn er seinen CPC erst seit ein paar Tagen besitzt. Meist stimmt auch der Preis und der Kunde erhält wirklich einen realen Gegenwert für sein Geld. Was aber macht ein Softwarehaus, wenn die Spiele, die nie zu einem rechten Hit wurden, nun schon einige Monate alt sind und die Verkaufszahlen stark nachlassen? Läßt man diese Software dann sang- und klanglos „sterben?“ Weit gefehlt! Man bringt sie in neuer Verpackung und neuer Zusammenstellung als Paket auf den Markt in der Hoffnung, noch einmal kräftig verdienen zu können, bevor diese Programme dann endgültig „begraben“ werden! Daß es sich in diesem Fall nicht um die Klassiker, die ehemaligen Spitzenreiter der Software-Hitparaden handelt, merkt manch einer erst zu Hause. Im Laden lockt der vermeintlich günstige Preis.

Dies jedenfalls scheint sich „Ariolasoft“ gedacht zu haben, als im Herbst die Softwaresammlung „Classix 1“ auf den Markt geworfen wurde. Laut Umschlagbild sind darin fünf Spiele enthalten. Spielt der Käufer dann diese Spiele durch, stellt er recht bald fest, daß es nur drei sind; von zwei weiteren gibt's lediglich eine Demoversion. Diese sind zwar aus der Originalversion übernommen, enthalten aber nur jeweils ein Screen. Also lediglich ein „Appetit“-Happen, um dem Käufer den Mund wäßrig zu machen in der Hoffnung, doch noch die Originalversion unter die Leute zu bringen.

**BOBBY BEARING
BEFREIEN SIE DIE FAMILIE**

Das erste Spiel, das wir auf der uns zur Rezension überlassenen Diskette gespielt haben, war „Bobby Bearing“ vom Hersteller „The Edge“. Sie steuern damit eine Kugel, genannt Bobby, durch ein Labyrinth von Gängen und Korridoren, immer auf der Suche nach den Geschwistern von Bobby, die Sie nach Hau-

se bringen müssen.

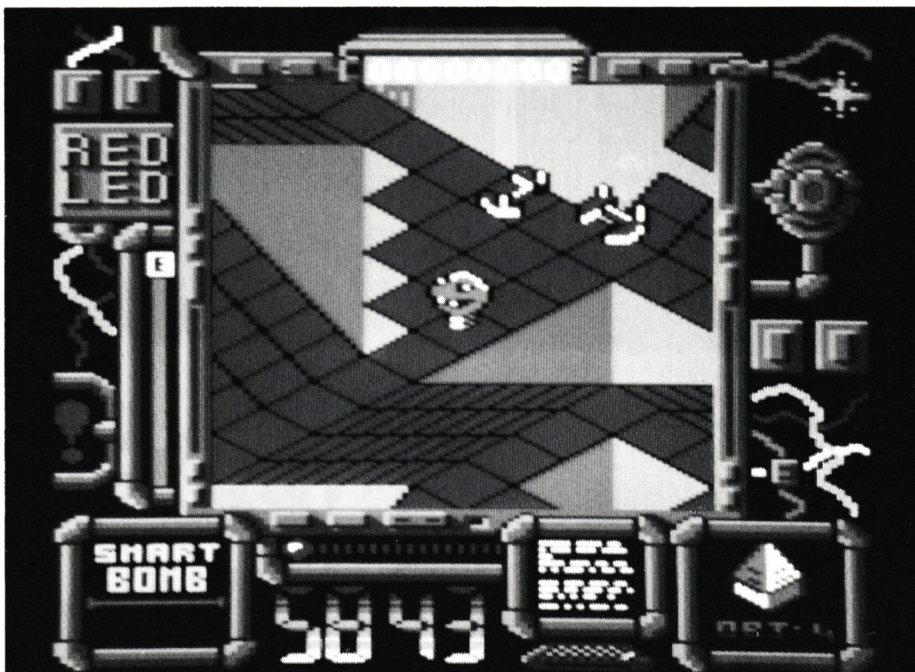
Dies ist leichter gesagt, als getan, denn überall lauern Fallen, um dem armen Bobby den Garaus zu machen. Mal wird er von einer Art Presse zermalmt (wenn er nicht aufgepaßt) und zeigt dann ein schmerzverzerrtes Gesicht. Oder Sie kommen mit Ihrem Bobby in eine Falle, aus der es kein Entrinnen mehr gibt.

Dieses Spiel führt in eine futuristische Welt aus schier endlos erscheinenden Gängen, die manchmal auch einen Irrweg darstellen und das ganze Bemühen von vornherein zum Scheitern verurteilen. Gesteuert wird Bobby mit Hilfe der Cursor-

keine besonderen Effekte. Die Grafik ist recht gut gelungen. Feinfühligste Steuerung zeichnet dieses Programm aus. Der Bildschirm wird von den einzelnen Screens allerdings nicht voll ausgenutzt. Manchmal erscheinen die Screens daher etwas mager.

**BRIAN BLOODAXE
EROBERN LEICHT GEMACHT**

Brian, der britische Eroberer, der mit der Blutaxt, wurde vor einigen Jahrhunderten von einem Schneesturm überrascht und in einem Eisblock tiefgefroren. Beim Auftauen



tasten oder dem Joystick. Zum Umschalten auf den Joystick drücken Sie die TAB-Taste und Cursor-hoch gleichzeitig. Nur im ersten Screen wirkt diese Umschaltung nicht, da muß auf jeden Fall mit den Cursortasten gelenkt werden.

Ab und zu lauert auch eine schwarze Kugel auf unseren Bobby, die dann umbarmherzig zuschlägt und den Bedauernswerten verschlingt. Um dieses Spiel erfolgreich zu spielen, müssen Sie sich sputen, denn die Zeit zum Befreien der Geschwister ist nicht unbegrenzt. Wird die eigene Kugel nun zerquetscht und gefressen, verlieren Sie erheblich an Zeit, die Ihnen dann zum Schluß fehlt.

Ein Spiel für Joystickakrobaten: kurzweilig, keine besonderen Schwierigkeiten, allerdings auch

findet er sich plötzlich im 20. Jahrhundert wieder und muß sich nun ungewohnten Gefahren aussetzen.

Dieses Spiel, begleitet von einem flotten Sound, ist recht nett zu spielen. Der Held dieses Spiels muß sich von Screen zu Screen allen möglichen und unmöglichen Gefahren aussetzen. Diese sind in einer solchen Vielzahl vorhanden, daß der Spieler aus dem Staunen nicht mehr herauskommt.

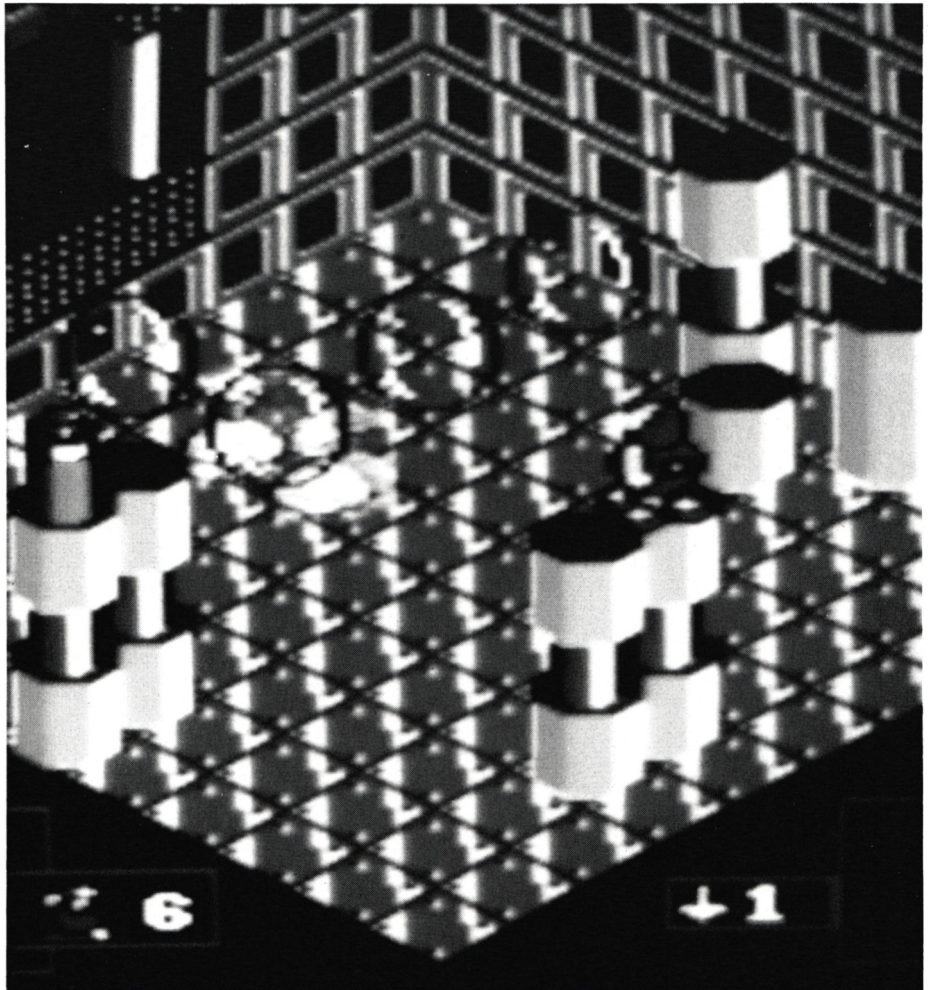
Neben schwebenden Riesenmessern sind auch einige Ganoven mit Keule vorhanden, die dem armen Brian nach dem Leben trachten. Wandelnde Skelette stellen ebenso eine Gefahr dar wie Stiefel, unter denen er zermalmt wird, wenn Sie nicht aufpassen und ihn rechtzeitig in Sicherheit bringen. Mal tauchen Piraten auf, mal Nashörner. Dann

sind Ritter zu sehen, die mit einer stacheligen Keule umherfuchteln und Enten, die zwar harmlos aus-



sehen, dies aber gar nicht sind. Das Ungeheuer von Loch Ness fehlt ebensowenig wie eine Bischofsmütze, aus der ein Schwert herausragt. Alle diese gefährlichen Sachen trachten unserem Helden nach dem Leben.

Die Grafik bei diesem Spiel ist zwar recht gut gemacht, aber nicht überwältigend. Eben normaler CPC-Durchschnitt. „Brians Bloodaxe“ ist ein Programm nur so zum Spielen. Nicht besonders schwierig, eben nur mal zum Entspannen.



PALITRON ZERSTÖREN SIE DIE KRISTALLE

Beim Spiel „Palitron“ von „Edge“ geht es darum, einen Roboter der Servoklasse (was immer dies sein mag) so geschickt wie möglich durch eine fremde Stadt zu steuern; immer auf der Suche nach den materiezerstörenden Kristallen. Diese Kristalle zerstören Sie am besten dadurch, daß Sie andere Gegenstände auf sie herunterfallen lassen. Aber Achtung: Nicht alle Dinge, die Sie beim Herumstöbern in der feindlichen Stadt finden, sind dazu geeignet. Da hilft nur Probieren!

Dann gibt es allerdings auch noch Roboter, die Ihnen bei dieser schweren Aufgabe helfen können. Dies sind die kleinen, tellerförmigen Druiden mit den auffälligen Antennen. Wenn Sie denen die richtigen Anweisungen geben, helfen Sie Ihnen in schwierigen Situationen.

Das Spiel fällt auf durch gute dreidimensionale Raumgrafik und gelungene Farbgebung. Die Roboter

lassen sich leicht lenken: entweder durch die voreingestellten Cursortasten oder, nach entsprechender Tastenumdefinition, durch den Joystick.

Natürlich lauern auch hier wieder jede Menge Fallen auf „Ihren“ Roboter. Aber Sie werden es schon schaffen! Noch ein Tip: Wenn Sie eine Batterie sehen, dann nichts wie hin, denn dadurch erhöhen Sie Ihre Energie!

ZWEI DEMOS ZUM APPETIT- HOLEN

Zwei Demoversionen der beiden Spiele „Shadow Skimmer“ und „Shao-Lin's-Road“ sind ebenfalls noch auf der Diskette enthalten. Leider aber nur eben Kurzfassungen dieser beiden Spiele. Den „Shadow-Skimmer“ haben wir in der Ausgabe 10/87 ausführlich besprochen. Deshalb verzichten wir hier auf einen Bericht über dieses Spiel. Das andere Kurzspiel („Shao-Lin's-Road“) ist ein Karate-Spiel, in dem es dar-

um geht, möglichst viele der Gegner umzulegen. Und davon gibt es wahrlich genug, so daß Sie ständig damit beschäftigt sind, die immer wieder auftauchenden fernöstlichen Krieger „umzunieten“, bevor Sie selbst K.o. gehen. Wahrlich nichts Neues und auch nichts Originelles. Wie gesagt, leider nur ein Ausschnitt aus dem Gesamtprogramm.

DEMOS SIND IN DIESER FORM ALS WERBUNG FRAGLICH

Eigentlich schade, daß den drei vollständigen Programmen die zwei anderen nur in der Kurzversion vorhanden sind. Irgendwie kommt sich der Käufer da vor, als sei er „auf die Rolle“ geschoben worden. Oder ist das Absicht?

Der Preis für die Diskette beträgt DM 39,95, für die Kassettenversion müssen Sie DM 29,95 berappen. Wir haben bewußt auf eine Gesamtwertung verzichtet. ■

JE

schneider
aktiv
special

Das Beste aus Schneider Aktiv
Sammelband Nr. 2/88
DM 19,80-ÖS 158-SFR 19,80

CPC-TEST-JAHRBUCH 88

CPC TEST JAHRBUCH 1988

Hard- und Software im Test

Alles über die CPC!

Kaufberatung

**Zwei
starke
Helfer
für jeden
CPC-
User!**

**Test-
Jahrbuch 88**

**Software-
Jahrbuch 88**

**Alles über
und für den CPC**

**An ausgewählten
Kiosken und im
Bahnhofs-Buchhandel**

schneider
aktiv
special

Nr. 1/88
DM 14,80
ÖS 124
SFR 14,80

CPC SOFTWARE JAHRBUCH 1988

SCHNEIDER AKTIV-SOFTWAREJAHRBUCH 88

Die schönsten Spiele - Die besten Anwender-Programme - Hilfreiche