DM 14,80/ÖS 124/SFR 14,80

Nr. 6/88

CPC-WELT

CPC SPECIAL



STORY:
Homecomputer – Erfolge und Flops

SOUND:
Der CPC in Stereo

SUPER:
RAM-Disk für den CPC

SPECIAL:
Computer-Wunder Archimedes

Nachruf auf die Freaks

Mit den Computern der „neuen Generation", so scheint es, wächst auch die Zahl der Anwender der „neuen Generation". Grund genug, einmal an die guten alten Computer-Zeiten zu denken.
Die ersten Heimcomputer waren eine Sensation. Endlich standen auch einem Privatmann die Möglichkeiten von Rechnern offen, die sich große Firmen schon lange leisten konnten. Große Computer-Leistungen zu einem erschwinglichen Preis – was wollte man mehr?
So schien es zumindest. Doch man wollte mehr: Software. Aber wo gab es die Programme, die sich der Anwender so sehnlich wünschte? Vorerst waren sie nirgends zu bekommen. Es blieb also nur eine Möglichkeit: selbst programmieren. Die Sprache BASIC war im Rechner eingebaut. Nichts schien einfacher zu sein, als die benötigte Software zu erstellen.
Daß das Programmieren nicht so leicht ist, wie es scheint, merkte mancher schon nach kurzer Zeit. Aber was half's – irgendwie würde man schon zurechtkommen. Das erste eigene Programm, nicht hundertprozentig lauffähig, aber brauchbar, war Grund genug zum Feiern.
Irgendwann kam dann auch die Industrie auf die Idee, daß hier eine Marktlücke sei. Spiele, Utilities und Anwendungen kamen auf den Markt. Anfangs war zwar manches eigene BASIC-Programm noch besser als die Software aus dem Laden. Aber auch die Profis wurden professioneller.
Nach kurzer Zeit konnte kaum ein Privat-Anwender noch Software erstellen, die sich mit kommerziellen Programmen vergleichen ließ. Sicher gab und gibt es hier und da Ausnahmefälle. Doch wer hat schon die Zeit, tage- und nächtelang an einem Programm zu arbeiten, das er für 49 Mark 80 im Kaufhaus kaufen kann?
Die Heimcomputer-Preise sanken drastisch: Die Technik wurde in geradezu ungeheurem Tempo weiterentwickelt. Bei den Freaks wurden gleichzeitig die Rufe nach dem Allerneuesten immer lauter. Der typische Computer-Aufsteiger begann seine „Karriere" mit einem Sinclair ZX-80. Inzwischen besitzt er Super-Maschinen vom Range eines Atari ST, Amiga oder PC-AT.
Mit der Weiterentwicklung der Hardware stiegen auch die Anforderungen an die Software. Ist ein Acht-Bit-Rechner wie der CPC noch einfach zu durchschauen, so benötigen Sie für eine ernsthafte Beschäftigung mit dem Amiga bereits eine gehörige Portion Grundlagenwissen.
Auch die Programmiersprachen wurden komplizierter. BASIC ist eben doch nicht gleich BASIC. Einschalten und einfach mal sehen, was sich tut – das war einmal.
Die erste Hürde, die es bei „modernen" Systemen zu überwinden gilt, ist die Aktivierung der Sprache selbst. Anschließend müssen Sie sich mit einem Editor-Interpreter- oder Editor-Compiler-System vertraut machen. Erst dann können Sie daran denken, die erste BASIC-Zeile einzutippen.
Die Software-Firmen, die solche Programmiersprachen herstellen, geben ihren Machwerken hochtrabende Namen wie „Programming Environment" oder „Development System". Professionelle Programmierer wissen die Unterstützung, die ihnen solche Systeme bieten, sicher zu schätzen. Für den privaten Anwender, der auf die Schnelle ein kleines Programm schreiben will, bewirken diese überflüssigen „Hilfen" eher das Gegenteil.
Eine Firma ist inzwischen so weit gegangen, ihr Programm in einer Anwender- und einer Programmierer-Version anzubieten. Andere Unternehmen werden diesem Beispiel folgen. Sind die Zeiten der programmierenden Anwender endgültig vorbei?
Zumindest die Software-Industrie wird sich das wünschen. Wer nicht selbst programmiert, muß die Software kaufen. Läuft nicht alles zur Zufriedenheit, legt man auch noch das Geld für das neue Update hin. Ein Vierteljahr später kommt dann das neueste Update, ein halbes Jahr darauf das allerneueste. Und so weiter.
Irgendwo aber muß es sie noch geben, die richtigen Freaks. Das beweisen nicht zuletzt die zahlreichen Programm-Einsendungen, die wir erhalten. Von der selbstgestrickten Anwendung, die entstand, weil die Software-Industrie in einem speziellen Bereich keine Absatz-Chancen sieht, bis hin zu Programmen, die aus Spaß an der Freud' entstanden sind und das Letzte aus dem CPC herausholen, ist alles vertreten.
CPC-User, so scheint es, bleiben weiterhin aktiv. Eines der Highlights haben wir für Sie herausgesucht: Ein Grafik-Programm für den 464, das keine Möglichkeit ungenutzt läßt.
Wir haben lange überlegt, ob dieses Programm für eine Veröffentlichung geeignet sei. Immerhin umfaßt allein das Listing fast dreißig Seiten, und das Programm ist nur für 464-Besitzer geeignet. Wir haben uns dennoch entschlossen, es Ihnen nicht vorzuenthalten – unter anderem, um zu zeigen, was sich mit etwas Fantasie auf dem CPC machen läßt.
Ein weiteres Thema dieses Hefts der CPC-WELT läßt sich mit „Vergangenheit und Zukunft" überschreiben.
Die junge Vergangenheit wird repräsentiert durch einen Rückblick auf Erfolge und Flops der

Homecomputer-Szene. Vielleicht erinnert sich der eine oder andere von Ihnen noch an diese „kleinen" Rechner. Den Einsteigern bieten wir Gelegenheit, einen Blick ins Computer-Museum zu werfen.
Die Zukunft ist vertreten durch den Acorn Archimedes, über den wir bereits in der letzten Ausgabe der CPC-WELT berichteten. Die Resonanz auf diesen Beitrag war so überwältigend positiv, daß wir Ihnen weitere Informationen auch in diesem Heft nicht vorenthalten. Eine feste Archimedes-Rubrik finden Sie übrigens in unserer Schwester-Zeitschrift PC POPULÄR. Auch wir werden Sie weiterhin über dieses Gerät informieren.
Egal, wie die neuen Maschinen heißen mögen: Über kurz oder lang werden sie die altbewährten Acht-Bit-Rechner ablösen. Doch welcher private Anwender braucht ein halbes Megabyte Speicher? Wer verwaltet solche Datenmengen, daß ein schneller 68000-Prozessor notwendig ist? Wer benötigt Grafik in 256 Farben und Sound auf zehn Kanälen? Zugegeben: Auch ich schreibe meine Texte auf einem PC. Doch zum Programmieren (und auch schon mal zum Spielen) setze ich mich lieber an den CPC. Und manchmal packe ich sogar den alten ZX-81 wieder aus.

Ihr
Alwin Ertl

REPORT

Computer, die Geschichte(n) machten
Erfolge und Flops aus der Geschichte der Homecomputer
Seite 4

Wie in Babylon
Die Vielfalt der Programmier-Sprachen
Seite 117

Das Renn-Tier
Neues vom Super-Computer Archimedes
Seite 136

BASIC-GRUNDLAGEN

Die Welt der Grafik
Der dritte Teil unseres Grafik-Kurses
Seite 10

Wer sucht, der findet
Schnelle Suchverfahren in BASIC
Seite 29

PROGRAMMIER-SPRACHEN

Sprechen Sie C?
Unser C-Kurs wird fortgesetzt
Seite 18

Fill-Routine für Turbo Pascal
Flächen füllen leicht gemacht
Seite 22

HARDWARE

Volle Dröhne
Stereo-Verstärker für den CPC selbst gebaut
Seite 20

TEST & TECHNICK

Disketten stets griffbereit
Testbericht – einmal anders
Seite 83

Das dritte Laufwerk
RAM-Disk für die CPCs
Seite 141

SERVICE

Dialog
Fragen, Wünsche, Anregungen – Wir haben immer ein offenes Ohr
Seite 15

NEWS

Neuigkeiten rund um den Computer
Seite 26

BÖRSE

Unsere kostenlosen Kleinanzeigen
Seite 126

INTERN

Leser als Autoren
Unsere Wunschliste für Programm-Einsendungen
Seite 60

LOAD & RUN

Die neuesten Spiele auf 16 Seiten – Getestet und für gut befunden von der Redaktion LOAD&RUN

LISTINGS

The CPC Graphics Designer
Unser Listing des Monats: Ein Grafik-Editor der Spitzenklasse
Seite 32

Checker
Der Eingabe-Checksummer für fehlerfreies Abtippen
Seite 59

Eck-Karte
Glückspiel ohne Risiko – auf dem CPC
Seite 84

Slot-Maschine
Eine echte Automaten-Simulation
Seite 87

Mau Mau
Kartenspiel mit Glück und Strategie
Seite 88

Winkelzüge
Wem „Mühle" nicht mehr genügt, der wird begeistert sein
Seite 93

Moonlander
Ein Spiel für Astronauten und alle, die es werden wollen
Seite 96

Harry
Auf Schatzsuche im verwunschenen Schloß
Seite 99

TaDat
Dateiverwaltung mit nützlichen Zusatz-Funktionen
Seite 103

Ordnung leicht gemacht
Programme schnell gefunden – durch Ordnung in der Disketten-Sammlung
Seite 109

Computer, die Geschichte(n) machten

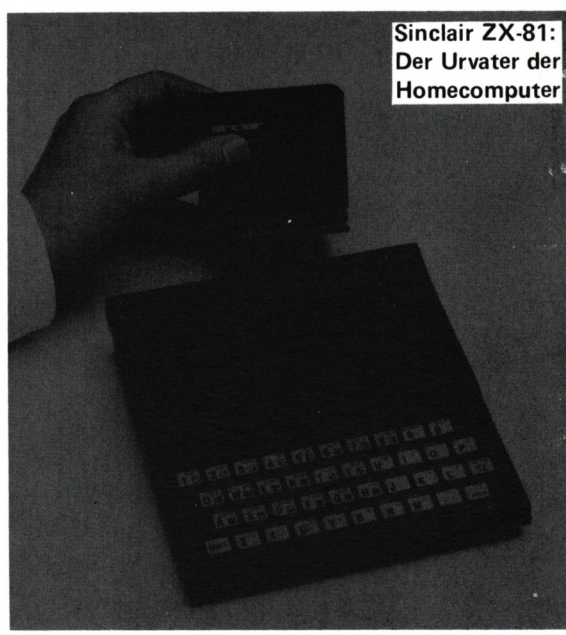
Erfolge und Flops aus der Geschichte der Homecomputer

Vor acht Jahren begann das Zeitalter der Homecomputer. Die Rechner, mit denen damals einige Hersteller den Anfang wagten, verschwinden langsam aus unserem Gedächtnis. Noch schlimmer ergeht es jenen Maschinen, die bereits bei der Markteinführung als Flop galten. Jüngere Computer-Begeisterte haben kaum etwas von diesen „Fehltritten" gehört. Alles ging so schnell, daß es heute bereits Geschichte ist. Aber eine interessante.

Die Ansprüche, die heute an einen Homecomputer gestellt werden, sind vergleichsweise enorm: 512 KByte Arbeitsspeicher, Maus, Supergrafik und haufenweise leistungsfähige Programme. Wer hat noch nicht, wer will noch mehr? Dabei ist es gerade mal sieben Jahre her, daß man sich wie ein Schneekönig über einen Arbeitsspeicher von einem KByte freute. Noch größer war die Sensation, daß ein Homecomputer Farbe auf den Bildschirm brachte. Diese Maschinen waren es, denen wir unser Hobby zu verdanken haben. Sie führten den Anwender in die primitivsten EDV-Kenntnisse ein und bewiesen den großen Firmen, daß es sich lohnt, auf dem Gebiet der Homecomputer Entwicklungsarbeit zu leisten. Ganz oben auf der Liste der Innovationen standen deshalb, kaum verwunderlich, die Urväter einer ganzen Industrie.

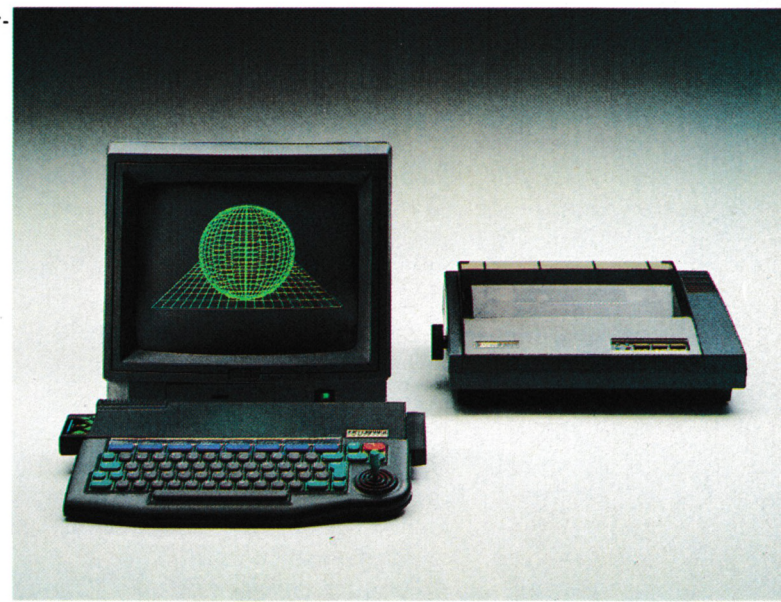
Sinclair ZX 80 – Der Erste und der Wichtigste

Geht es um Computergeschichte, dann reden alle vom Commodore PET oder von den ersten Apple-Computern. Sicherlich haben die zwei Geräte in der „Früh"-Geschichte der Massen-Computer Schule gemacht, aber sie waren wegen ihres hohen Preises nicht der Startschuß der Homecomputer für jedermann.



Sinclair ZX-81: Der Urvater der Homecomputer

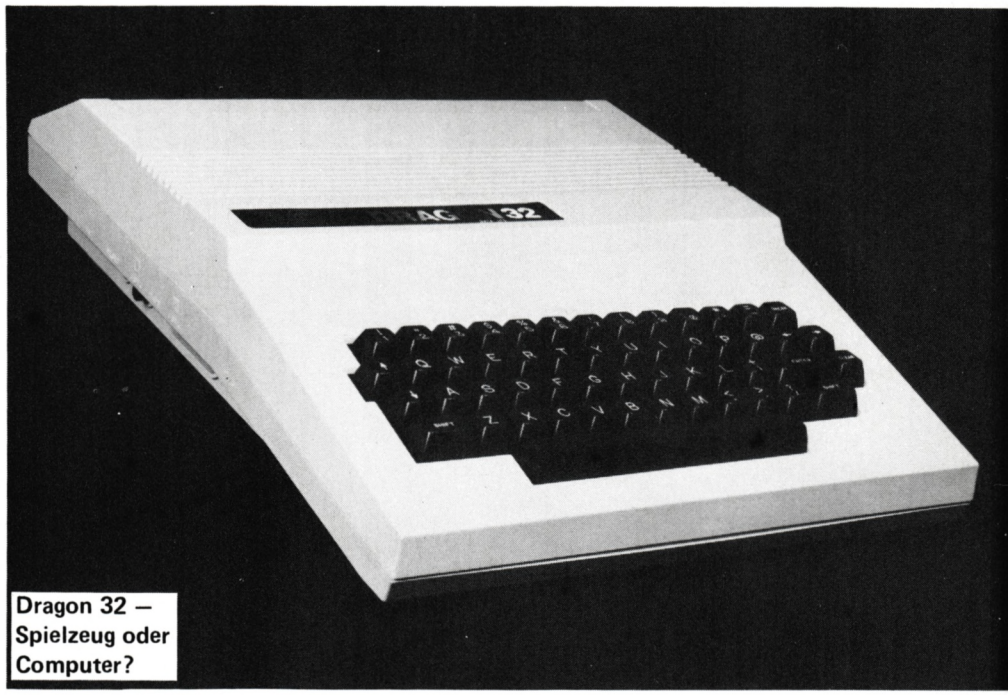
Den Knall dazu lieferte ein spleeniger Engländer namens Clive Sinclair. Er kam doch tatsächlich auf die Idee, einen Rechner herauszubringen, den sich fast alle leisten konnten. Für nur achthundert Mark wurde ihm der ZX 80 aus den Händen gerissen. Aber die Leistungsmerkmale waren ja auch erstaunlich: Ein volles KByte RAM und ein BASIC-Interpreter im ROM machten Laune zum Programmieren. Bei der Folientastatur war jede Taste vierfach belegt und es gab einen TV-Anschluß. Sogar ein Druckeranschluß war vorhanden, und der – nicht im Preis enthaltene – Thermoprinter sprühte Funken, daß man sich wie in der Silvesternacht bei einem Feuerwerk fühlte.
Das mag sich ironisch anhören, aber damals nahm die Kaufwut gar kein Ende und der Preis konnte auf fünfhundert Mark gesenkt werden. Meister Sinclair selbst machte sich ans Werk, um eine 16-KByte-Speichererweiterung zu liefern und die Hersteller von Zubehör schafften es sogar, ein 64-K-Byte-Ram zu entwickeln. Anstelle der schlechten Folientastatur, deren einziger Vorteil es war, daß man sie schnell einmal feucht abwischen konnte, kamen Ersatz-Keyboards auf den Markt, die die Arbeit erträglich machten.
Der BASIC-Interpreter war zwar einschläfernd langsam, aber was machte dies, wenn es der einzige war, den man sich leisten konnte. Da wurde halt in Assembler geschrieben, denn der Maschinencode machte die Schachtel zwanzigmal so schnell.
Der Konstrukteur und Hersteller beobachtete die Sache mit Wohlgefallen, verdiente er doch daran. Schließlich wurde er wegen dieses wirtschaftlichen Aspektes sogar geadelt, was ihn moralisch verpflichtete, sein Einkommen in



Die Oric-Computerserie: Mißglückter Versuch, den Sinclair-Rechnern Konkurrenz zu machen

Der Enterprise, ein teurer Super-Computer

England zu versteuern, auch wenn ihm dabei die Tränen in die Augen stiegen. Bei Computerfreaks gilt er heute schlicht und einfach als Genie, welches nicht ganz von dieser Welt zu sein scheint. Aber auch seine Fehlentwicklungen, die er sich im Laufe seines Erfinderlebens leistete, werden anerkannt. Dahinter verbirgt sich ja jene Innovation, die man bei vielen anderen Ingenieuren und Herstellern schmerzlich vermißt.
Mag der ZX 80 wegen seines windschnittigen Designs auch bei vielen heute als Keil zum Offenhalten der Tür dienen; Sir Clive Sinclair schuldet man dafür großen Dank.



Dragon 32 – Spielzeug oder Computer?

Commodore VC 20 – Keine Zeit, ein Star zu werden

Commodore war die nächste Firma, die auf den Privatanwender schielte. Mit dem PET, einer merkwürdigen Mischung aus Taschenrechner mit Kassettenrekorder und EDV-Anlage, hatten sich die Amerikaner am Markgeschehen beteiligt. Chuck Peddle zeichnete dafür in technischer Hinsicht verantwortlich. Er gilt als ebenso innovativ wie Clive Sinclair. Sein Manager bei Commodore hieß damals Jack Tramiel (richtig: der heutige Atari-Chef persönlich). Bei einer solchen Paarung konnte man 1981 eigentlich nur eines machen: Homecomputer entwickeln, herstellen, verkaufen und reich werden.
VC20 hieß das Prachtstück, wegen dem so mancher Hobby-Programmierer vor Freude an die Decke sprang. Der Arbeitsspeicher war im Zuge der fortschreitenden Technik auf 3,5 KByte angewachsen, und mit einem Modulator ließen sich bunte Bilder auf dem heimischen Fernseher erzeugen. Dazu gab es einen Joystick-Anschluß – und die Video-Games, bisher nur aus Spielhallen oder von teuren Konsolen bekannt, hielten Einzug ins Wohnzimmer. Der BASIC-Interpreter war schnell, was auf den von Chuck Peddle entwickelten Prozessor zurückzuführen war. Dazu kam eine Tastatur, die in der damaligen Zeit professionell genannt werden konnte, ohne daß den Werbeleuten die Schamesröte ins Gesicht schießen mußte. Wer beim ZX 80 noch keinen Spaß am EDV-Hobby gewinnen konnte, der wurde spätestens jetzt zum Freak.
Einen Wermutstropfen gab es allerdings auch: Commodores Preispolitik. Mit einer Datasette kostete der Einstieg satte achthundert Mark. Der VC 20 wurde oft, aber mit knirschenden Zähnen, gekauft. Es hätte etwas werden können aus dem Kleinen, obwohl man ihm das Design eines Kopfkissens verpaßte. Ein paar Speichererweiterungen und ein wenig mehr Systempflege hätten ihm gut getan, aber das Team Peddle–Tramiel gab einfach keine Ruhe. Bereits 1982 betrat der Nachfolger C64 die Bühne, und seitdem bestimmt er, wo es langgeht.

Pleiten, Pech und Pannen

Die Urviecher der Homecomputer sind nach einem kurzen, aber erfolgreichen Dasein im Markt längst in den wohlverdienten Ruhestand getreten. Je größer jedoch eine Branche wird, um so mehr Produkte erscheinen, die sich nicht durchsetzen können. Sir Clive Sinclair hatte es ja allen vorgemacht: Er lag auf Erfolgskurs, obwohl er mit nichts anderem als einer guten Idee gestartet war. Commodore und die schnell gegründete Firma des Engländers verdienten sich goldene Nasen. Nun, Ideen hatten andere auch. Und Geldverdienen macht jedem Spaß. Die Folge war, daß kein Monat verging, in dem nicht in den Fachzeitschriften von der Premiere eines neuen Homecomputers zu lesen war. Selten war es eine Galavorstellung, und oft wurde nur für die Kritiker aufgespielt. Hersteller, deren Namen man in Verbindung mit dem Homecomputer zum ersten Mal hörte, warben mit immer neuen Superlativen. Nur die wenigsten Maschinen erreichten ein Minimum an Bekanntheit und viele verschwanden sofort wieder von der Szene. Gar mancher Rechner nahm gleich die ganze Firma mit in den Abgrund, wenn er sich sofort wieder verabschiedete.
Typisches Beispiel für einen leisen Abgang war der Aquarius. Mit diesem Homecomputer wollte der

Spielwarenhersteller Mattel seine Chancen nutzen und errang nicht mehr als einen kleinen Achtungserfolg. Das Beste an dem Vier-K-Byte-Rechner war das gefällige Design. Es genügte natürlich nicht, den Kunden anzusprechen. Die Gummitastatur war nur dazu geeignet, jene Interessenten abzuschrecken, die sich trotz des hohen Preises für den „Schönling" erwärmten.

Aquarius — Computer vom Spielwarenhersteller

Nachteilig war auch das Konstruktionsprinzig, bei dem man auf einen Erweiterungs-Slot baute. Zusätzliche Speicherkapazität sollte ebenso nachgekauft werden, wie ein BASIC-Modul. Jeder ahnte, was da an zusätzlichen Ausgaben auf ihn zukam.
In Zeiten, in denen der C64 schon bewies, wie es richtig gemacht wurde, geriet der Aquarius ganz selbstverständlich zum Flop. Mattel verkaufte die Rechte an eine kleine Firma namens Radofin Electronics. Mit einer Preissenkung versuchte der Aquarius noch einmal die Gunst der Käufer zu erlangen, aber vergeblich. Im Jahre 1984 wurde das letzte Gerät gefertigt.

Der Initiator Mattel hat sich auf dem Homecomputermarkt die Finger verbrannt und scheut seitdem das Feuer. Bei Action-Puppen und Spielzeugautos scheint dem Hersteller die Welt noch in Ordnung und der Markt sicherer. Plastik ist Plastik.

Oric — Doppelter Fehlstart

Wer einen Lötkolben besaß und in seiner Garage die Nachbarschaft nicht störte, der entwickelte nachts einen Homecomputer. Den Platinen-Layoutern eilte es, denn gar zu viele Firmen wollten sich ein Stück vom großen Kuchen sichern. Doch wenn die Zeit drängt, wird mancher Fehler gemacht.
Nur so ist es zu erklären, was mit dem Oric 1 passierte, der in aller Eile vorgestellt wurde. In Serie produziert wurde er jedoch nie. Tangerine Computers hieß die Firma, die sich Sir Clive Sinclair und seinem mittlerweile erschienen und erfolgreich verkauften „Spectrum" in den Weg stellen wollte. Sie wurde dabei überrollt.
Äußerlich ähnelten sich Spectrum und Oric durch das schwarze Gehäuse, doch innerlich gab es Unterschiede wie Tag und Nacht.



Der erste Semi-Profi unter den Heimcomputern war der Sharp MZ 80

Der Oric 1 versuchte sein Glück mit einem 6502-Prozessor und 16 KByte RAM. Ohne Probleme ließ sich ein Diskettenlaufwerk anschließen, welches nicht nur angekündigt, sondern tatsächlich im Handel war, in der damaligen, „wilden" Zeit ein Novum. Auch der Listingdrucker war zu haben und die ganze Konfiguration war recht preiswert. Warum konnte der Oric 1 den Spectrum, dessen Besitzer auf der Jagd nach einem funktionierenden Diskettenlaufwerk waren, trotzdem nicht in die Tasche stecken?
Neben allen positiven Seiten wies der Zwerg auch einen schweren Nachteil auf: Sein ROM mit dem Betriebssystem und dem BASIC war eine ewig sprudelnde Quelle von Fehlern. Hier machte sich die Eile der Konstrukteure am schlimmsten bemerkbar und erschreckte die potentielle Kundschaft. Die Tester der diversen Zeitschriften überboten sich mit Highscore-Listen der Fehler, die sie entdeckt hatten. Tangerine Electronics zog sofort den Oric 1 zurück. Und ging in sich.
Leisten jedenfalls konnten sie sich die Ruhepause. Bei der Tochtergesellschaft eines Automobilkonzerns kamen die Gehaltsschecks der Angestellten auch dann pünktlich, wenn man mehr nachdachte als verkaufte. Ein Jahr kreißte der Berg, dann gebar er den Oric 2. Kaum auf dem Markt, wurde er schon umgetauft: Die Vertriebsmanager erinnerten sich in letzter Sekunde daran, daß es eine Nummer Eins nie wirklich gab und das bißchen, was der Öffentlichkeit noch darüber im Gedächtnis war, dem Nachfolger bestenfalls schaden könnte. Der „neue Alte" hieß Oric Atmos und war tatsächlich besser.
Das neue Betriebssystem lief ausgezeichnet und verwaltete den auf 48 KByte vergrößerten Speicher, ohne den kleinsten Fehler zu machen. Nach wie vor gab es für den Atmos ein breites Peripherie-Angebot zu günstigen Preisen. Der

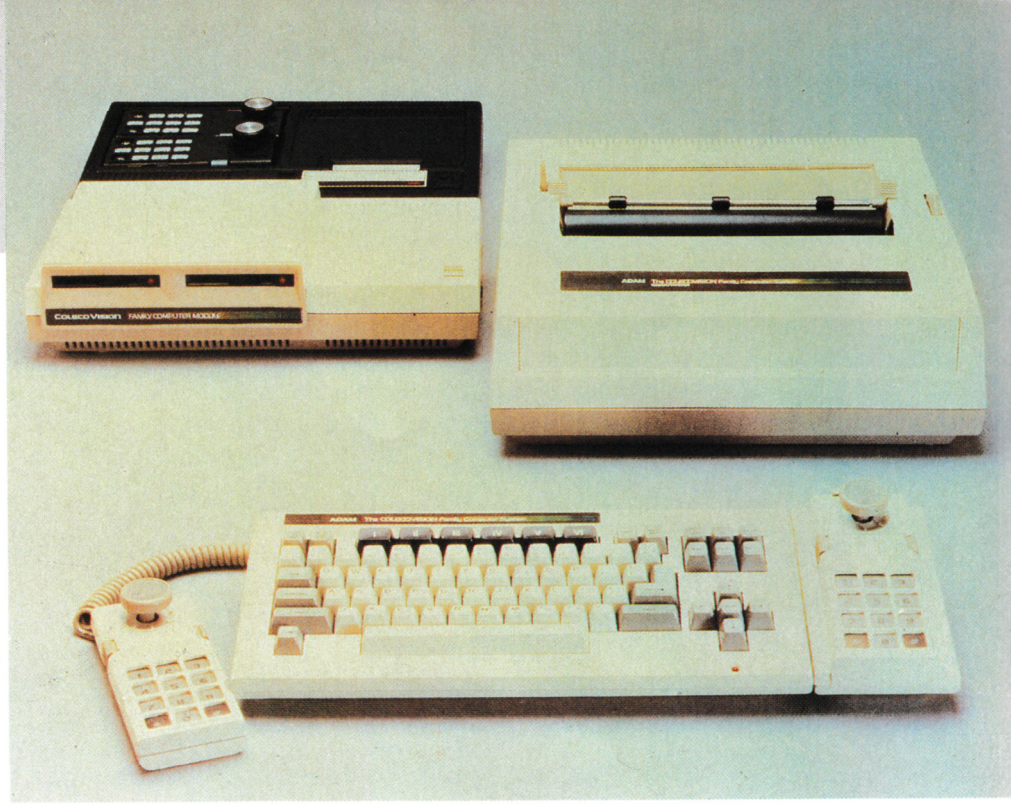
zweite Start war so recht nach dem Herzen der Entwickler. Aber: Niemand interessierte sich dafür. Der Spectrum hatte den Markt erobert und besaß Rang und Namen. Wenn in England jemand einen kleinen schwarzen Computer kaufen wollte, dann war es eine Sinclair-Maschine. Für den Oric Atmos wurde Schwarz jedoch zur Trauerfarbe. Er verhungerte mangels Software. Schließlich verkaufte Tangerine Electronics die Rechte nach Frankreich. Dort gab es ein paar Käufer, die nicht durch die Spectrum-Konkurrenz verwirrt waren. Ohne noch einmal Aufmerksamkeit zu erregen, wurde ein Jahr nach seiner zweiten Präsentation die Produktion des Oric Atmos eingestellt.

Enterprise — Viele Väter und kein Kind

In der Reihe der Flops ist der Enterprise einmalig: Bereits in der Jahresmitte 1983 wurde er zum erstenmal erwähnt, wenngleich es ihn noch gar nicht gab. Er geisterte in den Köpfen einiger Manager der Firma Samurai Computers. Kämpferisch wie der Firmenname war auch die Versprechung, daß es ein „ultimativer Homecomputer" werden sollte. So etwas machte die Presse natürlich neugierig, aber bedauerlicherweise

MSX-Computer





hatten die „Fernost-Krieger" nichts vorzuweisen, außer ein paar Entwicklungsplänen und den guten Willen.
Die Patente konnten sie wenigstens an Elan Computers verkaufen. Der Name des neuen Vaters paßte. Mit viel Elan stürzte sich die Firma in die Werbung, verstärkte den Rummel und erweiterte die Leistungsfähigkeit des Enterprise. Nur realisiert wurde er nie. Der Geister-Rechner blieb ein „Papiertiger".
Durch den Pressewirbel wurden wahre Begeisterungsstürme ausgelöst. Elan Computers kaufte, was zu kaufen war; also wieder nur Pläne und Ideen. Man war so enthusiastisch, daß man sich spontan entschloß, die ganze Firma nach der Neuerwerbung umzutaufen. Aus Elan Computers wurde Enterprise. Was aber niemand mehr erwartete, geschah unverhofft. Plötzlich, 1985, zwei Jahre nach der ersten Werbung, erschien der Enterprise. Nicht etwa auf dem Papier, sondern leibhaftig.
Der Rechner war tatsächlich etwas besonderes, zumindest äußerlich. Der eingebaute Joystick an der rechten Seite gab immer Anlaß zu Diskussionen. Die einen fanden die Idee gut, den „Freudenknüppel" aber von schlechter Qualität: die anderen lachten gleich darüber.
Die 128 KByte RAM hoben den

Colecos Adam: Nur die Pläne existierten

Computer aber tatsächlich in die Spitzenklasse. 1985 war kein Computer mit dieser Kapazität auf dem Markt. Auch die restlichen Spezifikationen ließen sich unter dem Stichwort „ultimativ" zusammenfassen: Eine eingebaute Textverarbeitung in Verbindung mit der guten Tastatur, ein ausgezeichnetes BASIC und die Möglichkeit, unter CP/M zu arbeiten.
Alles, mit Ausnahme des Joysticks als Billig-Ersatz für eine Maus, war fantastisch. Auch der Preis. Sagenhafte zwölfhundert Mark hatte der Käufer nur für das Keyboard zu investieren; und da hörte der Spaß auf. Für den notwendigen Monitor wären weitere achthundert Mark über den Ladentisch gegangen. Mit zweitausend Mark war die „Schmerzgrenze" bei den Freaks schon überschritten. So sensationell war die Kiste nun auch wieder nicht.
Da griff man doch lieber zu einem Konkurrenzprodukt namens Schneider, beziehungsweise Amstrad. Dieser Rechner stellte gleiche Leistung erst mal nur in Aussicht, war aber schon in der Grundkonfiguration wesentlich billiger. Die Liebe zum Computer geht auch durch die Brieftasche. Dies wurde der Firma Enterprise schmerzlich bewußt.
1983 wäre der Homecomputer

Spitzenklasse gewesen, dessen Preis der Kunde mangels Konkurrenzproduken nicht hätte vergleichen können. Vielleicht hätte er so seine Freunde gefunden. 1985 war der Enterprise immer noch gut, aber andere waren es auch – und sie waren gleichzeitig billiger. Im Kampf um die Kunden half der eingebaute Joystick nicht weiter.

Memotech MPX 500 – Niemand weiß, woran es lag

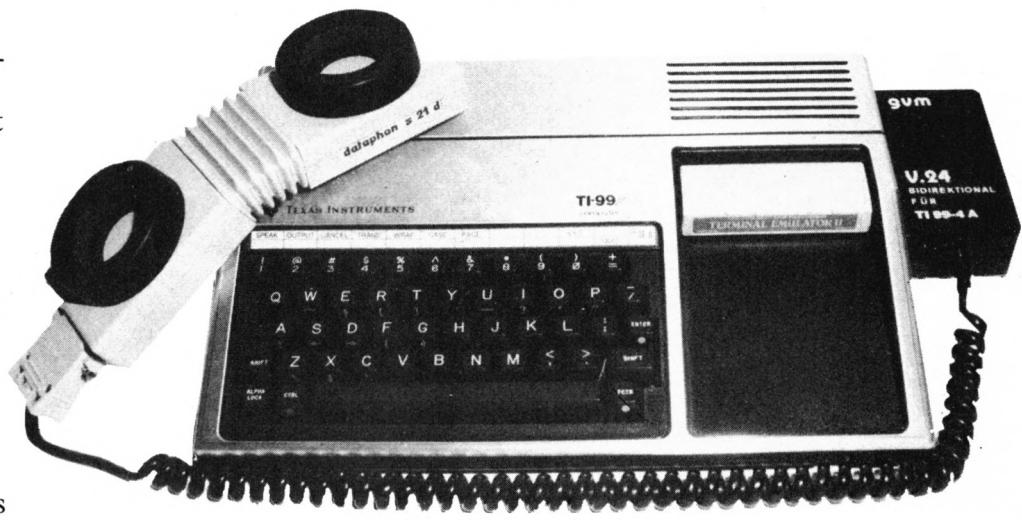
1983 war auch das Geburtsjahr des Memotech MPX 500. Die gleichnamige Firma gehörte zu den eingesessenen Computerherstellern und ist noch heute im PC-Bereich aktiv. Ihr Homecomputer, mit erstaunlichen Leistungsmerkmalen, konnte sich jedoch nie richtig durchsetzen, obwohl der Preis stimmte.
Das Gerät konnte als eines der ersten Rechner CP/M-Software verwenden. Speichersorgen gab es nicht. Die ursprüngliche Version mit 64 KByte ließ sich ohne Probleme auf 512 KByte ausbauen. Und dies im Jahr 1983, in dem so mancher Besitzer eines Homecomputers von 64 KByte nur träumen konnte. Die Tastatur des Memotech war, wie es sich für einen solchen Rechner gehörte, sehr übersichtlich und von ausgezeichneter Mechanik. Daß es dies alles schon bei einem Homecomputer gab, hätte eigentlich zu einem Verkaufserfolg führen müssen, denn er war die ideale Arbeitsmaschine. Nur wollte niemand eine solche Maschine, zumindest nicht in England. Vielleicht lag es an dem erschreckenden Wort „Arbeit".
In Großbritannien wollten die Käufer einen Spielcomputer. Etwas möglichst Buntes, was Krach machte, also einen guten Sound lieferte. Eben genau das, was die Firmen bislang als Homecomputer verkauften. WordStar, dBase und Multiplan waren reine PC-Software. Ein Rechner, für den es bei der Erstvorstellung keinen Pac Man oder keinen Defender gab, stieß auf äußerst geringes Interesse. Wozu sollte so ein Ding schon taugen? Noch heute haben die Games in Großbritannien einen höheren Stellenwert als die Anwendersoftware. Dies brach dem Memotech das Genick.
Schade um den Superrechner, von dem wir nicht einmal wissen, ob er noch hergestellt und verkauft wird, oder ob er sein karges Dasein leise und ohne Aufsehen beendete.

Dragon 32 – Der Drachenritt in die roten Zahlen

Warum es ausgerechnet Spielwarenhersteller drängte, einen Homecomputer herzustellen, mag wohl daran liegen, daß mit einem Rechner auch gespielt werden kann. Solch ein Vorhaben könnte theoretisch gelingen. Aber die Firmen sind schlichtweg überfordert, wenn es daran geht, das Produkt zu verbessern. Der Wechsel zum High-Tech-Hersteller kann nur selten vollzogen werden. Wie schwierig dies ist, mußte auch Mettoy erfahren.
Mettoy, ein englischer Spielwarenhersteller, versuchte sich, dem Zug der Zeit folgend, auf dem Computersektor. Mit dem Produktwechsel von Modellautos zu Homecomputern war die Firma sogar recht erfolgreich. Der Dragon 32 wurde gut präsentiert und verkaufte sich ausgezeichnet. Bei der Vorstellung 1982 stimmte die Technik. Das Prachtstück mit dem Design eines Apple IIe verfügte über ein 32 KByte großes RAM – das Doppelte dessen, was die Konkurrenz aufbringen konnte. Selbst die Tastatur war gut, was in England ja heute immer noch nicht selbstverständlich ist. Mettoy setzte sogar noch eins drauf und stattete den „Drachen" mit genormten Schnittstellen aus. Dies alleine war schon sensationell.
Ganz wichtig in der damaligen Unüberschaubarkeit des Marktes war das Engagement diverser Softwarefirmen. Für den Dragon gab es keine Probleme. Die Liste der Spiele war lang, der Spaß riesengroß. Der Anfangserfolg war also verständlich und der Rechner auf dem besten Wege, die magische Hunderttausender-Marke beim Verkauf zu erreichen.
Es wäre an der Zeit gewesen, einen Nachfolger vorzustellen, denn die Entwicklung ging ja mit Riesenschritten weiter. Mettoy war jedoch nicht in der Lage, an verbesserten Versionen zu arbeiten.

Lesen Sie bitte weiter auf Seite 128

Der TI-99/4A besaß bereits eine 16-Bit-CPU



Grafik-Kurs, Teil 3

Character-Sache

Der Schneider CPC bietet Ihnen die Möglichkeit, Grafik und Text gemischt darzustellen. Durch überlegten Einsatz dieser Option lassen sich Qualität und Aussagekraft jeder grafischen Darstellung erhöhen.

Das BASIC der Schneider CPCs stellt Ihnen einen **Grafik-Befehl** zur Verfügung, mit dem Sie Texte beliebig auf dem Bildschirm plazieren können. Dieser Befehl lautet

TAG

und bewirkt, daß der Text jeder nachfolgenden PRINT-Anweisung an der aktuellen Grafikposition erscheint. Über eine sinnvolle Übersetzung dieses Ausdrucks läßt sich streiten; to tag bedeutet im Deutschen anhängen. Sie können das so verstehen, daß der auszugebende Textstring an eine Grafik angehängt wird, indem er an der Position des Grafik-Cursors erscheint.
Damit Sie Texte wieder an den normalen Stellen ausgeben können, gibt es auch einen Befehl, um den TAG-Modus abzuschalten. Er lautet

TAGOFF

und sollte nach Möglichkeit nicht vergessen werden, da Sie sonst ein wildes Durcheinander auf den Bildschirm bekommen.

Doch wie bei jeder Anweisung gibt es auch bei TAG einiges zu beachten. Ein Textzeichen hat eine bestimmte **Ausdehnung,** die abhängig vom **Bildschirm-Modus** ist. Wieviel Platz ein solcher Character auf dem Monitor einnimmt, läßt sich leicht errechnen.
Bei normaler Textausgabe passen 25 Zeilen auf den Bildschirm. Bei 400 ansteuerbaren Bildpunkten in der Vertikalen ergibt sich daraus eine Ausdehnung von 16 Punkten entlang der y-Achse.

Zeichenausgabe mit Hindernissen

Der Platz, den ein Zeichen in der Horizontalen einnimmt, ist abhängig vom aktivierten Mode. In **Modus** 2 können Sie 640 Punkte ansprechen und damit 80 Zeichen darstellen. Wenn Sie 640 durch 80 dividieren, erhalten Sie den Wert 8. Daraus ergibt sich, daß ein Zeichen in Mode 2 einen rechteckigen Bereich von acht mal sechzehn Bildpunkten benötigt. Der Platzbedarf für die übrigen Modi kann auf dieselbe Art errechnet oder abgeleitet werden. Davon ausgehend, daß sich die Anzahl der darstellbaren Punkte und Zeichen jeweils halbiert, können Sie auf die Character-Größen in **Mode 1** und **Mode 0** schließen. In **Abbildung 1** ist das Verhältnis von Länge zu Breite eines einzelnen Zeichens grafisch dargestellt. Ein Kästchen entspricht dabei einem Quadrat aus acht mal acht Bildpunkten.
Um ein Zeichen exakt zu positionieren, genügt es nicht, lediglich seine Ausdehnung zu kennen. An welcher Stelle des Rechtecks wird mit der Ausgabe begonnen? Selbst, wenn Sie berücksichtigen, daß auf der y-Achse nur jeder zweite Punkt darstellbar ist, gibt es in Mode 2 bereits 64 verschiedene Möglichkeiten: Jeder Punkt der **Character-Matrix** käme als Startpunkt in Frage.
Welchen Sinn gäbe es aber, einen beliebigen Punkt des Rechtecks als Start zu wählen? Vernünftige Positionen sind nur die beiden linken Ecken. Die Autoren des **Locomotive-BASIC** haben sich für die linke obere Ecke entschieden. Um einen Text darzustellen, müssen Sie also zunächst den Grafikcursor an die gewünschte Stelle bewegen. Von diesem Punkt aus nimmt jedes Zeichen sechzehn Grafikpunkte nach unten in An-

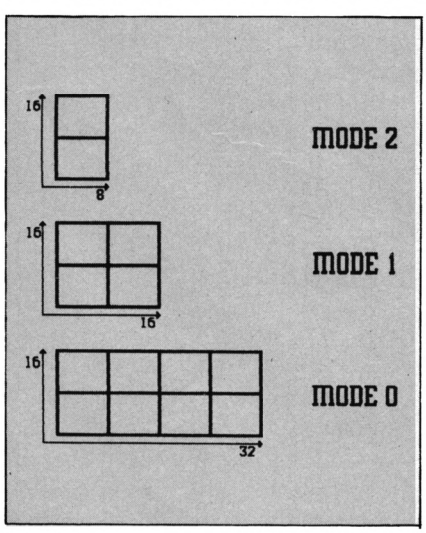
spruch. Wenn Sie etwa eine Linie beschriften wollen, müssen Sie dies berücksichtigen.

Auch Steuerzeichen werden dargestellt

Der TAG-Befehl besitzt jedoch noch eine Eigenheit, die Sie kennen sollten. Geben Sie folgende Programmzeile in Ihren CPC ein:

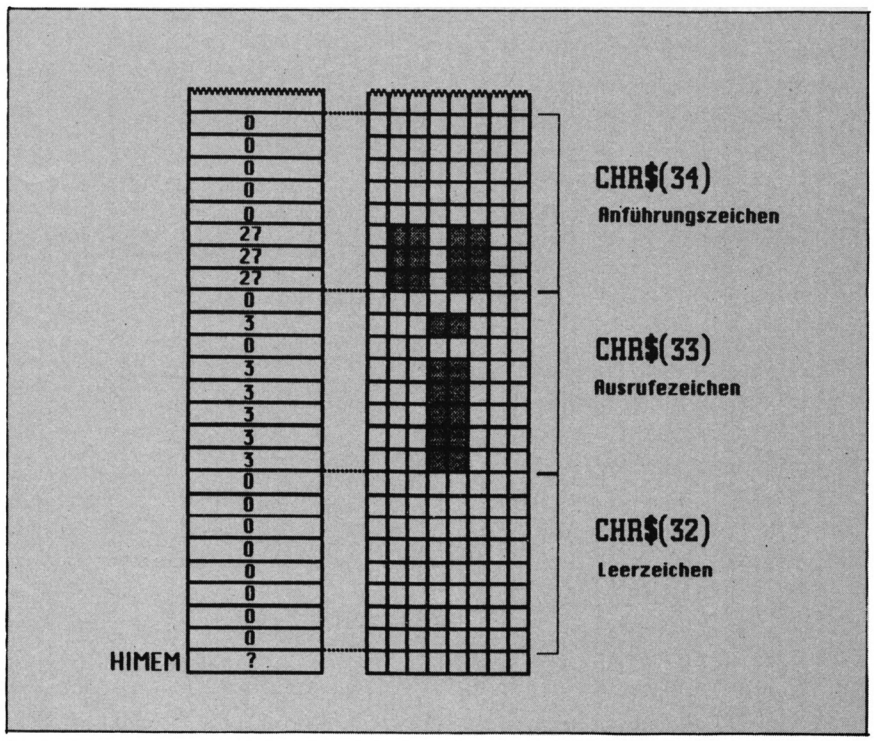
```
CLS:
TAG:
MOVE 300,100:
PRINT "Test"
```

Das Wort „Test" erscheint an der richtigen Stelle auf dem Bildschirm. Dahinter finden Sie aber noch zwei Zeichen, die höchst seltsam anmuten: einen nach links gebogenen Pfeil und einen Pfeil nach unten.
Die beiden Characters stellen Steuerzeichen dar. Der Pfeil nach links steht für **Wagenrücklauf**, der Pfeil nach unten symbolisiert einen **Zeilenvorschub**. Es handelt sich hierbei um die beiden Zeichen, die BASIC nach jedem PRINT-Befehl automatisch sendet, um den Text-Cursor an den Anfang der nächsten Zeile zu



Der Platzbedarf eines Zeichens ist abhängig vom Bildschirm-Modus

bewegen.
Andere Steuerzeichen sind etwa **Backstep** für eine Rückwärtsbewegung des Cursors um ein Zeichen oder **Bell** für einen Piepton. Im Textmodus erfüllen sie treu und brav ihre Aufgaben; bei der Ausgabe auf dem Grafikbildschirm werden sie hingegen durch die entsprechenden Symbole ersetzt.
Es wäre unsinnig, wenn Sie sich damit abfinden müßten, daß diese beiden Zeichen nach jedem Text erscheinen. In BASIC gibt es daher eine Möglichkeit, die Bewegung des Cursors in die nächste Zeile zu verhindern. Dadurch werden auch die oben genannten Steuerzeichen nicht ausgegeben. Sie kennen diese Anweisung seit Ihren ersten Experimenten in BASIC: Es handelt sich um das **Semikolon** nach dem PRINT-Befehl. Geben Sie die oben angeführte Programmzeile nochmals in Ihren Rechner ein, ergänzt um den Strichpunkt. Der Text erscheint wie angegeben auf dem Monitor.
Die Steuerzeichen, die Sie im Grafikmodus so einfach auf den Bildschirm bringen, bleiben im Textmodus in der Regel unsichtbar. Der Name Steuerzeichen oder Control-Code deutet bereits an, daß diese Characters eine steuernde Wirkung auf den Rechner haben, wenn sie durch einen PRINT-Befehl ausgegeben werden. Es gibt Control-Codes als Ersatz für BASIC-Befehle wie INK oder PEN sowie zur Ausführung bestimmter Cursor-Bewegungen. Die vollständige Liste finden Sie in Ihrem BASIC-Handbuch.
Listing 1 zeichnet eine Sinuskurve in einem Achsenkreuz. Anhand der Kommentarzeilen können Sie das Programm Schritt für Schritt nachvollziehen. Beachten Sie dabei besonders die **Zeilen 320 und 330.** Da der CPC in seinem vordefinierten Zeichensatz bereits Pfeile anbietet, müssen Sie das Ende der x- und y-Achse nicht Punkt für Punkt zeichnen. **Zeile 320** setzt zudem die Hälfte des Zeichens außerhalb des gültigen Bildschirmbereichs, so daß der gewünschte angeschnittene **Pfeil** erscheint.
Der CPC legt jedes Zeichen als Zahl von 0 bis 255 im Speicher ab. Dabei werden die Characters nach einem Code verschlüsselt, der als **ASCII** bekannt ist. Die Abkürzung steht für **A**merican **S**tandard **C**ode for **I**nformation **I**nter-



In dieser Form befindet sich die Zeichentabelle im RAM

change, also Amerikanischer Standard-Code für den Informationsaustausch. Bis zu einem Wert von 127 ist dieser Code genormt; darüber befindet sich ein Zeichensatz, der bei jedem Hersteller anders definiert ist.

ASCII – Das Alphabet des Computers

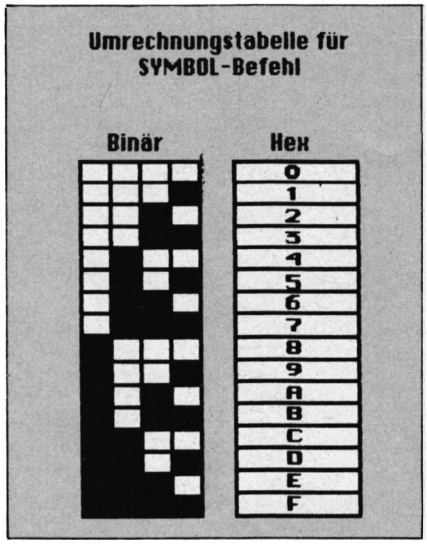
Alle Zeichen mit Steuerfunktionen haben die Werte 0 bis 31, alle für den Textausdruck benötigten Zeichen haben Werte von 32 bis 127. Dadurch können Sie jeden beliebigen Drucker an Ihren CPC anschließen, ohne daß es bei der Datenübertragung zu Mißverständnissen kommt.
Zusätzlich bietet Ihnen der CPC die Möglichkeit, das **Aussehen eines Zeichens** neu zu bestimmen. Aus leicht ersichtlichen Gründen hat diese Definition nur Einfluß auf das Aussehen eines Characters auf dem Monitor: Ein eventuell angeschlossener Drucker erhält lediglich die Code-Nummer des gewünschten Zeichens und verwendet für dessen Darstellung seine eigenen Informationen.
Der Befehl mit dem Sie die Matrix eines Zeichens neu definieren können, lautet

SYMBOL

Nach dieser Anweisung folgen neun Zahlen. Die erste davon gibt die Nummer des Zeichens an, das Sie neu gestalten wollen. Die restlichen acht Werte bestimmen das Aussehen.

Definition eigener Zeichen

Der CPC verwendet zur Darstellung eines Characters eine Matrix aus acht mal acht Punkten. Jeder der acht Parameter gibt die Gestalt einer horizontalen Punktreihe an, wobei ein Zeichen von oben nach unten definiert wird. Die Anzahl von acht horizontalen Punkten ist recht auffällig: Acht Bit bilden bekanntlich ein Byte. Da ein solcher Punkt entweder



Diese Tabelle ermöglicht einen raschen Zeichen-Entwurf

vorhanden oder nicht vorhanden sein kann, genügt ein Bit, um ihn eindeutig zu bestimmen. Ein **Byte** ist ausreichend für eine **Punktreihe**, wobei der Wert des Symbol-Parameters eine bit-signifikante Zahl ist. Um ein Zeichen vollständig zu definieren, benötigen Sie folglich acht Byte.
Um alle 256 Characters neu bestimmen zu können, müßten 2048 Byte zur Verfügung stehen. Das würde bedeuten, daß zwei KByte des Speicherplatzes allein für diese Definitionen verwendet werden müßten. Mit insgesamt 64 K-Byte Hauptspeicher ist der CPC jedoch nicht gerade großzügig ausgestattet, so daß Sie wohl nicht bereit wären, zwei KByte zu verschenken, wenn Sie keine neuen Zeichendefinitionen wünschen.

Zeichenmatrizen: Vagabunden im Computerspeicher

Normalerweise wird das Aussehen eines Zeichens daher aus dem **BASIC-ROM** gelesen. Dieser Festspeicher ist vom Hauptspeicher getrennt und konkurriert daher nicht mit Ihrem Programm. Einige Zeichenmatrizen werden beim Einschalten des Rechners ins RAM kopiert, so daß Sie immer die Möglichkeit haben, Characters ab einem Code von 240 neu zu gestalten.
Um diese Menge zu vergrößern, kennt das Schneider-BASIC den Befehl

SYMBOL AFTER

Mit der Anweisung **SYMBOL AFTER 0** steht Ihnen der gesamte Zeichensatz für neue Definitionen zur Verfügung. Alle Matrizen werden ins RAM kopiert, so daß BASIC darauf zugreifen kann. Damit verschiebt sich auch die oberste Speichergrenze nach unten. Sollte in einem BASIC-Programm ohne neue Character-Definitionen der Speicherplatz einmal wirklich knapp werden, so können Sie mit **SYMBOL AFTER 256** noch 128 Byte gewinnen. Die Verschiebung der Speichergrenze nach oben ist nämlich jederzeit möglich. Ihre eigenen Definitionen werden dabei gelöscht, und der CPC holt die Zeichenmatrizen wieder aus dem ROM.
Wenn Sie hingegen beschlossen haben, neue Zeichen etwa ab dem Code 100 zu definieren, sollten Sie in demselben Programm keinen weiteren SYMBOL-AFTER-Befehl mehr verwenden. Sie können das zwar tun; Matrizen, die Sie zwischen den beiden Anweisungen neu definiert haben, werden dann jedoch auf ihre Standardbelegungen zurückgesetzt. Der Zeichensatz wird, wie oben beschrieben, neu aus dem ROM kopiert.

MEMORY stört SYMBOL AFTER

In Verbindung mit dem MEMORY-Befehl können Sie mit SYMBOL AFTER einige unliebsame Überraschungen erleben. Vor dem SYMBOL-AFTER-Befehl dürfen Sie auf keinen Fall die oberste Speichergrenze mit **MEMORY** verändern. Durch die Zeichenreservierung muß diese Grenze verschoben werden können, was MEMORY verhindert. Bestimmen Sie nach SYMBOL AFTER eine obere Speichergrenze, so akzeptiert der CPC zunächst diesen Befehl.

Trifft er jedoch irgendwann auf ein neues SYMBOL AFTER, so bricht er das Programm mit einer Fehlermeldung ab: Der zu reservierende Speicher befindet sich jetzt in einem geschützten Bereich, auf den er keinen Zugriff mehr hat. Da MEMORY über das Programmende hinaus wirksam bleibt, können Sie Programme mit dieser Befehlskombination nur einmal mit RUN starten. Ein zweiter Lauf verursacht eine Fehlermeldung.
Die Zeichenmatrizen, die Sie mit SYMBOL AFTER ins RAM kopiert haben, können Sie mit dem PEEK-Befehl problemlos auslesen und für Ihre eigenen Zwecke verwenden. Damit ist es zum Beispiel möglich, eine Zeile in doppelter Höhe auszugeben, wie **Listing 2** demonstriert.
Zeile 200 berechnet, wo die Matrix eines Zeichens im Speicher zu finden ist.

```
100 'Sinus-Schwingung                    (1735)
110 '                                    (07E7)
120 'Farben festlegen                    (17FE)
130 INK 0,0:PAPER 0:BORDER 0:INK 1
,21:MODE 2                               (1881)
140 'Werte eingeben                      (1510)
150 INPUT"Frequenz in Hertz";f           (1E83)
160 INPUT"Amplitude";a                   (16F5)
170 'Kurve zeichnen                      (15AB)
180 CLS                                  (0693)
190 faktor=(640/(f*PI))                  (1EDF)
200 MOVE 0,200                           (0A99)
210 FOR i=0 TO f*PI STEP 0.1             (1B8D)
220 x=i*faktor                           (18C5)
230 y=200+SIN(i)*a                       (1AEC)
240 DRAW x,y                             (0F25)
250 NEXT i                               (0A65)
260 'Achsenkreuz zeichnen                (1BFF)
270 MOVE 0,200                           (0A25)
280 DRAWR 639,0'x-Achse                  (14C9)
290 MOVE 0,0                             (098D)
300 DRAWR 0,399'y-Achse                  (140B)
310 TAG                                  (06B0)
320 MOVE-4,399:PRINT CHR$(240);'Pf
eil nach oben                            (26E6)
330 MOVE 631,208:PRINT CHR$(243);'
Pfeil nach rechts                        (285B)
340 'Achsen beschriften                  (19FD)
350 MOVE 600,180:PRINT"1 sec";           (1655)
360 MOVE 4,399:PRINT"Amplitude";         (19A6)
370 TAGOFF                               (062B)
380 GOTO 380'Ready unterdruecken         (1EA8)
```

Listing 1: **Mit wenigen Berechnungen können Sie mathematische Formeln anschaulich darstellen**

Bestimmung einer Zeichenmatrix

Die acht Byte, die den ersten Character darstellen, beginnen bei der Adresse **HIMEM+1**. Acht Byte weiter oben befindet sich die Matrix des nächsten Zeichens. Nach dem Befehl

```
SYMBOL AFTER s
```

können Sie die Startadresse des Zeichens **n** mit folgender Formel errechnen:

```
start=HIMEM+1+8*(n–s)
```

Dabei müssen Sie beachten, daß sich die oberste Reihe des Zeichens an unterster Stelle im Speicher befindet. Die Zeichen stehen also auf dem Kopf, wie **Abbildung 2** es zeigt.
Zur Bestimmung der Gestalt einer Punktreihe können Sie jeden gesetzten Punkt als gesetzes Bit einer Binärzahl interpretieren und diese Zahl ins Dezimalsystem umrechnen. Dieses Verfahren ist jedoch etwas umständlich. Die ein-

```
100 'Zeichenmatrix                       (1496)
110 '                                    (07E7)
120 CLS                                  (061B)
130 SYMBOL AFTER 32                      (093E)
140 a$="Dies ist ein Test..."            (209B)
150 GOSUB 180                            (099E)
160 PRINT CHR$(10);CHR$(10);CHR$(1
0)                                       (1AA0)
170 END                                  (06B7)
180 'Doppelt hohe Zeichen                (1B57)
190 FOR i=1 TO LEN(a$)                   (1586)
200 start=HIMEM+1+8*(ASC(MID$(a$,i
,1))-32)                                 (2C23)
210 x0=PEEK(start):x1=PEEK(start+1
):x2=PEEK(start+2):x3=PEEK(start+3
):x4=PEEK(start+4):x5=PEEK(start+5
):x6=PEEK(start+6):x7=PEEK(start+7
)                                        (AA5F)
220 SYMBOL 254,x0,x0,x1,x1,x2,x2,x
3,x3:SYMBOL 255,x4,x4,x5,x5,x6,x6,
x7,x7                                    (6CB7)
230 PRINT CHR$(254);CHR$(8);CHR$(1
0);CHR$(255);CHR$(11);                   (28DF)
240 NEXT i                               (0A51)
250 RETURN                               (061B)
```

Listing 2: **Der direkte Zugriff auf die Zeichen-Matrizen ermöglicht erstaunliche Effekte**

fachste Möglichkeit besteht darin, jeder Zahl die **Kennung &X** voranzustellen und sie direkt in binärer Schreibweise – also mit Nullen und Einsen – anzugeben. Der Nachteil dieser Methode ist, daß sich die Programmzeilen für den SYMBOL-Befehl dadurch etwas in die Länge ziehen.
Ein Kompromiß ist hier die ideale Lösung. Mit der Kennung & können Sie dem CPC Zahlen in hexadezimaler Schreibweise mitteilen. Jede Ziffer einer Hex-Zahl steht für vier Bit. Wenn Sie ein Zeichen in einer 8x8-Matrix entwerfen, halbieren Sie einfach das Quadrat. Zur Bestimmung der Hex-Zahl können Sie **Abbildung** 3 heranziehen; **Abbildung** 4 ist als Entwurfschablone für Ihre Characters geeignet. Die Punktreihe

```
01100011
```

ergäbe im Hexadezimal-System den Wert

```
&63
```

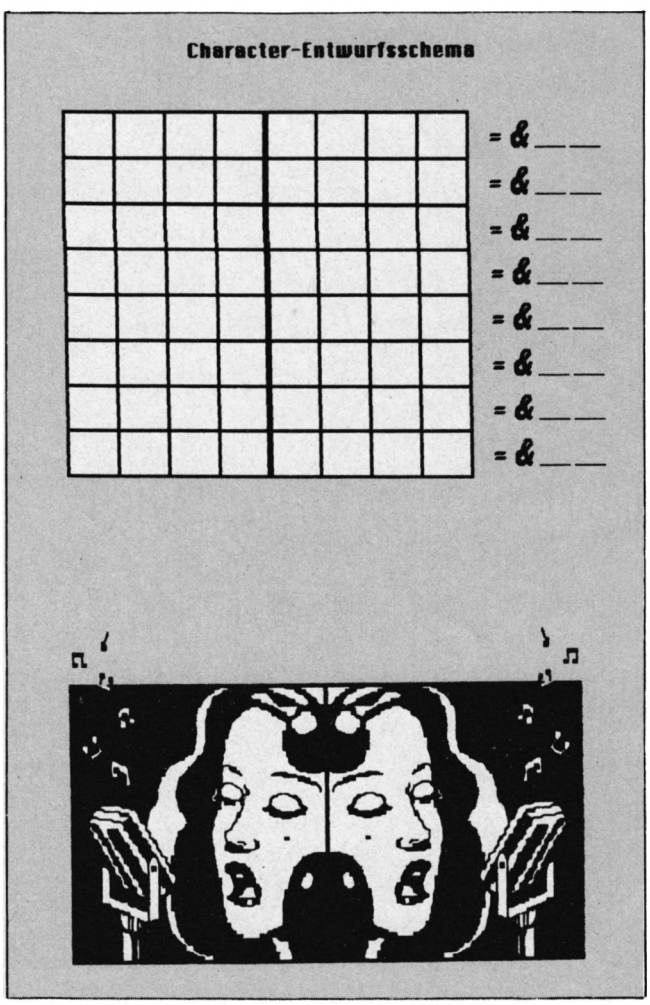
Bei Verwendung unserer Tabelle ersparen Sie sich die Mühe, ein neues und ungewohntes Zahlensystem zu erlernen.

Einsatzmöglichkeiten

Der SYMBOL-Befehl kann in Verbindung mit TAG neue Grafik-Möglichkeiten eröffnen. Sie können zum Beispiel Statistiken mit Symbolen beschriften, die Sie sonst punktweise mit PLOT auf den Bildschirm bringen müßten. Die Kenntnis der internen Zeichendarstellung ermöglicht Effekte wie die übergroße Schrift aus **Listing** 2.
Um Ihnen einen Ansatzpunkt für eigene Ideen zu liefern, sei hier noch eine abgewandelte Version des Zeichensatzprogrammes vorgestellt. **Listing** 3 stellt ebenfalls eine Schrift in doppelter Höhe dar, allerdings invertiert. Wenn Sie die gePEEKten Werte ins Binärsystem umrechnen, erkennen Sie sicher die angewandte Methode. *AE*□

Listing 3: **Invertierte Zeichen lassen sich auch ohne Einfluß auf die Matrizen darstellen**

```
100 'Zeichenmatrix                          <1496>
110 '                                       <07E7>
120 CLS                                     <061B>
130 SYMBOL AFTER 32                         <093E>
140 a$="Dies ist ein Test..."               <209B>
150 GOSUB 180                               <099E>
160 PRINT CHR$(10);CHR$(10);CHR$(1
0)                                          <1AA0>
170 END                                     <06B7>
180 'Doppelt hohe Zeichen                   <1B57>
190 FOR i=1 TO LEN(a$)                      <1586>
200 start=HIMEM+1+8*(ASC(MID$(a$,i
,1))-32)                                    <2C23>
210 x0=255-PEEK(start):x1=255-PEEK
(start+1):x2=255-PEEK(start+2):x3=
255-PEEK(start+3):x4=255-PEEK(star
t+4):x5=255-PEEK(start+5):x6=255-P
EEK(start+6):x7=255-PEEK(start+7)           <C2EA>
220 SYMBOL 254,x0,x0,x1,x1,x2,x2,x
3,x3:SYMBOL 255,x4,x4,x5,x5,x6,x6,
x7,x7                                       <6CB7>
230 PRINT CHR$(254);CHR$(8);CHR$(1
0);CHR$(255);CHR$(11);                      <28DF>
240 NEXT i                                  <0A51>
250 RETURN                                  <061B>
```



LESERBRIEFE

Prüfsummen nicht abtippen!

Bei unserer Hotline erreichte uns der Anruf eines Lesers, der Probleme mit dem Checksummer hatte: Nachdem er einige Listings der CPC-WELT 3/88 abgetippt hatte, bemerkte er, daß kein einziges Programm lauffähig war.
Wie sich im Laufe des Gespräches herausstellte, hatte er die neben den Programmzeilen abgedruckten Prüfsummen unseres Checksummers mit abgetippt. Daher noch einmal der Hinweis, wie Sie mit dem Checksummer arbeiten.
Tippen Sie den Checker ab und speichern Sie ihn etwa unter dem Namen CHECKER.BAS. Bevor Sie ein Listing abtippen, starten Sie ihn mit

```
RUN"CHECKER"
```

Entfernen Sie anschließend den BASIC-Teil des Checksummers mit

```
NEW
```

aus dem Speicher und geben Sie ein:

```
|EIN
```

Der Checksummer ist dadurch aktiviert. Tippen Sie das gewünschte Listing **ohne die angegebene Prüfsumme** ab. Nach Abschluß einer Zeile mit der ENTER-Taste erscheint die Prüfsumme auf dem Bildschirm. Vergleichen Sie die angezeigte Hexadezimal-Zahl mit der im Heft abgedruckten. Stimmen die Werte überein, so ist die Zeile korrekt. Unterscheidet sich die Zahl auf dem Bildschirm von der im Heft, so haben Sie einen Fehler gemacht.
An dieser Stelle noch ein Tip: Die ersten beiden Stellen der Prüfsumme symbolisieren die Länge der BASIC-Zeile. Weicht eine dieser beiden Ziffern von denen im Heft ab, so haben Sie wahrscheinlich ein Zeichen vergessen oder einen BASIC-Befehl falsch getippt. Die letzten beiden Stellen bilden eine Prüfsumme für die einzelnen Zeichen. Erhalten Sie hier eine abweichende Zahl, so haben Sie sich wahrscheinlich bei einem Variablen-Namen geirrt. □

Elektroniker sucht Hilfe

Gibt es ein Programm, mit dem ich Schaltpläne für Schütz-Schaltungen frei auf dem Bildschirm erstellen und anschließend auf dem Drucker ausgeben kann? Wichtig ist dabei auch, daß ich mit diesem Programm auch Meßreihen mit negativen Werten (Drehfelder, Sinusfunktion im Wechselstrom-Bereich) darstellen kann. Da ich beruflich sehr viel mit solchen Schaltungen und Meßreihen zu tun habe, würde mir ein derartiges Programm gute Dienste leisten.
Vielleicht hat einer unserer Leser ein vergleichbares Programm entwikkelt oder ist bereit, dies zu tun. Wer Rat weiß, wende sich an

Günter Oster
Königsbergerstr. 49
2208 Glückstadt

Dialog und Erscheinungsweise

Es wäre sehr gut, wenn im DIALOG ein Hinweis gegeben würde, auf welches Heft sich die Besprechung oder Änderung bezieht. Ohne diese Angaben muß man leider sehr lange suchen, um das Gewünschte zu finden.
Ich finde Ihre Zeitschrift inhaltlich gut. Nur: Warum haben Sie den Namen von SCHNEIDER AKTIV in SCHNEIDER CPC WELT geändert? Ihrem Vermerk in Heft 3/88 muß ich entnehmen, daß nur noch jeden zweiten Monat ein Heft erscheint, die Hefte zu 6 Mark also entfallen.

Josef Weiner
5010 Bergheim 3

Der fehlende Hinweis auf Ausgabe und Seite ist auch für uns oft Anlaß zu einer großen Suchaktion. Wir bitten daher unsere Leser, bei Anfragen zu Listings jeweils Ausgabe und Seite anzugeben. Bei Änderungen, die von unseren Autoren eingesandt werden, ist dies in der Regel ohnehin der Fall.
Die Namensänderung in SCHNEIDER-CPC-WELT hat ihren Grund in der Tatsache, daß wir uns inhaltlich auf die CPC-Modelle beschränken wollen. Der mangelnde Eingang an Beiträgen zu den PCs und den Joyce-Modellen, sowie das fehlende Interesse der Leserschaft an diesen Geräten – ein Ergebnis unserer Leser-Umfrage – zwang uns dazu. Damit tragen wir auch den Interessen der Mehrheit unserer Leser Rechnung. Durch die Trennung von Schneider und Amstrad sollte jedoch auch der neue Name nicht von Dauer sein; ab Ausgabe 4/88 sind wir daher als CPC EXTRA im Handel.
Auch die Umstellung auf zweimonatige Erscheinungsweise war wohlbegründet. Ein Ergebnis unserer Leserumfrage war auch der oft geäußerte Wunsch, mehr Informationen „gebündelt" zu erhalten. Bei einem Umfang von 144 Seiten zu einem Preis von 14,80 Mark bieten wir unseres Erachtens genug Lesestoff für zwei Monate, so daß diese Erscheinungsweise die günstigste sein dürfte. □

Sprachen im Vergleich

Ihren Artikel über den XBC-BASIC-Compiler in CPC EXTRA 4/88 habe ich mit großem Interesse gelesen. Ich besitze die XBC-Kassettenversion für den CPC 464 und habe ein Benchmark-Beispiel nicht nur mit diesem Compiler, sondern auch in Pascal und Assembler getestet. Hier die Laufzeiten:

- BASIC, REAL-Variablen: 131 Sekunden;
- BASIC, INTEGER-Variablen: 101 Sekunden;
- XBC-BASIC: zehn Sekunden;
- Pascal, INTEGER-Variablen: 19 Sekunden;
- Assembler: 0,98 Sekunden.

Der kleine Schönheitsfehler des XBC-Compilers (Copyright-Vermerk) konnte mittels eines Assembler-Editors leicht beseitigt werden.

Friedrich Oppermann
3150 Peine

Wir danken Ihnen für diese Hinweise, die für XBC-Anwender sicherlich hilfreich sind.
Uns ist natürlich klar, daß sich solche Schönheitsfehler wie ein Copyright-Vermerk von erfahrenen Anwendern leicht beseitigen lassen. Doch nicht jeder, der in BASIC programmieren kann, besitzt auch einen Debugger oder kann mit ihm umgehen. Daran sollten die Hersteller solcher Compiler denken, bevor sie jedem Software-Käufer zeigen, daß hier ein compiliertes BASIC-Programm vorliegt. BASIC hat, auch wenn es sich um eine Compiler-Version handelt, bei vielen Anwendern einen schlechten Ruf, der ihm eigentlich nicht gebührt. □

Formatierte Bildschirm-Ausgabe

Ich besitze seit kurzem den CPC 6128 und stehe vor folgendem Problem: Ich möchte ein Array A, das zweitausend Zahlen enthält, in Zeilen zu je sieben Zahlen ausgeben. Leider sind meine Versuche, dies zu bewerkstelligen, bis jetzt gescheitert. Für einen hilfreichen Tip wäre ich Ihnen dankbar.

Wilfried Fröhlich-Prieß
3391 Wildemann

Die beste Möglichkeit, eine Bildschirmausgabe in der von Ihnen gewünschten Form zu erzielen, bietet die MOD-Funktion des CPC. Ihr Programm muß lauten:

```
110 FOR n=1 TO
    2000
120 PRINT A(n);
130 IF n MOD 7=0
    THEN PRINT
140 NEXT
```

Zeile 120 bewirkt, daß jede Zahl unmittelbar hinter ihrem Vorgänger ausgegeben wird. Der MOD-Operator ermittelt den Rest einer Division. Der Wert

```
n MOD 7
```

ist immer dann Null, wenn der Schleifenzähler n ein Vielfaches von Sieben ist. Die PRINT–Anweisung im THEN-Teil bewirkt daher, daß nach jeder siebten Zahl eine neue Zeile begonnen wird.
Wenn Sie Zahlen verschiedener Länge verwenden, ist es eventuell wünschenswert, diese Werte rechtsbündig auszugeben. Zeile 120 muß dann lauten

```
PRINT USING
"#####";A(n);
```

Die Anzahl der # bestimmt, wie viele Stellen Sie für die Zahl reservieren. Bei Werten mit Nachkomma-Stellen können Sie das Format in der Form

```
"#####.##"
```

vorgeben. Der PRINT-USING-Befehl bietet noch viele weitere Möglichkeiten, die Sie am besten in Ihrem BASIC-Handbuch nachschlagen. □

Gemeinsam mit Kindern
Anwalt für das Leben
caritas

CPC-BASIC auf dem PC?

Seit 20 Monaten habe ich einen CPC 6128, mit dem ich täglich zuhause arbeite. Inzwischen habe ich, wie es meine Absicht war, eine große Anzahl Programme in BASIC erstellt. Im Wesentlichen geht es dabei um Berechnungen aller Art, die zum Teil sehr umfangreiche Formeln verwenden.
Berechnungen dieser Art sind „zu Fuß" ein recht mühsames Geschäft. Das Programm löst die Aufgaben spielend in kürzester Zeit. Deshalb interessieren sich Unternehmen für solche Lösungen, bei denen diese Berechnungen öfter vorkommen. Software hierfür gibt es offenbar nicht.
Mein Problem ist, daß keiner etwas mit den Drei-Zoll-Disketten und schon gar nicht mit CP/M oder AMSDOS anfangen kann. Deshalb werde ich, wohl oder übel, auf einen XT- oder AT-kompatiblen Computer umsteigen müssen. Meine Frage lautet: Wie kann ich meine auf Drei-Zoll-Disketten gespeicherten Programme auf 3,5- oder 5,25-Zoll-Disketten unter dem Betriebssystem MS-DOS übertragen?

Walter Wittke
2056 Glinde

Ein IBM-XT oder AT besitzt in der Regel eine serienmäßig eingebaute serielle Schnittstelle. Über diese Schnittstelle können Sie Daten senden oder, was in Ihrem

Fall wichtiger ist, empfangen.
Diese serielle oder RS-232-Schnittstelle steht Ihnen beim CPC nicht zur Verfügung. Sie ist jedoch als Zusatzgerät zu Preisen um 200 Mark erhältlich. Da die von Ihnen beabsichtigte Programm-Übertragung ein einmaliger Vorgang ist, können Sie ein solches Gerät eventuell leihweise von einem Bekannten oder einem Club erhalten.
Damit die BASIC-Programme übertragen werden können, müssen diese im ASCII-Format vorliegen. Auf dem CPC können Sie dies bewerkstelligen, indem Sie beim Abspeichern die Zeichenfolge ",A" anfügen. Der PC benötigt ein spezielles Programm, das solche ASCII-Files über die serielle Schnittstelle empfangen und wieder als Datei ablegen kann. Das könnte zum Beispiel ein Dfü-Programm sein. Da Sie BASIC-Programmiererfahrungen besitzen, können Sie ein solches Programm auch selbst erstellen.
Die ASCII-Dateien, die Sie so auf den PC übertragen haben, können Sie in den meisten PC-BASIC-Versionen laden. Probleme treten aber dann auf, wenn Sie versuchen, ein Programm zu starten.
In dem Listing, das Sie Ihrem Schreiben beigefügt haben, finden sich unter anderem die Befehle WINDOW, STRING$ und PRINT#x. Diese Anweisungen gelten nur für das BASIC des Schneider CPC.
Die BASIC-Versionen, die für PCs erhältlich sind (wie etwa GW-BASIC), kennen keine Fenster-Anweisungen. Auch einige Funktionen, die nicht im BASIC-Standard enthalten sind, sind ihnen unbekannt, zum Beispiel STRING$. Die entsprechenden Programmteile werden Sie daher anpassen müssen.
Ihre Programme sind zudem, wie wir feststellen konnten, hervorragend strukturiert. GW-BASIC bietet keine derartigen Möglichkeiten, so daß Sie die Zeilen linksbündig beginnen müssen.
Selbst wenn Sie die Daten auf einen PC übertragen haben, müssen Sie umfangreiche Änderungen an Ihren Programmen vornehmen. Die günstigste Lösung dürfte es also sein, auf der Basis der CPC-Listings die Programme in einer PC-BASIC-Version neu zu erstellen.
Beachten Sie beim PC-Kauf aber auch, daß auf diesen Rechnern BASIC erst in den Arbeitsspeicher geladen werden muß. Sofern BASIC nicht zum Lieferumfang des Rechners gehört, müssen Sie einen BASIC-Interpreter oder -Compiler zusätzlich kaufen. Diese Programme kosten in der Regel mehrere hundert Mark.
Einen ausführlichen Bericht über den Datei-Transfer vom Homecomputer zum PC sowie Testberichte über Personal Computer und zugehörige Software finden Sie in der neuesten Ausgabe der PC POPULÄR.



Bezugsquellen — Was gibt es wo? Von CPC-WELT getestet

Wir wollen hier keine Gratiswerbung betreiben – es finden sich auch die Vertreiber 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 diejenigen Interessenten sein, die bei ihrem Fachhändler auf leere Regale stießen.

Balloon (1/88)
Asphalt (1/88)
Fußball Manager (1/88)
Reisende im Wind 2 (1/88)
Sidewalk (1/88)
Renegade (1/88)
Classix 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

Plonker-Box (6/88)

Lindy-Elektronik GmbH
Karl-Kuntz-Weg 9
6800 Mannheim 25

Tas-Datenbank (1/88)

Henschke Datentechnik
Aidlinger Weg 6
7034 Gaertingen

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

Rushware GmbH
An der Gümpgesbrücke 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

BTX Modul zum CPC 464 (2/88)
Drucker LQ 3500 (1/88)

Im Fachhandel

---

dk'tronics Silicon Disc (6/88)

Weeske Computer &Elektronik
Potsdamer Ring 10
7150 Backnang

Sprechen Sie C?

Schleifenprogrammierung

Um Schleifenprogrammierung geht es in der heutigen Folge unserer C-Serie. Die wiederholte Abarbeitung eines Befehles gehört zu den wichtigsten Funktionen einer jeden Programmiersprache. Und gerade bei dieser Programmiertechnik hat C einiges zu bieten.

Die wichtigsten Befehle, die innerhalb einer Schleife stehen, wurden schon besprochen. Mit den Operatoren können mathematische Prozeduren aufgebaut werden, mit den Vergleichsfunktionen können Prüfungen innerhalb der Schleife vorgenommen und die Abbruchbedingungen festgelegt werden. C bietet dem Anwender folgende drei Methoden, um eine Befehlswiederholung zu programmieren:

for
while
do...while

Die Kleinschreibung weist – dies nebenbei – auf einen C-Sourcecode hin, während BASIC-Befehle groß geschrieben werden.

Ablauf mit Zählfunktion – Die for-Schleife

Fast jeder, der sich ernsthaft mit seinem Computer beschäftigt, kennt die FOR-NEXT-Schleife des BASIC-Interpreters. Das NEXT ist hier notwendig, um den Interpreter dazu zu bringen, an den Beginn der Schleife zurückzukehren. In C gibt es kein NEXT. Es ist nicht nötig, da der For-Befehl abgearbeitet wird, bis der Zähler aufaddiert ist oder eine Abbuchbedingung erfüllt wird. Sämtliche Werte oder Parameter sind hinter dem For-Befehl in Klammern anzugeben, so daß sich diese Syntax ergibt:

for (Zähler; Bedingung,
Inkrementieren)

Beim Zähler handelt es sich um die Deklaration einer Laufvariablen. Aus Gewohnheit durch BASIC benutzen die Programmierer auch hier die Buchstaben i, j und k. Diesen Variablen wird gleichzeitig mit der Deklaration ein Startwert zugewiesen.
Die Abbruchbedingung muß durch ein Semikolon abgegrenzt werden, ebenso wie die darauf folgende Inkrementierung. Soll die Schleife in einer bestimmten Anzahl durchlaufen werden, beispielsweise hundertmal, dann gilt zur Prüfung der Ausdruck i<100. Die Inkrementierung an letzter Stelle gibt die Schrittweite an, in der die Befehle der nächsten Programmzeile abgearbeitet werden. Wird lediglich Eins addiert, dann genügt ein doppeltes Pluszeichen mit der Laufvariablen. Ob Sie ++i oder i++ schreiben, ist dabei unerheblich. Der sonst so strenge Compiler akzeptiert beides.
Ähnlich dem STEP-Befehl in BASIC kann jede Schrittweite programmiert werden. Die Inkrementierung muß dann i=i+3 lauten, wenn zum Beispiel um drei erhöht werden soll. Abbildung 1 zeigt zwei C-Beispiele der For-Schleife und als Vergleich dazu die BASIC-Programmierung der gleichen Routine.
Achten Sie vor allem darauf, daß hinter der For-Anweisung kein Semikolon steht. Ist dies der Fall, dann wird die Schleife durchlaufen, ohne daß der Befehl in der zweiten Zeile ausgeführt wird. Erst wenn die Bedingung erfüllt ist, kommt die nächste Zeile an die Reihe. Statt hundertmal wird der Text nur einmal gedruckt.
Eine For-Anweisung gilt auch ohne Klammerausdrücke, aber Vorsicht: Die Schleife wird endlos durchlaufen, wenn nicht für einen Abbruch gesorgt wird. Mit der kleinen Funktion

Abbildung 1: for-Schleife

in C	und BASIC

`for (i=1;i<100;++i)`
`  printf("Text\n");`
	`FOR i=1 TO 100`
	`  PRINT "Text"`
	`NEXT i`

`for (i=1;i<100;i+3)`
`  printf("Text\n");`
	`FOR i=1 TO 100 STEP 3`
	`  PRINT "Text"`
	`NEXT i`

```
for(;;){
    break;
}
```

läßt sich also recht bequem eine INKEY-Schleife simulieren, wie sie von BASIC bekannt ist.

Komplexe-Bedingungen durch die While-Schleife

Die Bedingung in der For-Schleife war recht einfach. Es wurde nur geprüft, ob die Laufvariable den festgelegten Endwert erreicht hatte. Solange dies nicht der Fall war, solange also i<100 war, wurde die Schleife durchlaufen.
Dies ist der einfachste Vergleich, der innerhalb eines Programmes durchgeführt werden kann. Weitaus häufiger werden Operanden zur Festlegung einer Bedingung benötigt, oder die Abbruch-Bedingung richtet sich nach Variablen, die innerhalb der Schleife bearbeitet werden. Damit wird aber die Laufvariable überflüssig; sie kann sogar zu Fehlern führen, wenn sie zu niedrig angesetzt wird. Wer zum Beispiel ermitteln will, wieviele Datensätze in einer Datei liegen, dem nutzt die For-Schleife mit Laufvariable wenig. Der Programmierer weiß nicht von vornherein, wie oft die Schleife durchlaufen werden muß.
Aber außer der etwas großzügigeren Festlegung der Bedingung und dem Vorteil, daß auf eine Laufvariable verzichtet werden kann, gibt es keine weiteren Unterschiede zwischen den beiden Programmier-Methoden. Die while-Schleife besitzt die Syntax:

```
while (Bedingung)
        Anweisung
```

Wie dies in einem Programmtext zu lesen ist, zeigt Abbildung 2.
Die Funktionsweise des Mini-Programms ist recht einfach. Solange die Addition der Variablen wert nicht 20 ergibt, wird die Variable auf dem Bildschirm ausgegeben und anschließend um 1 erhöht. Dies ähnelt sehr einer For-Schleife, bei der als Bedingung i=20 steht. Doch bedenken Sie, daß auch andere Vergleiche vorgenommen werden können; etwa, ob ein Suchcode in einem Textstring des Datensatzes vorhanden ist. Dies erweitert die Möglichkeiten einer For-Schleife mit Laufvariable erheblich.

Abbildung 2: Programmierung einer while-Schleife

```
main()
{
  int wert;
  laufvar = 20;
  while(wert <20)
    {
      printf("%d\n",wert);
      wert = wert + 1;
    }
}
```

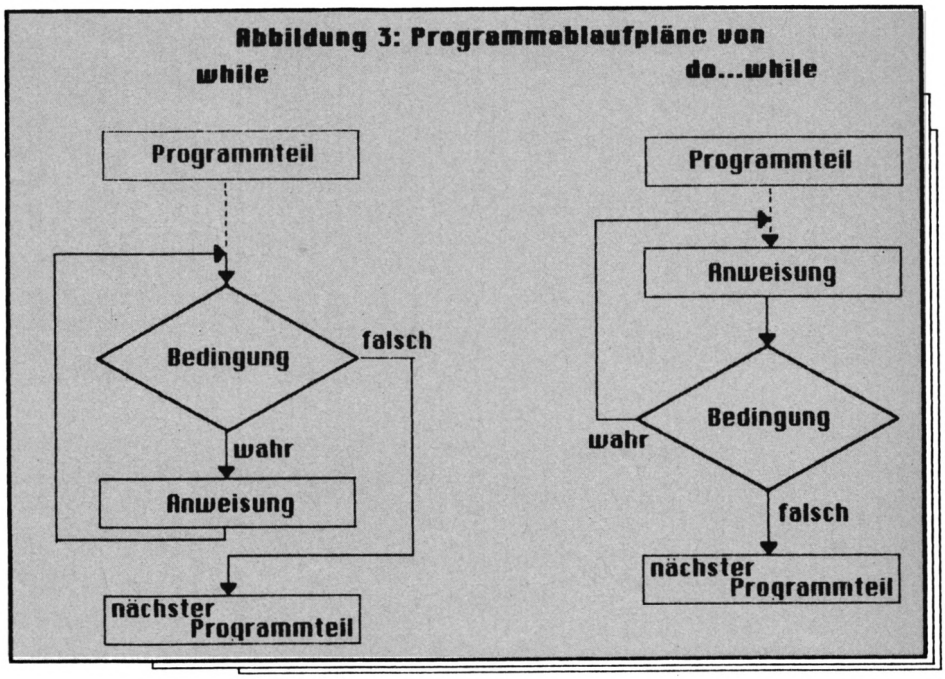
Do...While-Schleife: Erst die Anweisung, dann die Prüfung

Die letzte, heute besprochene Methode einer Schleifenprogrammierung, ist die Do...While-Schleife. Der wichtigste Unterschied zur For- und While-Schleife ist die Position der Schleifenbedingung. Bei Do...While wird die Bedingung erst nach der Ausführung geprüft. Der Quelltext einer derartigen Routine hat folgende Syntax:

```
do
        Anweisung
while(Bedingung);
```

Achten Sie unbedingt auf das Semikolon hinter while, das nur bei der Do...While-Funktion vorkommen darf. Mit den Programmablaufplänen der drei Schleifen schließen wir den heutigen Teil. Anhand der Abbildung 3 ist leicht zu verfolgen, wie sich eine Laufvariable ändert und wann die Abbruchbedingungen geprüft werden.

GS □



Anzeige

Kfm. Komplettpaket

für alle Amstrad- u. Schneider-Computer

Hiermit bestellen wir gegen Nachnahme
für Maschine:

( ) Programm komplett (alle Teile) für **DM 198,–** inkl. dtsch. Handbuch) oder
( ) Finanzbuchhaltung mit Kontenblättern, GuV und Rohbilanz, freier Kontenplan,
( ) Kundenbuchhaltung mit Umsatzstatistik, OP's, Mahnungen, Forderungsliste,
( ) Lieferantenbuchhaltung mit Umsatzstatistik, OP's, Verbindlichkeitsliste,
( ) Lagerbestandsführung mit Umsatzstatistik, Lagerliste, Bestellvorschlagsliste,
( ) Auftragsbearbeitung mit Lieferscheinen, Fakturierung, Kundenrabatt, Auftragsrabatt,
( ) Sofortfakturierung mit Kundenrabatt, Auftragsrabatt, Formular frei aufbaubar,

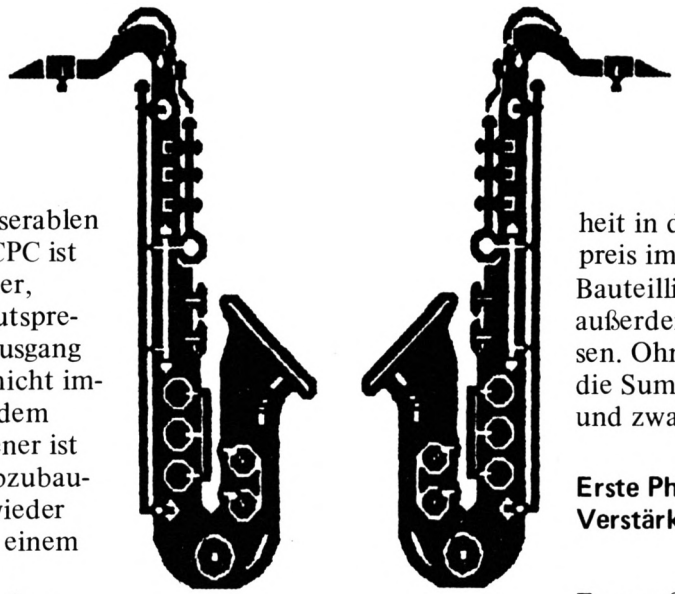
für à **DM 39,90** + 1 x DM 10,– für deutsches Handbuch.

infosystems
SOFTWARE-VERTRIEBS GMBH
Dorrhof 7 · 4419 Laer · Telefon (02554) 1232
DIE HELFENDE HAND IM SOFTWARELAND!

Volle Dröhne

Stereo mit dem CPC

Der Sound des CPC ist – das wissen die Wenigsten – hervorragend. Trotzdem haben Sie sich sicherlich schon über die schlechte Musikuntermalung bei Spielen wie Ghostbusters oder Masters of the Lamps geärgert. Abhilfe schafft unsere Bauanleitung für eine kleine Verstärkeranlage.



Der Grund für die miserablen Tonqualitäten des CPC ist der eingebaute Pieper, fälschlicherweise auch Lautsprecher genannt. Ein Stereoausgang ist zwar vorhanden, aber nicht immer steht der CPC neben dem HiFi-Turm und noch seltener ist man gewillt, die Anlage abzubauen und irgendwo anders wieder anzuschließen. Was bleibt einem da noch übrig?

Wer sich eine Verstärkeranlage kaufen kann, sollte erst die Gerätebeschreibung genau lesen. Die CPC-WELT schlägt bei diesem Thema einen preiswerteren Weg vor: Den Selbstbau, ohne gleich die Funktionsfähigkeit des Rechners aufs Spiel zu setzen. Kein Risiko also, wenn Sie mit etwas Fingerspitzengefühl ausgestattet sind. Sehen Sie sich zunächst den Schaltplan an. Sie wissen dann, worauf Sie besonders achten müssen, und wo die größten Schwierigkeiten liegen.

Sie sehen, daß zwei Schaltungen identisch sind. Diese doppelte Schaltung ist notwendig, weil der Sound-Chip des CPC, der AY-3-8912, ein dreistimmiger Synthesizer-Baustein, der für Telespiele entwickelt wurde, Stereo spielen kann.

Der Soundchip gibt hierbei den ersten Kanal auf dem einen Lautsprecher aus, den zweiten auf dem anderen Lautsprecher und den dritten in der Mitte.

Sie haben richtig gelesen: Stereo mit drei Kanälen. Beim dritten Kanal wird ein Lautsprecher in der Mitte simuliert, indem dessen Ton auf die beiden anderen Kanäle verteilt wird.

Die beiden echten Lautsprecher werden mit unterschiedlichen Frequenzen angesteuert. Daraus folgt, daß Sie jeden Kanal extra verstärken müssen. Falls Sie die beiden Schaltungen in den Kästchen allzu sehr verwirren – dann gibt es im Elektronik-Handel auch Bausätze für Fünf-Watt-NF-Verstärker. Das schraubt zwar geringfügig den Preis dieser Bastel-Einheit in die Höhe, ist aber im Endpreis immer noch tragbar. Die Bauteilliste zeigt Ihnen, was Sie außerdem noch einkaufen müssen. Ohne einen Kopfhörer liegt die Summe nur zwischen zehn und zwanzig Mark.

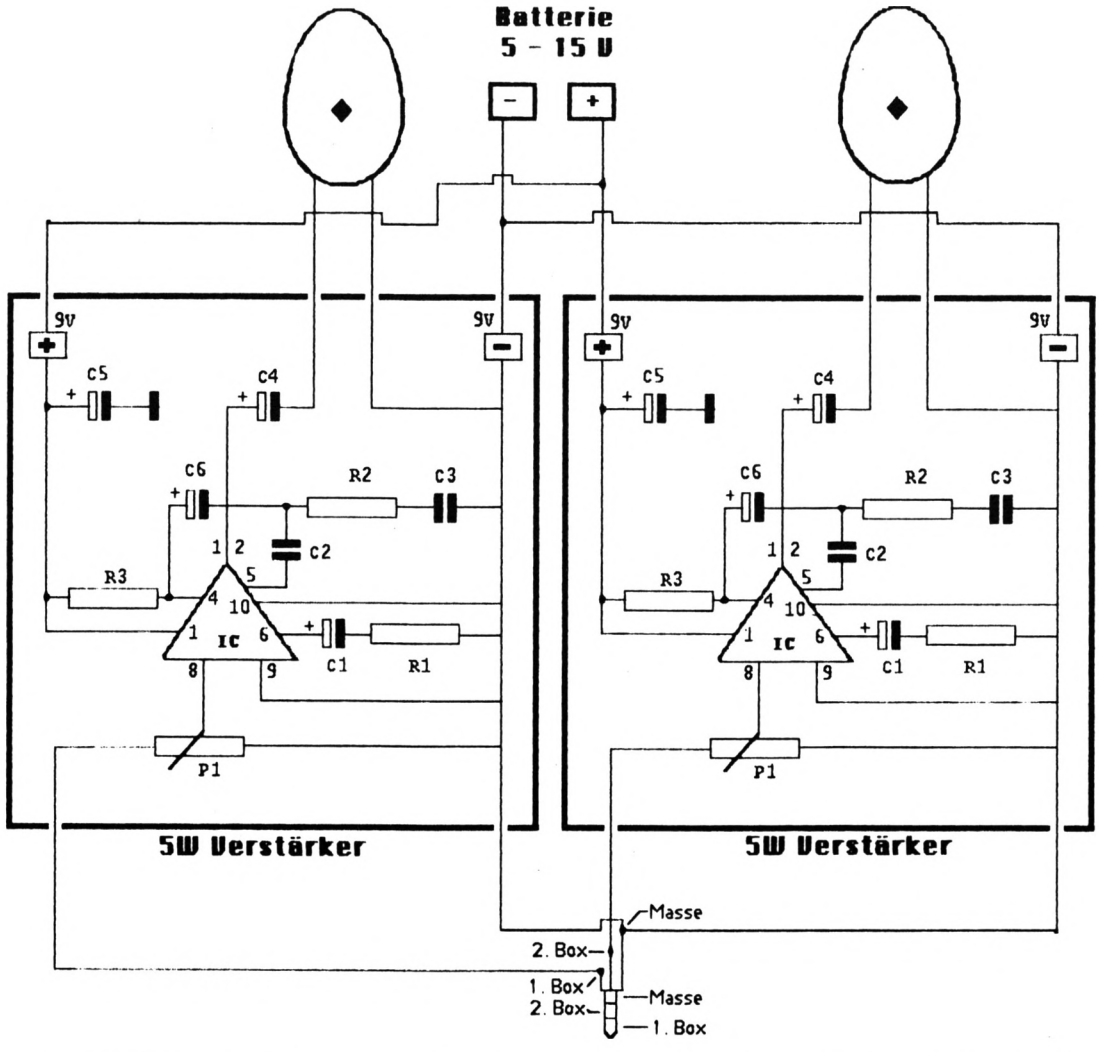
Erste Phase: Verstärkerbau

Es empfiehlt sich, die beiden Verstärker erst einmal einzeln zusammenzubauen und durchzutesten. Erst dann sollten Sie den Rest verkabeln.

Nehmen Sie die beiden Lochrasterplatinen und bauen Sie nach dem Schaltplan die Verstärker auf. Das Trimmpotentiometer P1 sollte nicht direkt auf die Platine kommen, sondern mit Kabeln festgelötet werden. Der fertige Bausatz läßt sich dann später leichter in einem Kasten unterbringen.

Zweite Phase: Endgültige Verkabelung

Alle Bauteile, Bastler wissen dies, sind hitzeempfindlich. Löten Sie deshalb nicht länger als ein oder zwei Sekunden daran; falls nötig,



machen Sie eine Pause zum Abkühlen. Die Warnung gilt insbesondere für den integrierten Schaltkreis IC1.
Nach der Fertigstellung der beiden Verstärker verdrillen Sie die Kabel zu den Boxen und schließen eine Batterie an. Fassen Sie dann das Trimmpotentiometer oder dessen Kabel an. Keine Angst, es ist ungefährlich.
Das Surren aus dem Lautsprecher zeigt an, daß alles in Ordnung ist. Hören Sie bei dieser simplen Testmethode nichts, dann liegt entweder zuwenig Spannung an, oder Sie haben falsch gelötet.
Sollte Ihnen dies ausgerechnet bei den Elektrolydkondensatoren C1, C4, C5, C6 passiert sein, dann heißt dies, daß diese falsch herum eingebaut sind. Dann sind sie mit Sicherheit durchgebrannt. Das ist keine Katastrophe, auch nicht in finanzieller Hinsicht.
Haben Sie diese Probleme gelöst, wagen Sie sich ans Verkabeln. Beginnen Sie am besten am Stecker S1 und löten Sie die Verbindung zwischen ihm und den Verstärkern an. Achten Sie beim Verdrillen der Kabel darauf, daß sich die blanken Enden am Stecker nicht berühren. Jetzt fehlen nur noch die Kabel zur Batterie, und schon ist das Werk vollendet.

Tip für Batterie-Gegner

Zum Schluß noch ein Trick, mit dem Sie sich die kostspieligen Batterien sparen. Ganz nebenbei tun Sie auch etwas für den Umweltschutz, denn Altbatterien gehören ja zu den Sorgenkindern der Müllabfuhr.
Besorgen Sie sich einen 50poligen Stecker, der in den Erweiterungsport des CPC paßt. Hier liegt nämlich die geeignete Spannung (Minus an Pin 49, Plus an Pin 27) für den Verstärker an.
Dies gilt freilich nur für den Fall, daß Ihnen das ewige Abklemmen der Batterien zu umständlich ist und Sie den Sound des CPC so häufig genießen wollen, daß sich die Anschaffung des Steckers lohnt. *Ulrich Lang* □

Oben:
Der Schaltplan des Stereo-Verstärkers für den CPC
Unten:
Für den Aufbau des Gerätes benötigen Sie nur wenige Bauteile

Einkaufsliste

2 kleine Lochrasterplatinen
Einpoliges Kabel
2 * R1 (56 Ohm)
2 * R2 (1 Ohm)
2 * R3 (100 Ohm)
2 * C1 (Elko 470 µF)
2 * C2 (470 pF)
2 * C3 (0,1 F)
2 * C4 (Elko 470 µF)
2 * C5 (Elko 100 µF)
2 * C6 (Elko 100 µF)
2 * P1 (Trimmpotentiometer 100 KOhm)
2 * IC1 (NF-Leistungs-IC a208 TbaB10 BSB)
2 * B (Batterie 5-15 Volt 9 Volt geeignet)
2 * S1 (3,5 mm Stereo-Stecker)
1 * Stereo-Kopfhörer oder (2 * Lautsprecher)

Fill-Routine für Turbo Pascal
Ausfüllen von Flächen
Rekursion macht's möglich

Das lückenlose Ausfüllen großer Flächen ist in manchen Programmiersprachen nicht problemlos zu realisieren. Auf dem CPC 664 und 6128 ist ein Fill-Befehl bereits im BASIC enthalten; Besitzer eines 464 müssen darauf leider verzichten. Doch auch die vielgepriesene Hochsprache Pascal stellt keinen derartigen Befehl bereit.

Mit Hilfe rekursiver Prozeduren kann ein Fill-Befehl in Pascal realisiert werden. Das hier vorgestellte Beispiel wurde in Turbo Pascal geschrieben; eine Anpassung an andere Pascal-Versionen können Sie durch Änderung der benötigten Grafikbefehle vornehmen. Da in Turbo Pascal alle Grafikroutinen die gleichen Namen haben, wie die entsprechenden BASIC-Befehle, wird Ihnen das sicher nicht schwerfallen.

Die Konstruktion einer Fill-Routine bedarf einiger Vorüberlegungen. In der Prozedur Testgrafik wird eine Zeichnung auf dem Bildschirm erzeugt, wie Sie sie in **Abbildung 1** sehen. An dem mit X gekennzeichneten Punkt soll mit dem Ausfüllen begonnen werden.

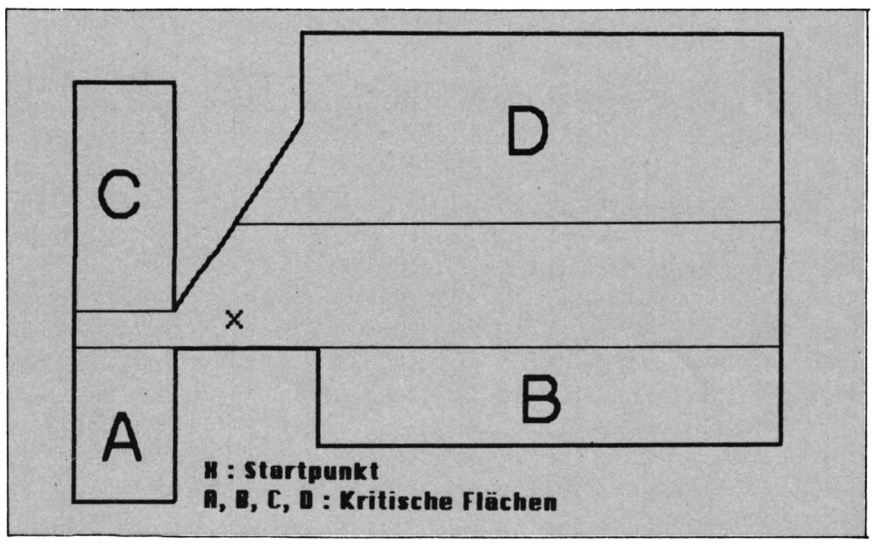


Abbildung 1
Diese Fläche soll ausgefüllt werden

Sie können zunächst vom Startpunkt aus nach oben gehen. In jeder Reihe müssen Sie den linken

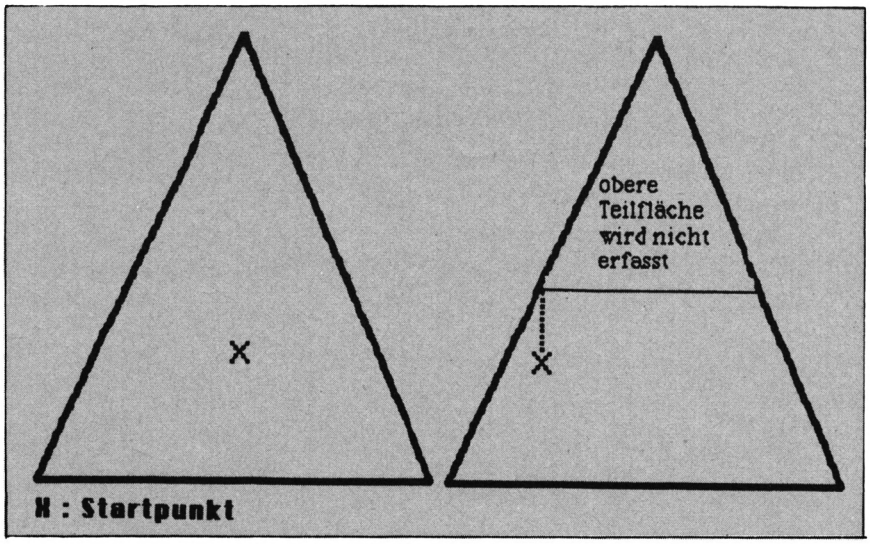


Abbildung 2
Die Wahl des richtigen Startpunktes ist entscheidend

und rechten Grenzpunkt der zu füllenden Fläche ermitteln. Wo sich dieser Grenzpunkt befindet, erfahren Sie, indem Sie jeden Bildpunkt auf seine Farbe hin überprüfen.
Finden Sie einen Punkt in der Farbe des Rahmens, so handelt es sich um die linke, beziehungsweise rechte Grenze der untersuchten Reihe. Da Sie nur die Fläche, nicht aber den Rahmen, in der neuen Farbe auffüllen wollen, müssen Sie von den gefundenen Punkten aus jeweils so weit zurückgehen, bis das Pixel nicht mehr der Rahmenfarbe entspricht.
Diese Aufgabe erledigen die Prozeduren **Finde_links** und **Finde_rechts.** In der Prozedur **Fuelle_Reihe** wird zusätzlich überprüft, ob der Punkt, der als linke Grenze ermittelt wurde, tatsächlich links liegt. Ist das der Fall, so wird zwischen den beiden Endpunkten eine Linie gezogen.
Sie können mit dieser Methode die Grafik vom Startpunkt aus nach oben durchgehen, bis ein Punkt in der Rahmenfarbe oberhalb der gerade gezeichneten Linie gefunden wird. Anschließend können Sie dieselbe Methode nach unten hin anwenden.
Damit ist es bereits möglich, viele Figuren auszufüllen, sofern sich der Startpunkt an einer günstigen Stelle befindet. Setzen Sie ihn etwa in die Mitte eines Dreiecks, so wird die gesamte Figur erfaßt. Plazieren Sie ihn jedoch etwas näher an den Rand, so kann nur ein Teil ausgefüllt werden: Beim Fortschreiten nach oben wird der Rand zu früh entdeckt. Ein Beispiel finden Sie in **Abbildung 2.**
Sie könnten sich mit dieser Lösung zufriedengeben und zum Füllen einer Fläche einfach mehrere Startpunkte verwenden, wenn das Gebilde zu unregelmäßig ist. Aber wie gehen Sie etwa in einem Malprogramm vor, bei dem Sie nicht im voraus wissen, welche Figur auf dem Bildschirm erscheint? Dem späteren Anwender ist es sicher nicht zuzumuten, diese Punkte ständig neu auszuwählen.

Ohne Rekursion geht's nicht

Es muß also eine bessere Lösung gefunden werden. Für die Programmiersprache Pascal bedeutet dies, daß rekursive Prozeduren verwendet werden.
Doch teilen wir das Problem zunächst in kleinere Stückchen auf und betrachten uns den geplanten Algorithmus anhand unserer Beispielfigur. Die erste Prozedur, die wir uns ansehen wollen, ist **Fill_down.** Sie berechnet den Teil der Fläche, der vom Startpunkt aus nach unten verläuft. Dicht unterhalb des Punktes X stößt die Prozedur bereits auf eine Grenze. Die Teilflächen A und B der Grafik würden folglich nicht berücksichtigt. Sie können allgemein davon ausgehen, daß eine Grafik Einbuchtungen haben kann, wie es hier zwischen A und B der Fall ist.
Um dies zu überprüfen, müssen Sie nur jeden Punkt, der sich direkt unterhalb der letzten gezeichneten Linie befindet, abfragen. Besitzt er eine andere Farbe als die des Rahmens, so haben Sie eine noch ungefüllte Teilfläche entdeckt. Die Prozedur **Fill_down** muß nochmals rekursiv aufgerufen werden, um dieses Stück zu erfassen.

Randbedingungen müssen berücksichtigt werden

Soweit ist **Fill_down** einfach zu verstehen. Doch wozu dienen die komplizierte IF-Abfrage innerhalb der REPEAT-Schleife und die gespeicherten Werte **xl_alt** und **xr_alt**?
In unserem Beispiel haben Sie für **Fill_down** keine Funktion. Ihren Nutzen erkennen Sie in **Fill_up.** Der Teil **C** unserer Grafik ist ebenfalls eine kritische Fläche. Hier genügt es jedoch nicht, die letzte gezogene Linie zu prüfen, da sich diese erst an der Grenze zu Teil **D** befindet. Das Problem ist folgendes: Eine Grafik kann Ausbuchtungen oder Seitenteile besitzen, die innerhalb einer Fläche beginnen, welche nach dem bisher bestehenden Algorithmus komplett ausgefüllt würde. Wenn Sie ausschließlich die Begrenzungen des ausgefüllten Teils überprüfen, werden diese Flächenstücke nicht berücksichtigt.
Ein solcher Seitenast einer Fläche ist daran zu erkennen, daß die nächste zum Ausfüllen verwendete Linie kürzer wird, als die vorhergehende. In einem solchen Fall müssen Sie entlang der zuletzt ge-

zeichneten Linie nach Seitenteilen suchen. Da die Grenzen dieser Strecke jedoch bereits vergessen wären, ist ein Zwischenspeicher nötig, in dem sich der Rechner jeweils zwei Linien merkt: die vorhergehende und die aktuelle.
Mit diesem Algorithmus können Sie Seitenteile erfassen, die am Ende oder innerhalb einer gefüllten Fläche angrenzen. Beide Überprüfungen sind sowohl in **Fill_up** als auch in **Fill_down** notwendig. In manchen Grafiken bleiben sie natürlich ohne Ergebnis, etwa in einem Rechteck.
Was aber geschieht, wenn unser Algorithmus in Konflikt mit einer weiteren unangenehmen Eigenschaft von Grafiken gerät, nämlich schrägen Grenzlinien? Dies ist in Abschnitt **D** der Fall. Da die letzte Linie jedoch als Ende einer Fläche erkannt wurde, wird sie, wie oben beschrieben, überprüft, so daß der Rechner die Fläche darüber erkennt. Nachdem dort mit einem rekursiven Prozeduraufruf ein Stück gefüllt wurde, wird wieder ein Grenzpunkt gefunden. Das geht so weiter bis zum Ende der schrägen Fläche.

Effizienz und Kompaktheit

Daraus ist erkennbar, daß unsere Prozedur **Fill** in einem solchen Bereich nicht besonders effizient ist. Sie benötigt dort ein Vielfaches mehr an Speicherplatz und Zeit, als unter „normalen" Bedingungen. Sie müssen jedoch auch die Einfachheit der beiden Routinen **Fill_up** und **Fill_down** bedenken. Bevor Sie sich daran machen, eine wirksamere Lösung zu suchen, sollten Sie abwägen, ob sich der Programmieraufwand lohnt.
Beachten Sie, daß nur schräge Linien, die eine Verengung der Grafik bewirken, diese negativen Folgen haben. Verbreitert sich die Fläche, so bewältigt unser Algorithmus die Aufgabe, ohne den rekursiven Teil verwenden zu müssen.

```
 1 PROGRAM FillTest;
 2
 3 {$I GRAFIK1.INC}
 4 {$I GRAFIK2.INC}
 5
 6 PROCEDURE Testgrafik;
 7   BEGIN
 8     Ink (1,6,6);
 9     Ink (2,26,26);
10     Mode (1);
11     GrafPen (1);
12     GrafMove (10,10);
13     Draw (50,10);
14     Draw (50,90);
15     Draw (90,90);
16     Draw (90,30);
17     Draw (230,30);
18     Draw (230,200);
19     Draw (80,200);
20     Draw (80,160);
21     Draw (50,100);
22     Draw (50,180);
23     Draw (70,180);
24     Draw (70,190);
25     Draw (10,190);
26     Draw (10,10)
27   END;
28
29 PROCEDURE Finde_links (farbe, rahmen : INTEGER;
30                         VAR x : INTEGER; y : INTEGER);
31   BEGIN
32     WHILE (Test (x,y) <> rahmen) AND (Test (x,y) <> farbe) DO
33       x := PRED (x);
34     WHILE (Test (x,y) = rahmen) OR (Test (x,y) = farbe) DO
35       x := SUCC (x)
36   END;
37
38 PROCEDURE Finde_rechts (farbe, rahmen : INTEGER;
39                          VAR x : INTEGER; y : INTEGER);
40   BEGIN
41     WHILE (Test (x,y) <> rahmen) AND (Test (x,y) <> farbe) DO
42       x := SUCC (x);
43     WHILE (Test (x,y) = rahmen) OR (Test (x,y) = farbe) DO
44       x := PRED (x)
45   END;
46
47 PROCEDURE Fuelle_Reihe (farbe, rahmen : INTEGER;
48                          VAR  xlinks, xrechts : INTEGER;
49                          y : INTEGER);
50   BEGIN
51     Finde_links (farbe,rahmen,xlinks,y);
52     Finde_rechts (farbe,rahmen,xrechts,y);
53     IF xlinks < xrechts
54       THEN
55         BEGIN
56           GrafMove (xlinks,y);
57           Draw (xrechts,y)
58         END
59   END;
60
61 {$A-}
62
63 PROCEDURE Fill_up (x, y, farbe, rahmen : INTEGER);
64   VAR
65     xl, xr, xl_alt, xr_alt, yakt, i : INTEGER;
66   BEGIN
```

```
67      yakt := y;
68      xl_alt := 0;
69      xr_alt := 0;
70      REPEAT
71        xl := x;
72        xr := x;
73        Fuelle_Reihe (farbe,rahmen,xl,xr,yakt);
74        yakt := yakt + 2;
75        IF (xl_alt <> 0) OR (xr_alt <> 0)
76          THEN
77            IF xr-xl < xr_alt-xl_alt
78              THEN
79                FOR i := xl_alt TO xr_alt DO
80                  IF NOT (Test(i,yakt-2) IN [farbe,rahmen])
81                    THEN
82                      Fill_up (i,yakt-2,farbe,rahmen);
83        xl_alt := xl;
84        xr_alt := xr
85      UNTIL Test (x,yakt) IN [farbe,rahmen];
86      FOR i := xl TO xr DO
87        IF NOT (Test(i,yakt) IN [farbe,rahmen])
88          THEN
89            Fill_up (i,yakt,farbe,rahmen)
90    END;
91
92  PROCEDURE Fill_down (x, y, farbe, rahmen : INTEGER);
93    VAR
94      xl, xr, xl_alt, xr_alt, yakt, i : INTEGER;
95    BEGIN
96      yakt := y;
97      xl_alt := 0;
98      xr_alt := 0;
99      REPEAT
100       xl := x;
101       xr := x;
102       Fuelle_Reihe (farbe,rahmen,xl,xr,yakt);
103       yakt := yakt - 2;
104       IF (xl_alt <> 0) OR (xr_alt <> 0)
105         THEN
106           IF xr-xl < xr_alt-xl_alt
107             THEN
108               FOR i := xl_alt TO xr_alt DO
109                 IF NOT (Test(i,yakt+2) IN [farbe,rahmen])
110                   THEN
111                     Fill_up (i,yakt+2,farbe,rahmen);
112       xl_alt := xl;
113       xr_alt := xr
114     UNTIL Test (x,yakt) IN [farbe,rahmen];
115     FOR i := xl TO xr DO
116       IF NOT (Test(i,yakt) IN [farbe,rahmen])
117         THEN
118           Fill_down (i,yakt,farbe,rahmen)
119   END;
120
121 {$A+}
122
123 PROCEDURE Fill (x, y, farbe, rahmen : INTEGER);
124   BEGIN
125     GrafPen (farbe);
126     Fill_up (x,y,farbe,rahmen);
127     Fill_down (x,y-2,farbe,rahmen)
128   END;
129
130 BEGIN
131   Testgrafik;
132   Fill (55,96,2,1)
133 END.
```

Mit etwas Aufwand läßt sich eine Fläche auch mit einem Muster füllen. Sie müssen zunächst die Prozedur **Fuelle_Reihe** so abändern, daß nur dann ein Punkt gesetzt wird, wenn die dafür vorgesehene Bildposition bestimmte Kriterien erfüllt. Auf diese Art können Sie etwa ein Schachbrettmuster als Hintergrund verwenden.

Achten Sie jedoch darauf, daß die Überprüfungsroutinen nicht ins Leere greifen, da sich Ihr Programm sonst unwiderruflich verabschiedet. In der Regel müssen Sie mehrere Punkte überprüfen und eine Fläche bereits dann als ausgefüllt anerkennen, wenn einer dieser Punkte von der Hintergrundfarbe abweicht.

Füllen Sie nur geschlossene Flächen aus. Da unsere Routine nicht überprüft, ob beim Zählen der Bildschirmbereich verlassen wird, füllt sie nicht einfach das ganze Bild aus, sondern verläuft sich in einer Endlosschleife.

Wenn Sie nicht ausschließlich mit Pascal arbeiten, werden Sie sich fragen, ob es nicht auch in BASIC eine Möglichkeit gibt, eine derartige Fill-Routine zu schreiben. Für Besitzer eines CPC 464 wäre es das einfachste, einen seiner großen Brüder zu kaufen, der diesen Befehl bereits kennt. Doch es gibt auch eine preiswertere Lösung.

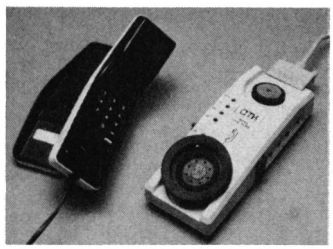
Um effizient zu arbeiten, müssen Sie auf eine Simulation der Rekursion in BASIC verzichten. Sie ist zwar theoretisch möglich, würde aber einen zu großen Aufwand an Zeit und Speicherplatz mit sich bringen. Daher ist die eingangs beschriebene Methode die einzig akzeptable.

Wandeln Sie den Pascal-Text unter Verzicht auf alle rekursiven Aufrufe in ein BASIC-Programm um. Daß Sie dann bei komplizierten Gebilden mehrere Startpunkte angeben müssen, ist zwar ärgerlich, aber kaum vermeidbar. Immerhin sparen Sie dadurch Speicherplatz und gewinnen Zeit.

AE□



CTK-Akustik-Koppler



Zwei Weltneuheiten präsentierte die CTK-Systeme GmbH auf der CeBIT: Der „CTK Adam 1200 Plus" ist der erweiterte und verbesserte Nachfolger des 1200-Baud-Vollduplex-Kopplers CTK Adam 1200. Neu sind unter anderem der zusätzliche 300-Baud-Betrieb, die Unterstützung des Synchronbetriebs mit interner oder externer Uhr und die Verbesserung der Bitfehlerrate. Mit unter einem Bitfehler bei 100 000 Bit Leitungsübertragung ist damit die Übertragungsqualität von Postmodems erreicht. Das Gerät kostet 1 750 Mark.

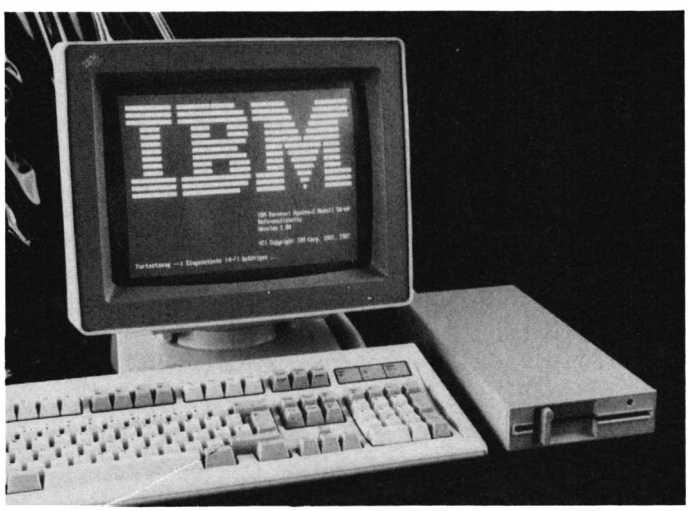
Beim „CTK Adam 2412 High Speed" handelt es sich um den ersten Akustik-Koppler mit 2 400 Baud Übertragungs-Geschwindigkeit. Er besitzt sämtliche Eigenschaften von Adam 1200 Plus und verfügt unter anderem über einen adaptiven Equalizer. Der Koppler kostet 2 190 Mark. ■

IBM-kompatibel

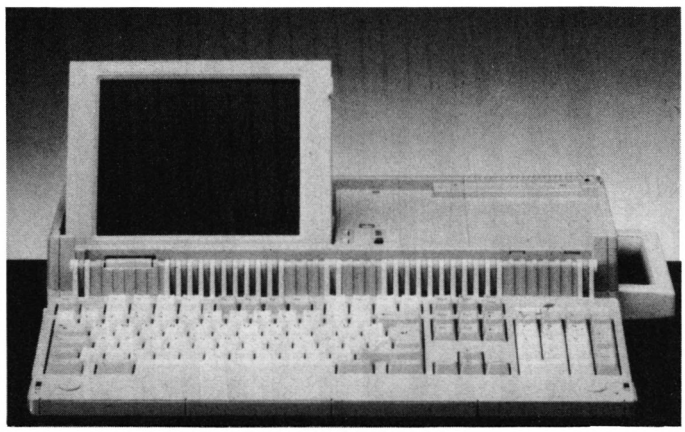
Zwei IBM-Kompatible 5 1/4-Zoll-Disketten-Laufwerke sind seit kurzem auf dem Markt. Beide Versionen, ursprünglich nur für Toshiba Laptops einsetzbar, sind für die IBM PS/2-Serie verfügbar. Sie enthalten ein eingebautes Netzteil mit Störschutzhülle und entsprechenden VDE- und FTZ-Richtlinien. Beide Stationen laufen ebenfalls an den IBM-Modellen 50/60/80/ und sind kompatibel zum Original IBM PS/2-Laufwerk.

Im Unterschied zur 360-KByte-Version ist die 1,2 MByte/360-KByte-Station auch voll kompatibel zum IBM-AT und mit einer automatischen Formaterkennung ausgestattet. Mit ihr können 1,2 MByte- und 360-MByte-Disketten geschrieben und gelesen werden.

Konvertierungen erspart ein zusätzliches 5.25 Zoll Laufwerk an der PS/2 Familie



Amstrad PPC



Deutschland-Premiere feierte auf der CeBIT der PPC 512, Amstrads Portabler. Er ist voll kompatibel zum Industrie-Standard und besitzt eine autonome Batterieversorgung über acht Stunden.

Der PPC ist mit einem 8086-Prozessor ausgestattet, für Spezialanwendungen kann ein optionaler 8087-Coprozessor zugeschaltet werden. Der Computer wiegt knapp elf Pfund. Er wird geliefert mit wahlweise einem oder zwei 3.5-Zoll-Floppy-Laufwerken à 720 KByte, MS-DOS 3.3, MDA- und CGA-kompatibler Grafik, batteriegepufferter Echtzeituhr, parallelen und seriellen Schnittstellen, Modem und Monitor.

Der PPC kostet mit einem Laufwerk 1699 Mark, mit zwei Laufwerken 1999 Mark. ■

Bezugsquelle:
RFI Elektronik GmbH, Dohrweg 63, 4050 Mönchengladbach 1, Telefon 02161/ 6 00 60 ■

Amiga 2500

In zwei Versionen wird der neue Amiga 2500 von Commodore angeboten. Modell 2500 AT ist mit zwei MByte-32-Bit-RAM und einem neuen Set Chips ausgestattet. Er ermöglicht eine Auflösung von 640 mal 512 Punkten im Non-Interlaced Modus. Von der 40-MByte-Festplatte kann sowohl im Amiga- als auch im AT-Modus gebootet werden. Der Amiga AT ist mit einem Bridgeboard, das den Intel-80286-Coprozessor enthält, und zusätzlich 640 KByte Arbeitsspeicher ausgestattet.

Durch das 5 1/4-Zoll- (für PC/AT) und das 3 1/2-Zoll-Diskettenlaufwerk für AT- und Amiga-Formate hat der Anwender Zugang zur Amiga- und MS-DOS-Welt. Auf dem Monitor 1950 können in zwei Fenstern gleichzeitig Text monochrom und Grafik in Farbe dargestellt werden.

Die zweite Version, Amiga 2500 UX, verfügt über eine 100-MByte-Festplatte und einen Vier-MByte-Arbeitsspeicher (32 Bit). Ein Arithmetik-Prozessor 68881 ist ebenfalls vorhanden. Eingebaut ist ein 3 1/2-Zoll-Diskettenlaufwerk, ein Streamer zur Datensicherung ist optional. Mitgeliefert wird das Betriebssystem Unix 5.3. Beide Versionen sind mit dem Prozessor Motorola 68020 ausgestattet, der 400 Prozent mehr Leistung verspricht.

Ebenfalls neu bei Commodore: der 24-Nadel-NLQ-Matrixdrucker MPS 1224 C. Als Ersatz für die bisherigen Bau-

reihen 2000 und 2010 ist er vor allem für den Einsatz an Amiga und AT gedacht.

Der Farbdrucker hat bei Entwurfs-Qualität eine Geschwindigkeit von 220 Zeichen pro Sekunde, bei NLQ druckt er noch 70. Er verfügt über parallele und serielle Schnittstellen und ist kompatibel zu Epson LQ-1500 und IBM 24 XL. ■

Apple-Aufwind

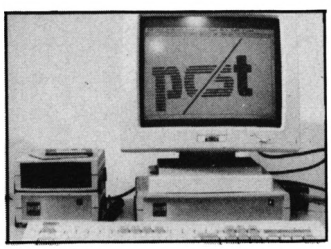
Über das beste Quartalsergebnis seit Bestehen des Unternehmens kann sich derzeit Apple freuen. Der Umsatz im ersten Quartal des Geschäftsjahres 1987/88 beträgt 1,042 Milliarden Dollar, eine Steigerung von 57 Prozent im Vergleich zum Vorjahreszeitraum. Der Gewinn liegt mit 121,4 Millionen Dollar um 108 Prozent höher als im gleichen Quartal des Vorjahres. Im Kalenderjahr 1987 belief sich der Umsatz über drei Milliarden Dollar. „Ein bedeutender Meilenstein für das Unternehmen", so Apple-Chef John HP-kompatiblen Laserdruckern. Das Programm „Angebotswesen Raumplan 3D" übernimmt automatisch die Planungsdaten und fertigt ein Einrichtungsangebot an. „Raumplan 3D" wird von der Nixdorf Computer AG unter dem Namen „Nixas Raumplan 3D" vertrieben. ■

NCR Bilanz

Keinen Grund zum Pessimismus für die EDV-Branche sieht das amerikanische EDV-Unternehmen NCR. Laut Präsident Charles E. Exley erwartet man für 1988 gar neue Umsatz- und Gewinnrekorde. Trotz momentaner Unsicherheit an fast allen Börsenplätzen und Finanzmärkten gebe es keinen Anlaß, die Unternehmenspolitik zu ändern oder Abstriche an beschlossenen Programmen vorzunehmen. Mit einem Umsatzwachstum von 13 Prozent im Jahr 1987 liegt NCR in der Spitzengruppe der zehn umsatzstärksten amerikanischen EDV-Unternehmen. Europa und die BRD würden, so Exley, auch weiterhin ein Schlüsselmarkt für NCR sein. Eine Auslagerung der Produktionskapazitäten in Billiglohn-Länder komme derzeit nicht in Betracht. ■

PCST Mini-AT 286

Als Arbeitsplatz-Rechner für lokale Netzwerke ist der Mini-AT 286 konzipiert. Mit den Maßen 300x400x60 Millimeter läßt sich die kleine Systemeinheit leicht unter Monitoren unterbringen. Sie basiert auf einer 80286 CPU



mit umschaltbarer Taktfrequenz von acht auf 10 MHz. Insgesamt stehen vier 16-Bit-Slots zur Verfügung, wovon zwei durch die Systemplatine und den Grafik-Adapter belegt sind. Die maximale System-Konfiguration erlaubt zwei Festplatten- und zwei Floppy-Anschlüsse, zwei serielle und zwei parallele Schnittstellen, monochrom oder EGA/VGA-Grafikadapter.

Durch den vollen Speicherausbau von einem MByte auf der Systemplatine erreicht die Rechnereinheit mit 16 Bit Buszugriff den vollen Leistungsumfang bei einer Taktfrequenz von 13,5 MHz.

Der Mini-AT 286, entwickelt von der Gautinger PCST GmbH, verfügt über eine deutsche Tastatur mit 102 Tasten. Er kommt ohne Lüftersystem aus und arbeitet geräuschlos. ■

Hoppenstedt Firmen-Datenbank

43000 Unternehmensprofile liefert seit kurzem eine Firmendatenbank auf CD-Rom. Sie ist in erster Linie zugeschnitten auf Direktwerbung, Marktanalysen und Vertriebssteuerung. Das System enthält detaillierte Angaben zu den Unternehmen sowie zu mehr als 130 000 Führungskräften.

Informationen bei:
Hoppenstedt Wirtschaftsdatenbank GmbH
Postfach 4006
6100 Darmstadt 1
Telefon 06151/380-357
oder -282 ■

Computerwiese

Auch auf dem Campingplatz hält der Computer Einzug: Das Programm „Wiese"

IMPRESSUM

CPC-SPECIAL erscheint zweimonatlich in der CA Verlags GmbH, einer Gesellschaft der AKTUELL-Gruppe, Heßstraße 90, D-8000 München 40, Tel. 089/1 29 80 11, Telex: 5214428 cav d.

VERANTWORTLICH FÜR DEN INHALT:
Alwin Ertl

REDAKTION UND STÄNDIGE MITARBEITER:
Dipl.-Ing. Manfred Eibisch, David Krasucki, Torsten Seibt, Gert Seidel, Gerhard Wertenbach

HERSTELLER:
Anton Kult

LAYOUT:
Sonja Anderle

GESCHÄFTSFÜHRER:
Werner E. Seibt

VERANTWORTLICH FÜR ANZEIGEN:
Brigitte Kostic

ANZEIGEN-VERWALTUNG:
ADV-Mediendienste beide: Postfach 101124, 8900 Augsburg 1, Tel.: (0821) 79 04-0, Telekopierer: (0821) 79 04-243, Telex: adv 533502, Teletex: 821887

Es gilt Preisliste Nr. 9 vom 1.6.1988. Media-Unterlagen bitte anfordern.

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:
Postfach 1161, 8044 Unterschleißheim, Tel.: 089/1 29 80 11, Telex: 5214428 cav-d

© 1988 by CAV-GmbH, 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. Jede Verwertung ist untersagt. Namentlich gezeichnete Beiträge unserer Mitarbeiter stellen nicht unbedingt die Meinung der Redaktion dar.

VERTRIEB:
Verlagsunion Wiesbaden

Printed in Germany by ADV, Aindlingerstr. 17–19, 8900 Augsburg 1

Veröffentlichungen gem. § 8,3 Bayr. PrG: Gesellschafter der CAV-GmbH sind Henny Rose Seibt, Kauffrau, zu 26 v.H. und Werner E. Seibt, Journalist, beide Elisabethstraße 1, 8044 Unterschleißheim, zu 74 v.H.

wurde speziell für die Verwaltung von Campingplätzen entwickelt. Es übernimmt unter anderem die Verbuchung von Anmeldungen und Zahlungseingängen, erstellt Meldelisten und Umsatzstatistiken und managt die Reservierungen. „Wiese" ist modular aufgebaut. Damit ist es möglich, nur die wirklich benötigten Teile zu installieren. Die Grundversion kostet 2 350 Mark.

Zu beziehen bei:
Büro für neue Systeme
Oltmannsstraße 34
7800 Freiburg ■

Bibliotheks-katalog auf CD

Ein Pilotprojekt „Optische Speichertechnik" für Bibliotheken hat die Universität Bielefeld mit der Daten-CD gestartet. Zusammen mit dem Hamburger Systemhaus CCS und der Philips Kommunikations Industrie AG will sie die Technologie für die Literatur-Recherche der Bibliotheks-Benutzer einsetzen: Der komplette Bibliothekskatalog wird auf Compact-Disk übertragen und als Datenbank über entsprechende, an PCs angeschlossene, Laufwerke zugänglich gemacht. ■

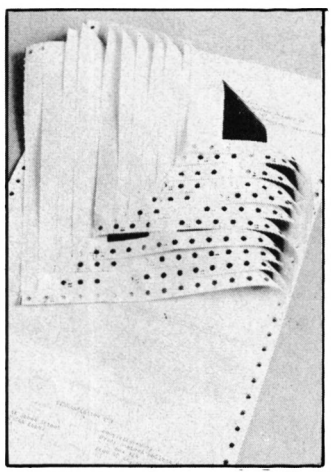
SABREWorks

Eine Büro-Automations-Software namens „SABREWorks" will künftig den Reisebüros effektiveres Arbeiten ermöglichen. Die Arbeitserleichterung besteht unter anderem darin, daß der Reiseberater mit einem einzigen Computer alle Informationen abfragen, Buchungen und Reservierungen vornehmen und dem Kunden sofort ein Angebot machen kann. Die Software wird von SABRE Travel Information Network, einem Unternehmensteil von American Airlines, angeboten und basiert auf Microsoft WORKS.

„SABREWorks" läuft auf IBM-PS/2-Arbeitsstationen, die derzeit in über 600 Reisebüros installiert werden. ■

Daisy M45-Turbo

Einen leistungsstarken Formulardrucker stellt die holländische Firma Daisy vor: Der M 45 Q und verfügt über einen neu entwickelten Druckhammer-Mechanismus. Er kann Dokumente mit bis zu



16 Durchschlägen drucken, die durchschnittliche Druckgeschwindigkeit beträgt 35 cps. Der M 45-Turbo ist kompatibel zu den meisten Mainframe-, Mini- und Mikro-Computern. Er kostet 5 375 Mark. ■

Tron Security Modem

Den Hackern keine Chance läßt das neue Security Modem. Es verhindert das Eindringen in einen Rechner über die frei zugänglichen Telefonlinien, noch bevor der Anrufer im Computer ist:

Das Modem hebt bei einem eingehenden Gespräch den Hörer ab. Im Security Modem muß sich der Anrufer mit einem Passwort legitimieren. Der Modem vergleicht es mit seiner Datenbank und bricht die Verbindung ab. Waren die Passwörter identisch, ruft das Security Modem die zugehörige Nummer zurück. Erst dann wird die Verbindung zum Rechner hergestellt.

Das Sicherheitsmodem wird in den Niederlanden bereits von Regierungsstellen eingesetzt. In der BRD erwartet der Hersteller Tron einen offenen Markt im Bereich der Banken und Versicherungen. Für Banken wird die Kombination mit dem Chipcard-Modem, das für den geschützten Verbindungs-Aufbau verantwortlich ist, vorgeschlagen.

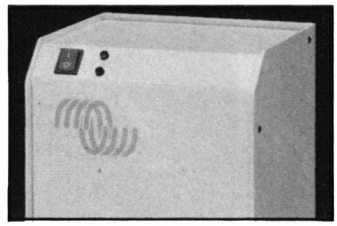
Informationen (nur an Datenschutzbeauftragte und Rechenzentren der Zielgruppen) bei:
Tron Deutschland GmbH
8034 Germering
Telefon: 089 / 84 65 53 ■

Sport-PC

Zum Vertrieb einer einheitlichen Vereinsverwaltungs-Lösung für die im DSB zusammengeschlossenen Sportvereine haben der Deutsche Sportbund (DSB), der Landessportbund Nordrhein-Westfalen und die Philips Kommunikationsindustrie AG eine Kooperationsgemeinschaft geschlossen. Das Komplett-Angebot umfaßt Philips-PC, Drucker und die Vereinsverwaltungs-Software EVAS 1. Die Software unterstützt die Mitglieder- und Übungsleiterverwaltung, die Finanzbuchhaltung und die Kostenstellenrechnung. Um den unterschiedlichen Vereinsgrößen Rechnung zu tragen, werden zwei Komplett-Systeme angeboten. ■

Victron USV

Etwa 60 Prozent aller Verarbeitungsfehler in der EDV werden durch Netzstörungen



verursacht. Den besten Schutz bieten unterbrechungsfreie Stromversorgungen (USV), die die Spannung regulieren, Störwellen und Überspannungen eliminieren und bei Ausfall des Netzes die Verbraucher einige Zeit weiterversorgen.

USV-Anlagen, die ohne Umschaltzeit arbeiten und die Spannung über die gesamte Überbrückungszeit stabil halten, stellt die holländische Firma Victron her. Angeboten werden Anlagen mit Nennleistungen von 400 bis 5000 VA mit einem Klirrfaktor von unter fünf Prozent.

Die kleineren Geräte überbrücken Unterbrechungen bis zu zwölf Minuten. Bei einer Erweiterung der Batterie mit Zusatzmodulen ist eine Dauer von mehreren Stunden möglich. Die größeren Anlagen können, je nach Batterie-Kombination und Belastung, Netzausfälle von zwölf bis 120 Minuten überbrücken. Sie sind auch mit elektronischem oder manuellem Bypass lieferbar.

Vertrieb:
Theile electronic destribution
Beethovenstraße 20
7153 Weissach ■

Wer sucht, der findet

Schnelle Suchverfahren in BASIC

Im zweiten Teil unserer Serie über Sortier- und Suchroutinen haben wir einen Leckerbissen für BASIC-Programmierer: Das binäre Suchen ist enorm schnell. Sonderbar, daß diese Routine so selten verwendet wird.

Stellen Sie sich vor, Sie sollten ein Programm schreiben, das aus den Namen der 67.108.864 Bundesbürger einen ganz bestimmten herausfindet. Vorausgesetzt, Ihr Rechner hat genügend Arbeitsspeicher zur Verfügung: Es ist eine langwierige Sache, den gesuchten Namen mit allen vorhandenen Datensätzen zu vergleichen. Aber es geht auch anders.

Wenn diese Daten sortiert vorliegen, ist es das einfachste Verfahren, den Namen mit dem mittleren Satz zu vergleichen. Ergibt dieser Vergleich, daß in der unteren Hälfte weitergesucht werden soll, so wird die obere Hälfte außer acht gelassen und aus der unteren Hälfte wieder der mittlere Satz genommen. Bei jedem neuen Zugriff auf die Liste halbiert sich also die Anzahl der noch verbleibenden Datensätze. Da dies mathematisch nach der Funktion zwei hoch Zugriff abläuft, ist das Verfahren gerade für große Datenmengen gut geeignet.

Nach spätestens 26 Zugriffen auf die Einwohner-Datei haben Sie Ihren Namen gefunden, da zwei hoch 26 die Zahl 67.108.864 ergibt.

Dieser Zahlenspielerei können Sie entnehmen, daß nicht die ganze Datei Satz für Satz abgesucht und vergleichen werden muß. Wir haben Ihnen dazu ein kleines Programm geschrieben, das Ihnen die Leistungsfähigkeit dieser Such-Art zeigt und mit dem sequentiellen Suchen vergleicht. Tippen Sie das Listing ab und Sie können sich davon überzeugen.

In den ersten Zeilen des Programms werden die Zahlen von eins bis achttausend den Variablen **zahl (b)** zugewiesen. Ihr Rechner braucht dazu Zeit, Sie müssen sich einige Sekunden gedulden.

Zeile 180 stellt alle wichtigen Variablen auf Null, damit Sie auch bei mehrmaligem Eingeben von zu suchenden Zahlen immer die gleichen Startbedingungen haben. Die nächsten Programmschritte sind für die Eingabe der gesuchten Zahl und die Prüfung auf den richtigen Bereich dieser Eingabe zuständig. Es dürfen keine Zahlen, die kleiner als Eins oder größer als 8000 sind, eingegeben werden. Falls Sie Dezimalzahlen eingeben, wird gerundet.

In **Zeile 260** wird der Variablen **timealt** der aktuelle Wert des internen Zeitzählwerkes zugewiesen, da wir wissen wollen, wie schnell die beiden Routinen sind.

Die Suchschleife zum Finden der eingegebenen Zahl umfaßt die **Zeilen 300 bis 350.** Sie ist als WHILE-WEND-Schleife programmiert. Die untere Variable hat beim ersten Durchlauf durch die Zuweisung den Wert eins, die obere den Wert 8000. Dies ist der Zahlenbereich, den wir untersuchen wollen.

In der nächsten Zeile wird lediglich ein Zähler hochgezählt, da wir neben der Zeit auch die Zahl der Schleifendurchläufe und damit die Zahl der Zugriffe wissen wollen.

Anschließend wird die Mitte berechnet. Mit diesem mittleren Datensatz wird in den folgenden Zeilen die eingegebene Zahl verglichen. Sind die beiden Zahlen gleich, so wird in **Zeile 330 timeneu** gesetzt und die Suchzeit als Differenz der beiden Time-Werte berechnet. Nun wird die Bildschirmmeldung, daß die Zahl gefunden wurde, ausgegeben und gleichzeitig die zum Suchen verbrauchte Zeit und die Zahl der Zugriffe gemeldet.
Interessant ist **Zeile 340.** Fällt nämlich der Vergleich in Zeile 330 negativ aus, wurde also die Zahl nicht beim ersten Durchlauf der WHILE-WEND-Schleife gefunden, wird die obere oder die untere Grenze des Suchbereichs verändert.
Nehmen wir an, Sie lassen die Zahl 6.666 suchen. Im ersten Durchlauf liegt die Mitte bei 4.000. Der Vergleich in 340 sagt aus, daß die eingegebene Suchzahl größer ist als die Mitte. Es trifft also die ELSE-Bedingung zu, und die untere Grenze wird auf die alte Mitte zuzüglich eins neu festgesetzt. Das Programm trifft anschließend auf den WEND-Befehl und beginnt den Schleifendurchlauf erneut, da unten kleiner ist als oben.

Wo ist die neue Mitte?

Der Schleifenzähler wird um Eins erhöht und die neue Mitte wird berechnet. Als Bereich kommen jetzt nur noch die Zahlen von 4.001 bis 8.000 in Frage. Jetzt wird erneut verglichen und in Zeile 340 festgestellt, daß die Zahl 6.666 wieder größer ist als die (neue) Mitte. Also wird die aktuelle Untergrenze nochmals angehoben, und das Ganze beginnt von neuem. Nach maximal 13 Zugriffen ist die gesuchte Zahl gefunden. Zur Verdeutlichung schauen Sie sich am besten die **Tabelle** an. Dort sehen Sie, wie sich die Grenzen und die jeweilige Mitte verändern.

Hat das binäre Suchen zum Erfolg geführt, wird anschließend in den **Zeilen 390 bis 430** sequentiell, also der Reihe nach, gesucht. Jetzt muß, da der Startwert bei Eins liegt, 6.666mal verglichen werden, bis die richtige Zahl gefunden wurde. Durch die Ausgabe der verbrauchten Zeit können Sie genau feststellen, wie lange die beiden Routinen benötigen.
Wenn Sie sehr kleine Zahlen suchen lassen, etwa bis 50, wird sich der Vorteil des binären Suchens kaum bemerkbar machen. Aber bei allen anderen Größenordnungen ist diese Methode eindeutig schneller.
Voraussetzung ist jedoch, daß die Liste schon sortiert vorliegt, denn sonst versagt das System. Selbstverständlich können Sie die Vergleiche nicht nur mit Zahlen ausführen lassen. Ebenso schnell werden Strings gefunden. Dazu muß das Beispielprogramm allerdings angepaßt werden.

Auch binäres Sortieren ist möglich

Auch das Sortieren nach dem binären Verfahren ist möglich.
Wenn Sie zum Beispiel eine Adressendatei programmieren und sofort nach der Eingabe eines jeden Satzes den jeweils zuletzt eingegebenen an der richtigen Stelle plazieren, liegt Ihnen immer die sortierte Liste vor. Und dies schon gleich beim Eingeben.
Mit dieser Routine brauchen Sie dann lediglich die Stelle zu finden, an der der neue Satz eingefügt werden muß. Die nachfolgenden müssen um einen Platz nach oben geschoben werden. Damit haben Sie eine Such- und eine Sortierroutine, die beide sehr schnell und leistungsfähig sind. *ME* □

```
10 'Listing 1
20 '***************
30 'Initialisierung
40 '***************
50 MODE 2
60 PRINT"Bitte einen Moment warten, die
zu suchenden Zahlen werden erzeugt"
```

Zugriff	Untere Grenze	Obere Grenze	Mitte	Suchzahl <oder>
1	1	8000	4000	>
2	4001	8000	6000	>
3	6001	8000	7000	<
4	6001	6999	6500	>
5	6501	6999	6750	<
6	6501	6749	6625	>
7	6626	6749	6687	<
8	6626	6686	6656	>
9	6657	6686	6671	<
10	6657	6670	6663	>
11	6664	6670	6667	<
12	6664	6666	6665	>
13	6666	6666	6666	=

Beispiel:
Ablauf der Suche der Zahl: 6666
Erforderliche Zugriffe: 13

```
70 '****************************
80 'Einlesen von 8000 Variablen
90 '****************************
100 DIM zahl(8000)
110 zae=1
120 WHILE b<8001
130 zahl(b)=b:b=b+1
140 WEND
150 '******************
160 'Startwerte setzen
170 '******************
180 zahl=0:zeit=0:unten=1:oben=8000:zae=
0:mitte=0:such=0:timealt=0:timeneu=0:a=F
RE("")
190 PRINT"Zu suchende Zahl eingeben (Ber
eich 1-8000)"
200 PRINT"Programmabbruch durch Eingabe
von 0"
210 INPUT such
220 IF such=0 THEN END
230 IF such<1 THEN PRINT"Zahl zu klein":
PRINT"Mindestens 1!":GOTO 210
240 IF such>8000 THEN PRINT"Zahl zu gros
s":PRINT"Hoechstens 8000!":GOTO 210
250 such=CINT(such)
260 timealt=TIME
270 '****************
280 'Binaeres Suchen
290 '****************
300 WHILE unten<=oben
310 zae=zae+1
320 mitte=(unten+oben)\2
330 IF such=zahl(mitte) THEN timeneu=TIM
E:zeit=timeneu-timealt:PRINT "Die zu suc
hendende Zahl";zahl(mitte);"wurde mit";z
ae;"Zugriffen gefunden!":PRINT "Zeit in
Sekunden:";ROUND(zeit/300,2):PRINT CHR$(
7);:PRINT"Und jetzt wird sequentiell ges
ucht":GOTO 390
340 IF such<zahl(mitte) THEN oben=mitte-
1 ELSE unten=mitte+1
350 WEND
360 '**********************
370 'Sequentielles Suchen
380 '**********************
390 zeit=0:zae=0:timealt=0:timeneu=0:a=F
RE("")
400 timealt=TIME
410 FOR i=1 TO 8000
420 IF such=zahl(i) THEN timeneu=TIME:ze
it=timeneu-timealt:PRINT"Die zu suchende
nde Zahl";zahl(i);"wurde mit";i;"Zugriff
en gefunden!":PRINT "Zeit in Sekunden:";
ROUND(zeit/300,2):PRINT CHR$(7);:PRINT:G
OTO 180
430 NEXT i
```



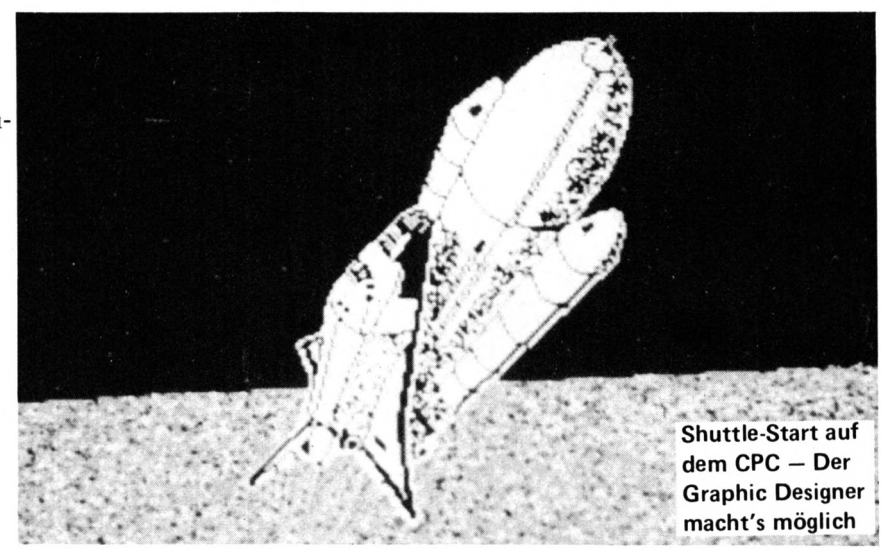

Nur noch wenige Exemplare beim Verlag!

CPC-Welt Hotline!
Jeden Montag
15 - 19 Uhr
Tel.: 089/18 40 24

CPC Graphic Designer

Der GRAPHICS DESIGNER ist ein universelles Programm zum Erstellen und Bearbeiten von Bildschirmgrafiken im Mode 1 und 2. Es ist ein Leckerbissen für alle CPC-Besitzer. Das Abtippen lohnt sich; noch besser ist natürlich, Sie bestellen die Softbox. Dann sind Sie sicher, daß das Programm läuft; außerdem sparen Sie viel Zeit und das mühsame Fehlersuchen.

Das Programmpaket GRAPHICS DESIGNER besteht aus zwei Teilen, die unabhängig von einander benutzt werden können: dem Designer und dem Editor.



Shuttle-Start auf dem CPC – Der Graphic Designer macht's möglich

I. Der Designer

Der Designer ist das Hauptprogramm. Mit ihm werden die Grafiken erstellt. Er stellt alle wichtigen Grundfunktionen zur Verfügung: ZOOM, CIRCLE, ELLIPSE, BOX, STREAMS, TEXT, INK, LINE, FILL, DRAW, CLG, TRIANGLE.
Zusätzlich enthält er noch die Funktion TOOLS sowie einige Ausschnitt- und Ganzbildschirm-Operationen. TOOLS besteht aus drei Teilen: PEN, SPRAY und MULTISPRAY.
Zu den Ausschnitt-Funktionen gehören MAGNIFY, SHRIKE, MIRROR, TURN, SAVE-PART und LOAD-PART. Die Ganzbildschirm-Operationen des GRAPHICS DESIGNER sind zum Teil einzigartig für den Schneider CPC. Es sind die Funktionen BEND/WAVES, MODE, INVERT und MERGE. Einige Befehle sind auch direkt über die Tastatur einzugeben: UNDO, SAVE-UNDO und CURSOR-MODE.
Der Designer wird mit RUN "GRAPH" gestartet und fragt gleich zu Beginn nach der Nummer des Zeichensatzes. 15 Sätze sind auf der Diskette der Softbox bereits enthalten. Wenn Sie die Listings jedoch lieber abtippen wollen, so stehen sie Ihnen ebenfalls zur Verfügung. Die entsprechenden Datalader sind als Listing abgedruckt. Also dürfen Sie, solange keine weiteren Zeichensätze definiert worden sind, nur

die Zeichensätze eins bis 15 aufrufen.
Weitere – maximal 26 – können mit dem EDITOR erstellt werden. Nach der Eingabe der Nummer, die mit ENTER abgeschlossen werden muß, wird das Programm geladen und meldet sich mit dem Hauptmenü. Nun kann die gewünschte Funktion mit Joystick oder Cursortasten angewählt und durch FEUER/ENTER bestätigt werden.

Die Funktionen

I. 1. Die Tastatur-Funktionen

Gesteuert werden kann das Programm entweder mit Joystick oder mit den Cursortasten + ENTER. Einige Funktionen können jedoch nur über die Tastatur eingegeben werden. Dies sind:
UNDO: Durch Druck auf die DEL-Taste werden alle Operationen der aktuellen Funktion rückgängig gemacht.
SAVE-UNDO: Durch Betätigen der COPY-Taste wird der Bildschirminhalt gegen Löschen durch UNDO gesichert.
CURSOR-MODE: Durch das Drücken von TAB wird der Darstellungsmodus zwischen einem Pfeil und einem Fadenkreuz umgeschaltet.
PEN: Mit SHIFT + Cursortasten/Joystick hoch/runter wird der aktuelle Zeichenstift für alle Standard-Funktionen gesetzt. Der Cursor und der Border nehmen ebenfalls immer diese Farbe an.
RETURN: Mit ESC kommt man aus allen Funktionen wieder ins Hauptmenü. Ausnahmen sind SAVE, SAVE-PART und TEXT, diese Funktionen werden durch ENTER verlassen.

I. 2. Die Standard-Operationen

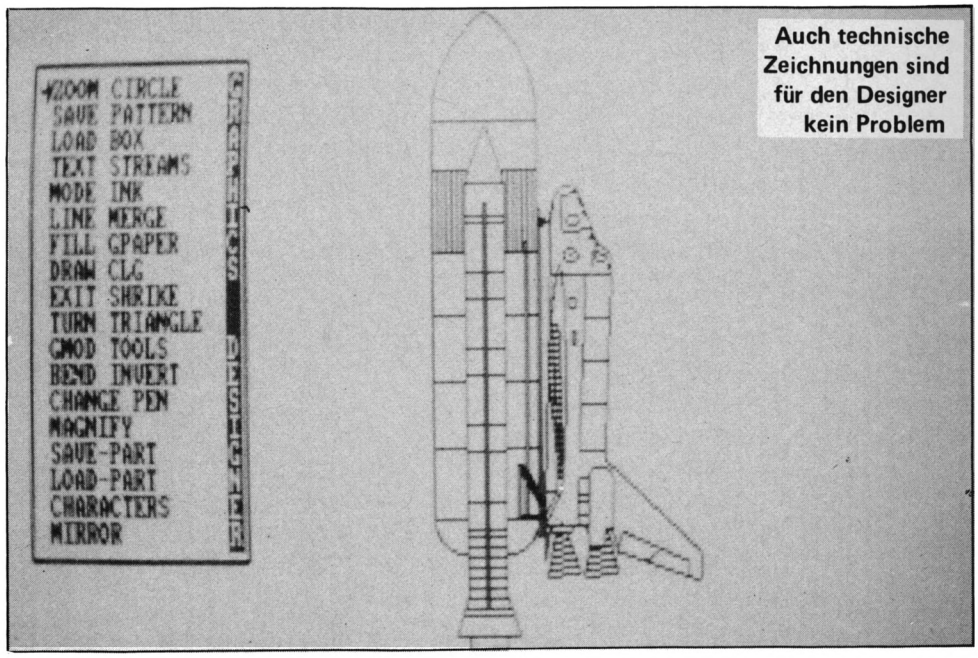
Der GRAPHICS DESIGNER enthält neun Standard-Funktionen. Sie sind alle aus dem Hauptmenü anzuwählen, CIRCLE und ELLIPSE sind jedoch zu CIRCLE zusammengefaßt.
ZOOM: Nach dem Anwählen dieser Funktion erscheint auf dem Bildschirm ein 40/80 (je nach Bildschirmmodus) * 24 Pixel großes Rechteck, das wie der Cursor über den Bildschirm bewegt werden kann. Durch ENTER/FEUER wird der im Rechteck liegende Bildausschnitt achtfach vergrößert auf dem ganzen Bildschirm dargestellt und kann pixelweise editiert werden. Die Steuerung im vergrößerten Modus erfolgt genauso wie im Standard-Modus. Der Rücksprung in dem Standard-Modus erfolgt durch ESC, die Veränderungen werden jetzt im richtigen Maßstab dargestellt. Das Rechteck kann zu anderen Stellen des Bildschirmes bewegt werden. Drückt man FEUER/ENTER, so wird der editierte Ausschnitt an diese Stelle kopiert.
CIRCLE: Diese Funktion verzweigt nach ihrem Aufruf in ein Untermenü, in dem festgelegt werden kann, ob man einen Kreis (CIRCLE) oder eine Ellipse zeichnen möchte.
Hat man sich für CIRCLE entschieden, so braucht man nur den Mittelpunkt des Kreises und einen beliebigen, auf dem Kreisbogen liegenden Punkt mit FEUER/ENTER festzulegen. Danach wird in der aktuellen Zeichenfarbe ein Kreis um den Mittelpunkt durch den Kreispunkt gelegt.
ELLIPSE: Der GRAPHICS DESIGNER gehört zu den wenigen Programmen, die in der Lage sind „schiefe" Ellipsen zu zeichnen. Daher ist deren Konstruktion etwas schwieriger. Für diese Funktion muß zuerst der Mittelpunkt, dann der X- und danach der Y-Radius festgelegt werden. Anschließend ist die Abweichung von der X-Achse (Winkel) festzulegen. Dazu wird der Cursor auf einer Geraden durch den Mittelpunkt gesetzt und zum Schluß der Winkel durch den Y-Abstand zu dieser Geraden festgelegt.
BOX: Diese Funktion zeichnet auf einfache Weise ein Rechteck. Zuerst muß mit FEUER/ENTER ein Rechteck, das mit dem Cursor in die gewünschte Form gebracht werden kann. Nach Druck auf FEUER/ENTER werden das Rechtwerden kann. Nach Druck auf FEUER/ENTER wird das Rechteck endgültig gezeichnet und der Cursor wieder dargestellt.
TEXT: Mit dieser Funktion lassen sich Texte im aktuellen Zeichensatz (vergleiche I.3. CHARACTERS) auf den Bildschirm ausgegeben. Dazu erfolgt eine Textabfrage im aktuellen Zeichensatz. Es dürfen alle Tastaturzeichen verwendet werden, bei Verwendung von ',' muß der Text jedoch in Anführungszeichen eingeschlossen werden.
Nachdem der Text eingegeben und die Eingabe durch ENTER bestätigt wurde, fragt das Programm nach einem Vergrößerungsfaktor, der eine ganze Zahl größer Null sein sollte. Der Text wird um diesen Faktor vergrößert auf dem Bildschirm ausgegeben. Der linke obere Anfangspunkt des Textes muß mit FEUER/ENTER festgelegt werden, danach erfolgt die Text-Ausgabe. **Achtung:** Die TEXT-Funktion darf während der Eingabe-Routine nicht mit ESC abgebrochen werden.
Der Rücksprung erfolgt durch Drücken von ENTER ohne weitere Eingaben. Sobald das Programm zum Grafikbildschirm zurückkehrt, wird wieder mit ESC ein Rücksprung ausgelöst.
LINE: Diese Funktion dient zum Zeichnen einer Linie zwischen zwei festgelegten Punkten. Hierzu wird zuerst der Startpunkt der Linie durch Druck auf SPACE (Leertaste) festgelegt, daraufhin der Endpunkt mit FEUER/ENTER. Es muß nicht unbedingt wieder ein neuer Startpunkt festgelegt werden. Der Endpunkt der letzten Linie wird bis zum Verlassen der Funktion als Startpunkt gewählt, sofern nicht ein neuer Startpunkt mit SPACE definiert wird.
STREAMS: Diese Funktion arbeitet ähnlich wie LINE, nur wird

hier nicht der Endpunkt der Linie als neuer Anfangspunkt definiert, sondern der Anfangspunkt bis zu seiner Neufestlegung beibehalten.
DRAW: Dies ist die Option zum Freihand-Zeichen. Das heißt, Sie können mit einem Cursor wie mit einem Stift beliebige Figuren auf dem Bildschirm zeichnen. Die Option arbeitet sehr einfach: Durch Druck auf FEUER/ENTER wird ein Punkt in der aktuellen Zeichenfarbe gesetzt.
TRIANGLE: Mit Triangle kann leicht ein Dreieck erstellt werden. Dazu werden die drei Eckpunkte mit ENTER/FEUER festgelegt. Das Programm zeichnet dann automatisch ein Dreieck durch diese Punkte.

I. 3. Zeichensatz- und Pattern-Operationen

Als zweite große Befehlsgruppe nach den Standardoperationen enthält der GRAPHICS DESIGNER auch sechs Pattern- und eine Zeichensatz-Operation. Von diesen sind wiederum drei – PEN, SPRAY und MULTISPRAY – unter TOOLS zusammengefaßt. Alle übrigen Optionen lassen sich direkt vom Hauptmenü aus anwählen. Alle Pattern-Operationen arbeiten nur im Mode 1. Im Mode 2 werden die entsprechenden Funktionen einfarbig durchgeführt, wie dies im Mode 1 nach dem Anwählen des SOLID-Symbols der Fall ist.
PATTERN: Bei allen Pattern-Operationen lassen sich, wie der Name schon andeutet, sogenannte Patterns verwenden. Sie sind nichts anderes als Muster, durch die sich zum Beispiel beim Füllen von Flächen interessante Effekte erzielen lassen.
Beim GRAPHICS DESIGNER entspricht ein Pattern einem Textzeichen. Zum Auswählen eines Patterns dient diese Funktion.
Der GRAPHICS DESIGNER enthält in der Standardausrüstung 91 solcher Muster. 26 davon sind die Großbuchstaben des Alphabets, eines ist für SOLID FILL, ein Füllen ohne Muster, vorgesehen. Die übrigen sind verschiedene Patterns, also spezielle Füllmuster. Der gesamte Patternsatz kann mit dem EDITOR verändert werden (vergleiche II. 1. FONT DESIGNER). Nach Aufruf dieses Menüpunktes werden alle Patterns auf dem Bildschirm dargestellt und können mit dem Pfeil angewählt werden. Dieser läßt sich mit Joystick/Cursortasten rechts/links bewegen. Bestätigt wird mit FEUER/ENTER. Danach stellt das Programm das Pattern mehrfach auf dem Bildschirm dar, da sich am kleinen Zeichen oft noch nicht die endgültige Form des Musters erkennen läßt.
Hierauf erfolgt eine weitere Abfrage. Beantworten Sie diese mit NEIN, so springt das Programm in die Auswahl zurück, andernfalls wird in das Hauptmenü gesprungen.
GRAPER: Diese Funktion setzt die Hintergrundfarbe für Operationen mit Verwendung von Patterns. Da ein Pattern aus einer Zeichenfarbe (im Auswahlmenü der PATTERN-Funktion Pen 1) und einer Hintergrundfarbe (im Auswahlmenü Pen 0) besteht, muß diese ebenfalls definiert werden. Auf die Frage CODE? muß einer der vier Zeichenstifte angegeben werden (null bis drei). Danach erfolgt ein Rücksprung ins Hauptmenü.
FILL: Diese Funktion dient zum Füllen einer geschlossenen Fläche. Man bewegt den Cursor in die Fläche hinein und bestätigt mit FEUER/ENTER. Jetzt wird sie mit dem aktuellen Pattern gefüllt.
Die aktuelle Farbe dient als Pen- und die mit GRAPER eingestellte Hintergrundfarbe als Paperfarbe des Musters. Beide dürfen jedoch nicht mit der bereits vorhandenen Farbe der Fläche übereinstimmen, sonst wird die Funktion nicht ausgeführt.
TOOLS: Unter TOOLS sind drei Funktionen zusammengefaßt: PEN, SPRAY und MULTISPRAY.
SPRAY: Dies ist die Standard-Sprühdosenfunktion. Wie bei den anderen TOOLS-Funktionen wird auch hier erst die Zeichengröße erfragt (WIDTH). Sie entspricht bei SPRAY der dreifachen Anzahl des eingegebenen Faktors in Pixeln. In diesen Bereich oberhalb des Cursors setzt SPRAY bei Druck auf FEUER/ENTER zufällig Punkte.
Solange nicht die SOLID-Funktion (ausgefülltes Pattern) angewählt ist, werden diese Punkte dem aktuellen Pattern entsprechend gesetzt. Dabei verhalten sich Pen- und Paperfarbe wie bei FILL, nur ist die SPRAY-Funktion auf jedem Hintergrund anwendbar.
PEN: Diese Funktion entspricht weitgehend der LINE-Option, nur ist die Linienbreite variabel. WIDTH fragt sie in reellen Bildpunkten ab. Die gezeichnete Linie entspricht außerdem immer dem aktuellen Pattern. Das heißt, die Linie wird im entsprechenden Muster dargestellt. Die Farbeinstellung ist identisch mit der von SPRAY.
MULTISPRAY: Ähnelt der SPRAY-Option, jedoch werden die Farben nicht dem Muster entsprechend, sondern zufällig gesetzt, es entsteht also eine zufällige Farbverteilung. Die aktuelle Penfarbe hat hierauf keinen Einfluß.
CHARACTERS: Diese Option dient zum Anwählen eines Zeichensatzes für die TEXT-Funktion. Dazu werden die drei möglichen Zeichensätze, Standardzeichensatz und ein oder zwei Zeichensätze aus dem aktuellen geladenen Set, in einem Menü angezeigt und können mit Joystick/Cursortasten angewählt werden. Der gewählte Zeichensatz wird nur für die TEXT-Funktion und nicht für die Ausgabe im Menü verwendet. Bei den zugeladenen Zeichensätzen werden die Groß- und Kleinbuchstaben getrennt behandelt. Es kann also vorkommen, daß ein Set, das nur einen

Zeichensatz – allerdings Groß- und Kleinbuchstaben – enthält, die Groß- und Kleinbuchstaben als zwei getrennte Zeichensätze ansieht.
Innerhalb der CHARACTERS-Funktion gibt es auch die Option CHANGE. Sie dient dazu, einen anderen externen Zeichensatz – statt des momentan aktiven – in den Speicher zu laden. Auf der Diskette sind bereits 18 solcher Sets enthalten, bis zu acht eigene können mit dem EDITOR erstellt werden (vergleiche II. 1. FONT DESIGNER). Zum Laden eines solchen Sets wählt man CHANGE an und gibt dann die Nummer des entsprechenden Zeichensatzes an (zwischen eins und 26). Ein Rücksprung aus dieser Funktion geschieht nicht mit ESC, sondern mit ENTER ohne vorherige Eingabe des Sets.



Auch technische Zeichnungen sind für den Designer kein Problem

I. 4. Die Disketten-Operationen

Der GRAPHICS DESIGNER enthält fünf spezielle Disketten-Operationen. Zwei davon – LOAD-PART und SAVE-PART – werden später noch ausführlich erklärt (vergleiche I. 5. LOAD-PART und SAVE-PART). Bei allen Disketten-Operationen sollte sich eine formatierte Diskette im aktiven Laufwerk befinden, andernfalls erfolgt ein Rücksprung ins Hauptmenü.
Gelegentlich kann es außerdem vorkommen, daß das Programm beim Aufruf oder während der Arbeit ins Hauptmenü zurückspringt. Dies liegt daran, daß im GRAPHICS DESIGNER nur sehr wenig freier Speicherplatz zur Verfügung steht und der Computer dann eine Garbage Collection durchführt. In diesem Fall muß die Funktion noch einmal aufgerufen werden.
SAVE: Diese Funktion speichert den Inhalt des Grafikbildschirms zusammen mit dem Bildschirmmodus und den einzelnen Farben auf Diskette. Bei ihrem Aufruf überprüft der Computer automatisch, ob noch genügend freier Speicherplatz auf der Diskette im aktiven Laufwerk vorhanden ist. Ist dies nicht der Fall, so wird ins Hauptmenü zurückgesprungen, andernfalls wird nach dem Namen der zu speichernden Datei gefragt. Ein Rücksprung erfolgt mit ENTER, auf keinen Fall mit ESC. Gespeicherte Bilder können unter BASIC einfach mit LOAD"name" geladen werden.
LOAD: Diese Funktion dient zum Laden der gespeicherten Bilder des GRAPHICS DESIGNER. Das Programm liest alle Bilder auf einer Diskette ein und stellt sie in einem Auswahlmenü dar. Das gewünschte File muß nur mit Joystick/Cursortasten angewählt und mit FEUER/ENTER bestätigt werden.
Der GRAPHICS DESIGNER installiert automatisch den richtigen Bildschirm-Modus und wählt die richtigen Farben an. **Achtung:** Normalerweise dürfte es kein Problem sein, Bilder aus anderen Grafikprogrammen mit dem GRAPHICS DESIGNER zu laden. Meistens sind dazu nur zuerst der entsprechende Mode einzustellen und hinterher die entsprechenden Farben zuzuordnen (vergleiche I. 7. INK).
Bei Bildern aus anderen Programmen, die ebenfalls die Bilddaten mitspeichern, dies aber in einem anderen Format tun, kann es jedoch vorkommen, daß der GRAPHICS DESIGNER sich „aufhängt", also das Bild im Mode null darstellt und/oder in einer Endlosschleife verschwindet. In diesem Fall muß das Bild mit folgendem kleinen BASIC-Programm bearbeitet werden.

```
10 MODE <Modus des Bildes>
20 LOAD"<Name des Bildes>",
   &c000
30 FOR n=&FFFB to &FFFF
40 POKE n,0
50 NEXT
60 SAVE"<NAME>",b,&C000,
   &4000
```

In den meisten Fällen müßte das Bild dann auch vom GRAPHICS DESIGNER aus zu laden sein.
MERGE: Die MERGE-Funktion entspricht weitgehend LOAD, nur daß das alte Bild nicht gelöscht, sondern über XOR mit der neuen Datei verknüpft wird. Hierdurch ergeben sich interessante Effekte wie etwa eine dem Hintergrund angepaßte Schrift. Der Bildschirmmodus und die Farben werden nicht verändert, so daß sich der Bildschirm schon vor dem Aufruf der Funktion im richtigen Modus befinden sollte.

Allerdings kann es vorkommen, daß das Lesen des Bildes schon vor dessen Ende abgebrochen wird. Das liegt an einem Betriebssystem-Fehler, der den MERGE-Befehl im BASIC des CPC 464 mit Floppy ebenfalls unwirksam macht.

I. 5. Die Ausschnitt-Funktionen

Der GRAPHICS DESIGNER enthält mehrere Funktionen, mit denen einzelne Bildschirmausschnitte bearbeitet werden können. Eine davon – ZOOM – wurde bereits zu Anfang erklärt (vergleiche I. 2. ZOOM). Für alle Optionen werden die zu bearbeitenden Ausschnitte gleich definiert: Zuerst muß die linke untere Ecke des Ausschnittes mit FEUER/ENTER bestimmt werden. Jetzt öffnet sich ein Rechteck, das den zu bearbeitenden Ausschnitt festlegt. Der Ausschnitt wird mit FEUER/ENTER bestätigt.
SHRIKE: Dies ist die Verkleinerungs-Funktion. Das Programm fragt zuerst nach dem Faktor, um den verkleinert werden soll. Dieser muß ganzzahlig sein und sollte nicht über zehn liegen, da sonst die Verkleinerung kaum noch zu erkennen sein dürfte.
Danach kehrt das Programm zur Grafikseite zurück und der zu verkleinernde Ausschnitt kann in der oben beschriebenen Weise definiert werden. Danach erscheint wieder der Cursor, und die linke untere Ecke der Stelle, an die der verkleinerte Ausschnitt gesetzt werden soll, muß mit FEUER/ENTER festgelegt werden. Jetzt kann ein neuer Ausschnitt um denselben Faktor verkleinert werden, indem man wie oben verfährt. Will man den Faktor ändern, muß man ins Hauptmenü zurückkehren.
TURN: Mit dieser Funktion kann ein Ausschnitt um einen Winkel zwischen null und 90 Grad gedreht werden. Nachdem der Ausschnitt wie oben definiert wurde, muß ein Punkt eingegeben werden, um den er gedreht werden soll. Es folgt der Winkel relativ zur X-Achse. Drückt man, statt einen Winkel festzulegen, nur FEUER/ENTER, so wird als Winkel 90 Grad angenommen.
MAGNIFY: Dieser Punkt ist die Umkehrung zu SHRIKE, also eine Vergrößerungsfunktion. Die Steuerung ist identisch mit der von SHRIKE, nur daß der Ausschnitt an der festgelegten Stelle um den eingegebenen Faktor vergrößert ausgegeben wird.
MIRROR: Diese Funktion spiegelt einen Ausschnitt an der Y-Achse. Die Steuerung ist hier wieder identisch mit SHRIKE und MAGNIFY, lediglich der Faktor muß nicht eingegeben werden. Der Ausschnitt wird an der festgelegten Stelle gespiegelt wiedergegeben.
CHANGE PEN: Mit diesem Befehl ist es möglich, eine Farbe in einem Ausschnitt durch eine andere zu ersetzen.
Zuerst muß der Ausschnitt definiert werden. Dies geschieht wie bei den übrigen Ausschnittfunktionen, nur darf er hier größer sein. Die Farbe des Borders beim Bestätigen mit FEUER/ENTER entspricht der zu ersetzenden Farbe. Danach erscheint wieder der Cursor auf dem Bildschirm und die zweite Farbe läßt sich anwählen. Während dieses Vorgangs darf nicht auf die SPACE-Taste gedrückt werden, da sonst der Ausschnitt verschoben würde. Nach dem Festlegen werden im Ausschnitt alle Punkte mit Farbe eins durch Punkte mit Farbe zwei ersetzt.
SAVE-PART: Mit dieser Funktion lassen sich definierte Ausschnitte auf Diskette speichern. Nachdem ein Ausschnitt festgelegt worden ist, springt das Programm in eine Abfrageroutine, in welcher der Name des Ausschnittes ohne Extension angegeben werden muß. Er wird dann unter dem Namen + Extension .PRT auf Diskette gespeichert.
LOAD-PART: Mit dieser Funktion können die mit SAVE-PART gespeicherten Ausschnitte geladen werden. Dazu zeigt das Programm alle Ausschnitte, die auf der Diskette im aktuellen Laufwerk sind, in einem Auswahlmenü an (vergleiche I. 4. LOAD). Das gewünschte File kann dann mit Joystick/Cursortasten angewählt werden. Nach dem Laden kehrt das Programm zum Grafikbildschirm zurück, und der Ausschnitt kann mit dem Cursor beliebig auf dem Bildschirm positioniert werden.
COPY: Der GRAPHICS DESIGNER enthält keine spezielle COPY-Funktion, jedoch lassen sich mehrere Funktionen zu einer solchen umbauen. Man kann zum Beispiel einen zu verschiebenden Ausschnitt mit ZOOM vergrößern und danach beliebig positionieren (vergleiche I. 2. ZOOM). Besser geeignet zum Verschieben sind jedoch die SHRIKE- und die MAGNIFY-Funktionen, indem man als Vergrößerungs-/Verkleinerungsfaktor die Eins eingibt. Dabei arbeitet die SHRIKE-Funktion schneller als MAGNIFY.

I. 6. Ganzbildschirm-Operationen

Der GRAPHICS DESIGNER enthält mehrere Funktionen, deren Ausführung sich auf den gesamten Bildschirm bezieht:
MODE: Mit dem MODE-Befehl wird der Bildschirmmodus gewechselt. Im Gegensatz zu BASIC und den meisten anderen Grafikprogrammen hat der GRAPHICS DESIGNER einen „intelligenten" Modiwechsel. Das heißt, daß beim Umschalten in den jeweiligen Modus der Bildschirm nicht gelöscht, sondern byteweise umgerechnet wird. Es ist klar, daß bei der Umwandlung in den höheren Bildschirmmodus Farben und in den niedrigeren Modus Auflösung verloren gehen. Daher erfolgt am Anfang eine Sicherheitsabfrage.
Beim Umschalten von Mode eins in Mode zwei werden die Farben folgendermaßen verändert:

MODE 1	**MODE 2**
PEN 0	PEN 0

PEN 1	PEN 1
PEN 2	PEN 0
PEN 3	PEN 1

CLG: CLG dient zum Löschen eines Bildschirmes mit einer bestimmten Farbe. Nach dem Aufruf und der Sicherheitsabfrage will der Computer die Farbe wissen, mit der der Bildschirm gelöscht werden soll. Damit ist die Nummer des Pens gemeint; sie muß im Mode eins zwischen null und drei, im Mode zwei zwischen null und eins liegen.
INVERT: Diese Funktion invertiert den gesamten Bildschirm, das heißt, die Farben werden durch XOR verändert. Da dieser Vorgang durch nochmaliges Invertieren rückgängig gemacht werden kann, erfolgt keine Sicherheitsabfrage. Die Farben werden je nach Modus folgendermaßen vertauscht:

MODE 1

VORHER	NACHHER
PEN 0	PEN 3
PEN 1	PEN 3
PEN 2	PEN 1
PEN 3	PEN 0

MODE 2

VORHER	NACHHER
PEN 0	PEN 1
PEN 1	PEN 0

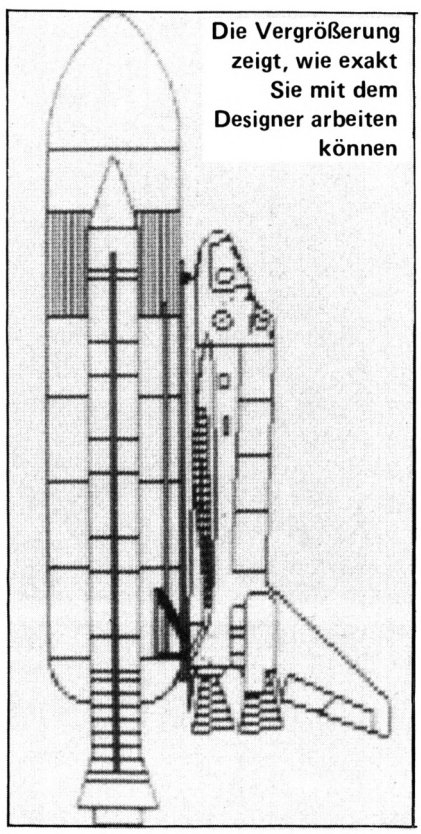
BEND: Diese Funktion „verbiegt" einen ganzen Bildschirminhalt. Es wird ein Effekt erzeugt, als wenn man ein Blatt Papier mit einem Bild verbiegen würde. BEND ist in zwei Unterfunktionen unterteilt: BEND und WAVES.
BEND: Hier wird der Bildschirminhalt einfach verbogen. Nach dem Aufruf der Funktion wird der Cursor dargestellt. Er dient in diesem Fall nur dazu, die Hintergrundfarbe für BEND einzustellen. Dies ist nötig, da beim Verbiegen des Bildschirms ein Rand sichtbar wird. Durch FEUER/ENTER wird die Funktion gestartet. BEND ist relativ zeitintensiv, sie benötigt für den kompletten Bildschirm zirka zwei Minuten. Der Vorgang kann zwar zwischendurch durch ESC unterbrochen werden, allerdings läßt er sich dann nicht mehr rückgängig machen. Nach Beendigung des Verbiegens wird wieder der Cursor dargestellt. Falls das Ergebnis nicht gefällt, kann die Funktion mit DEL storniert werden (vergleiche I 1. UNDO). Drückt man erneut FEUER/ENTER, wird der Bildschirm nochmals verbogen.
WAVES: Diese Funktion entspricht BEND, doch wird der Bildschirminhalt wellenförmig verbogen.

I. 7. Sonstiges

Im GRAPHICS DESIGNER gibt es einige Funktionen, die nicht in die bisher aufgeführten Rubriken passen:
INK: Diese Option dient zum Einstellen der Bildschirmfarben. Nach dem Aufruf werden die Pens des entsprechenden Modus in ihren aktuellen Farben angezeigt. Nun kann jeder Pen einzeln mit Joystick/Cursortasten hoch/runter angewählt und durch rechts/links verändert werden. Die veränderten Werte werden automatisch übernommen. Die Standardeinstellungen beim Programmstart sind:

PEN	INK
0	26
1	00
2	11
3	06

EXIT: Nach Aufruf dieser Funktion wird ein Neustart durchgeführt. Da dabei der Bildschirm gelöscht wird und alle Daten verlorengehen, erfolgt eine Sicherheitsabfrage. Durch EXIT werden alle Parameter auf den Startwert gesetzt.
GMOD: Damit wird der Zeichenmodus für die Funktion LINE und STREAMS gesetzt. Nach dem Anwählen der Funktion erscheint ein Auswahlmenü, und mit Joystick/Cursortasten hoch/runter kann einer der Menüpunkte STANDART, XOR, AND oder OR angewählt werden.



Die Vergrößerung zeigt, wie exakt Sie mit dem Designer arbeiten können

II. Der Font-Designer

Der FONT-DESIGNER dient zum Erstellen beziehungsweise Verändern der Zeichensätze des GRAPHICS DESIGNER. 18 Zeichensätze sind bereits auf der Diskette enthalten, bis zu acht weitere können drei definiert werden. Diese Zeichensätze sind ebenfalls vom DESIGNER aus zu laden (vergleiche I. 3. CHARACTERS). Der FONT-DESIGNER enthält folgende Funktionen:
DEFINE SYMBOL: Mit dieser Funktion kann ein einzelnes Zeichen eines Sets verändert werden. Im Gegensatz zu den meisten anderen Symbol-Editoren lassen sich die Zeichen mit dem FONT-EDITOR nicht nur neu definieren, sondern das entsprechende Zeichen des aktuellen Zeichensatzes wird auf dem Bildschirm dargestellt und kann verändert werden. Hierzu sind folgende Steuerungsmöglichkeiten über die Tastatur gegeben:
Die Steuerung des Cursors erfolgt über die Cursortasten. Mit ENTER kann ein Punkt gesetzt, mit

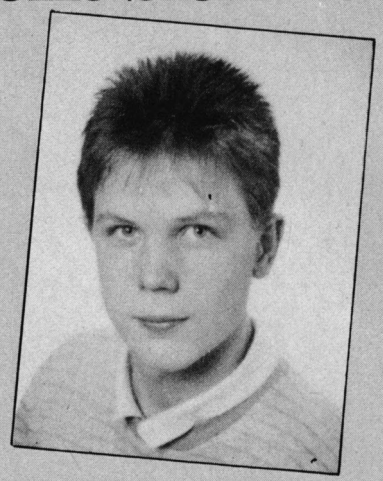
DEL kann er wieder entfernt werden. Der Rücksprung ins Hauptmenü erfolgt mit ESC.
Nach dem Aufruf der Funktion fragt das Programm nach der Nummer des zu verändernden Zeichens. Dieses muß zwischen 32 und 127 liegen. Ist das nicht der Fall, so erfolgt ein Rücksprung. Danach erscheint auf dem Bildschirm das Editierwindow. In dessen linker Seite ist das Zeichen achtmal vergrößert dargestellt. Im Window befindet sich auch der Cursor.
Außerdem enthält es einige Tastaturerklärungen sowie das Zeichen noch einmal im Maßstab 1:1. Alle Veränderungen, die vorgenommen werden, werden hier automatisch dargestellt, so daß man sich das Zeichen auch gleich in Originalgröße ansehen kann. Bei einem Rücksprung mit ESC werden die Veränderungen übernommen.
LOAD SET: Diese Funktion dient dem Laden eines Zeichensatzes. Der Computer fragt nach dessen Nummer, die zwischen Eins und der Anzahl der vorhandenen Zeichensätze liegen muß. Durch die Eingabe von „27" wird der Patternsatz geladen. Eine ungültige Eingabe sorgt für einen Rücksprung ins Hauptmenü.
SAVE SET: Diese Funktion speichert den aktuellen Zeichensatz auf Diskette. Die Steuerung entspricht LOAD PART.
DELETE SET: Mit dieser Option werden alle definierten Veränderungen rückgängig gemacht, der Zeichensatz wird in den Einschaltzustand versetzt. **Achtung:** DELETE SET zerstört automatisch alle noch nicht gespeicherten Veränderungen.
EXIT: Durch Aufruf dieser Funktion wird zum EDITOR-Menü zurückgesprungen. **Achtung:** Auch hierbei werden alle definierten Veränderungen zerstört.

III. Die SLIDESHOW

Das SLIDESHOW-Programm ist ein Utility besonderer Art. Mit diesem Programm lassen sich Bilder, die mit dem DESIGNER erstellt wurden, wie in einer Diashow anzeigen.
Nach dem Start meldet sich das Programm mit INSERT SOURCE DISC. Jetzt wird die Diskette eingelegt, auf der sich die Bilder befinden. Danach drückt man eine beliebige Taste. Nach einer kurzen Wartezeit zeigt das Programm die auf der Diskette enthaltenen Bilder als Menü an.
Mit den Cursortasten wird angewählt. Will man ein Bild laden, so muß man es mit COPY festlegen, der Name wird dann auf grauem Hintergrund angezeigt. Drückt man auf derselben Datei ein zweites Mal COPY, wird die Festlegung rückgängig gemacht.
Hat man alle gewünschten Bilder angewählt, wird das Menü durch ENTER verlassen. Das Programm meldet sich mit einem Titelbild und lädt dabei die erste Datei. Sobald der Ladevorgang abgebrochen ist, kann man sich das Bild

Der Autor stellt sich vor

Wir haben – nach langer Zeit – wieder ein Programm, dem der Titel „Listing des Monats" gebührt. Der Autor ist Jörg Schwieder aus Überlingen am Bodensee.
Wenn man die Leistungsfähigkeit des Programms betrachtet, möchte man kaum glauben, daß Jörg Schwieder neben seinem Computer noch andere Interessen hat. Doch seine weiteren Hobbies, Tennis und Musik, haben mit Technik wenig zu tun. Daß er sich zudem mit Luft- und Raumfahrttechnik beschäftigt, scheint eher verständlich.
Jörg Schwieder ist am 30. Dezember 1971 geboren und besucht zur Zeit die zehnte Klasse des Gymnasiums. Seinen CPC 464 besitzt er seit fast drei Jahren. Die Spiele-Programmierung interessiert ihn weniger; er beschäftigt sich hauptsächlich mit der Erstellung von Anwendungs-Programmen aller Art.



Der Graphic Designer von Jörg Schwieder ist ein weitaus umfangreicheres Programm, als wir es Ihnen in gedruckter Form vorstellen könnten. Deshalb haben wir uns entschlossen, Ihnen das Listing auch auf einer gesonderten Diskette anzubieten. Dort finden Sie einige zusätzliche Schriftsätze sowie Demo-Bilder, die vom Autor erstellt wurden.
Alle diejenigen, die die Mühe des Abtippens nicht scheuen, finden jedoch eine lauffähige Minimal-Version in dieser Ausgabe. □

per Tastendruck ansehen. Dabei wird das alte Bild, wie bei einem Dia-Projektor, links aus dem Bildschirm herausgefahren, durch das neue ersetzt und wieder hereingefahren.
Für die SLIDESHOW können nur Bilder verwendet werden, die vom GRAPHICS DESIGNER stammen. Fremdbilder müssen zuvor mit dem DESIGNER geladen und wieder abgespeichert werden, da die SLIDESHOW das GRAPHICS DESIGNER-Aufzeichnungsformat verwendet. Dadurch erscheint jedes neue Bild automatisch im richtigen Mode und in den richtigen Farben, unabhängig vom Mode und von den Farben des vorhergehenden Bildes.
Sobald das neue Bild vollständig hereingefahren ist, wird das nächste geladen. Der Vorgang wiederholt sich bis zum letzten Bild. Sobald das letzte dargestellt ist, wartet der Computer auf einen Tastendruck und fragt dann, ob das Programm neu gestartet oder zum EDITOR zurückgekehrt werden soll.

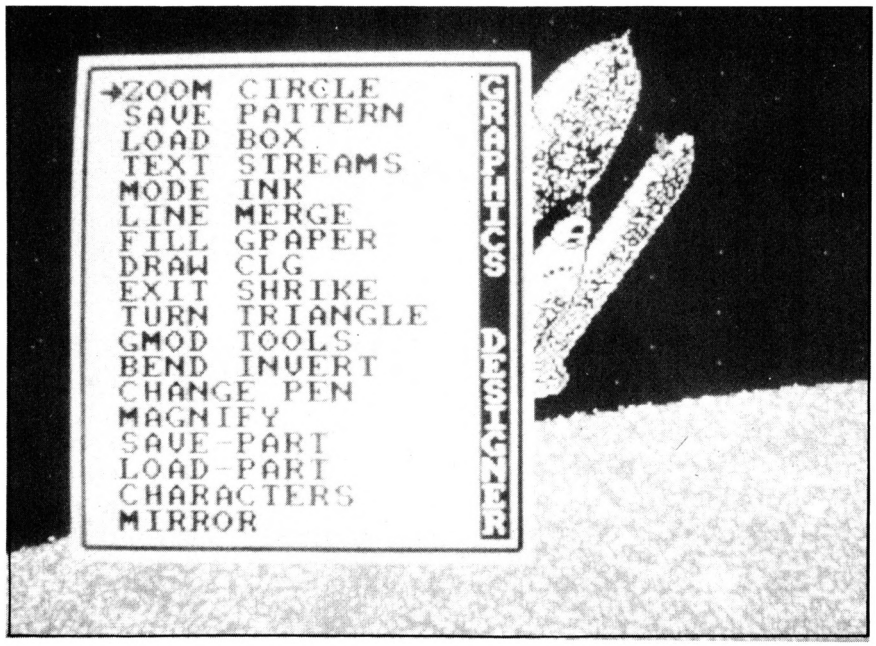
IV. Der Befehlssatz

Der GRAPHICS DESIGNER besteht zu zirka 50 Prozent aus BASIC und zu 50 Prozent aus Maschinensprache-Unterroutinen. Diese Unterroutinen sind der besseren Übersichtlichkeit wegen als RSX-Befehle definiert.
Der GRAPHICS DESIGNER enthält insgesamt 21 RSX-Befehle, die im folgenden einzeln erklärt werden. Sie befinden sich in mehreren Dateien, nach denen sie geordnet dargestellt werden. Alle RSX-Befehle lassen sich natürlich auch in eigene Programme einbinden. Sie müssen dazu nur mit 'LOAD"<NAME>", <STARTADRESSE>' und 'CALL <STARTADRESSE>' initialisiert werden. Alle Befehle werden hier ohne den RSX-Strich (SHIFT + Klammeraffe) angegeben, er wird jedoch zum Aufruf benötigt. Sie werden nach den Dateien geordnet, in denen sie sich befinden. Folgende Variablen werden verwendet:

Variable	Bedeutung
x	X-Position
y	Y-Position
xl	Länge in X-Richtung
yl	Länge in Y-Richtung
xf	Vergrößerungsfaktor in X-Richtung
yf	Faktor in Y-Richtung
f	Faktor allgemein
modus	Bildschirmmodus
as	ASCII-Nummer eines Zeichens
a$	beliebiger, bereits definierter String
name$	Dateiname
farbe	Farbstift

Die Dateien:
G2: Startadresse: &AF00
Befehle:
IN: Speichert den aktuellen Bildschirm in den zweiten Bildschirmspeicher ab &6000 SYNTAX: IN
OUT: Kopiert den Inhalt des zweiten Bildschirmspeichers in den ersten Bildschirmspeicher, SYNTAX: OUT
GPEN: Setzt den Farbstift für Grafikfunktionen (PLOT, DRAW...). SYNTAX: GPEN, farbe
GPAPER; Setzt die Grafikhintergrundfarbe für Textausgabe mit TAG oder für Patterns. SYNTAX: GPAPER, farbe
COPYCHR: Liest ein Textzeichen von der aktuellen Text-Cursorposition. SYNTAX: COPYCHR, <KLAMMERAFFE>a$
FILL: Füllt einen geschlossenen Bereich ab der aktuellen Cursorposition mit der aktuellen Farbe oder einem Pattern. Für diese Funktion muß G6 nach &A000 geladen sein; G6 darf nicht gestartet werden. SYNTAX: FILL, farbe
G4: Startadresse: &5F00
Befehle:
MERGE: Verknüpft den aktuellen Bildschirminhalt mit einem gespeicherten Bild. Für diese Funktion muß G9 nach &BE00 geladen sein, darf aber nicht gestartet werden. SYNTAX: MERGE, <KLAMMERAFFE>name$
CHANGE: Mit diesem Befehl werden in einem Bildausschnitt alle Punkte, die die Farbe eins (i1) haben, durch Punkte mit Farbe zwei (i2) ersetzt. SYNTAX: CHANGE, x,y,x1,yx,i1,i1
G5: Startadresse: &5480
Befehle:
MODE: Wechselt den Bildschirmmodus (nur Mode eins und zwei), ohne den Bildschirm zu löschen. SYNTAX: MODE, modus
BEND.IN: Liest eine Grafikzeile vom Bildschirm und speichert sie ab &5600. SYNTAX: BEND.IN,y
BEND.OUT: Gibt eine mit BEND.IN geladene Zeile ab der X-Position wieder auf dem Bildschirm aus. SYNTAX: BEND.OUT,x,y
INVERT: Invertiert den ganzen Bildschirm. SYNTAX: INVERT
G7: Startadresse: &5170
G6: Ladeadresse: &A000 (nicht starten)
Befehle:
PART.IN: Lädt einen Bildschirmausschnitt in den Speicher ab Adresse &5600. SYNTAX: PART.IN,x,y,x1,y1,modus
ZOOM.OUT: Gibt einen ab Adresse &5600 abgelegten Ausschnitt achtfach vergrößert auf den Bildschirm aus. SYNTAX: ZOOM.OUT,x,y,x1,y1,modus
PART.MAG: Gibt einen ab &5600 im Speicher stehenden Ausschnitt vergößert aus. SYNTAX: PART.MAG,f,x,y,x1,y1, modus
PART.OUT: Gibt einen ab &5600 im Speicher stehenden Ausschnitt in Originalgröße wieder. SYNTAX: PART.OUT,x,y, x1,y1,modus
SHRIKE.IN: Legt einen Ausschnitt verkleinert ab &5600 ab. SYNTAX: SHRIKE.IN,f,x,y,x1, y1,modus
MIRROR.OUT: Gibt einen im Speicher stehenden Ausschnitt in Y-Richtung gespiegelt aus. SYNTAX: MIRROR.OUT,x,y,x1,y1, modus
PATTON: Ein Pattern wird für die Grafikausgabe festgelegt. SYNTAX: PATTON, pen, paper, asc

PATTOFF: Schaltet die Grafikausgabe auf die Normalausgabe zurück.
Achtung: x1 und y1 sind die Größen des im Speicher befindlichen Ausschnittes. Es muß y1 daher immer die Hälfte der Bildpixel, x1 im Mode eins ebenfalls die Hälfte der Bildpixel betragen.
G8: Startadresse: &5E00
Befehle:
CHAR: Gibt ein Textzeichen vergrößert an einer Grafikposition aus. SYNTAX: CHAR,asc,xf,yf, x,y
Achtung: Die x- und y-Werte sind wirkliche Bildschirmpositionen, das heißt, x darf im Mode eins maximal 320 sein, y ist immer maximal 200.



Das Menü des Graphic Designers bietet zahlreiche Optionen

Hinweise zum Programm Convert.bas

Aus Speicherplatzgründen muß die lauffähige Version des Hauptprogramms "G1.bas" Steuerzeichen enthalten. Da diese aber nicht abgedruckt werden können, mußten wir einen kleinen Umweg machen. Im Listing G1 finden Sie an einigen Stellen folgende Zeichen: "{" und "£".
Geben Sie diese Zeichen genauso ein, wie sie im Listing angegeben sind und speichern dieses Teilprogramm als "G1.cvt",a ab. Beachten Sie, daß G1.cvt wirklich als ASCII-Datei gespeichert wird. Anschließend muß der Konverter abgetippt, gesichert und dann gestartet werden. Er fügt an den entsprechenden Stellen im Listing die erforderlichen Steuerzeichen ein. Meldet sich der Konverter mit Ready zurück, muß die jetzt lauffähige Version als G1.bas abgespeichert werden. War alles korrekt, wird der Konverter nicht mehr benötigt. Aber bitte noch nicht löschen, sondern erst überprüfen, ob das Gesamtprogramm wirklich fehlerlos arbeitet.
Danach müssen Sie im Programm G1.bas in der Zeile 20 nach dem DEFINT-Befehl ein Leerzeichen(!) einfügen, ebenso in der Zeile 1880. In der Zeile 20 muß nach dem DATA-Befehl ebenfalls ein Leerzeichen eingefügt werden.
Wichtig: Nach erfolgter Konvertierung sollte ein Reset (CTRL-SHIFT-ESC) durchgeführt werden, da CONVERT die Betriebssystem-Routine 'CAS IN CHAR' umbiegt und daher bei allen Programmen, die als ASCII-Datei vorliegen und nach dem Start von 'CONVERT' geladen werden, die Zeichen '{' und '£' in die Steuerzeichen 23 beziehungsweise 24 verwandelt. □

```
1 '******************************* (2506)
2 '*    CPC GRAPHICS DESIGNER    * (25C8)
3 '*            VON              * (2538)
4 '*       JOERG SCHWIEDER       * (25B0)
5 '*            FUER             * (2558)
6 '*          CPC-WELT           * (2537)
7 '*  STARTPROGRAMM (GRAPH.BAS)  * (25EB)
8 '*          CPC 464         ME* (2597)
9 '******************************* (2516)
10 SYMBOL AFTER 256                (0A9D)
20 POKE &B296,&70:POKE &B294,32:PO
KE &BE78,255:POKE &BDEE,&C9        (2615)
30 KEY DEF 76,1,88,88,88           (1442)
40 KEY DEF 18,1,13,13,13           (142B)
50 MODE 1                          (070C)
60 INPUT"Number of character set(1
-26)";num                          (2CE3)
70 IF num>0 AND num<27 THEN nam$="
G3"+CHR$(64+num)                   (32AD)
80 MODE 1                          (0748)
90 MEMORY &516F                    (0982)
100 LOAD"g2.bin"                   (0E81)
110 CALL &AF00                     (09C9)
120 nam$=nam$+".BIN":LOAD nam$,&A0
70                                 (2521)
130 LOAD"g4.bin",&5F00:CALL &5F00  (177A)
140 LOAD"g5.bin":CALL &5480        (1326)
150 LOAD"g6.bin",&A000             (129D)
160 LOAD"g3pat.bin",&A370          (15C5)
170 LOAD"g7.bin",&5170:CALL &5170  (17C6)
180 LOAD"g8.bin",&5E00:CALL &5E00  (17BE)
190 LOAD"g9.bin",&BE00             (12BC)
200 RUN"g1                         (090B)
```

```
**********************************
*    The CPC-Graphics Designer   *
*          Hauptprogramm         *
*    speichern als 'g1.cut',a    *
*               von              *
*         Joerg Schwieder        *
*              fuer              *
*            CPC-Welt            *
*            CPC 464          me*
**********************************
10 MEMORY &516F:ON ERROR GOTO 1880
                                  (14E3)
20 DEFINT a-z:modus=1:MODE 1:FOR n
=0 TO 3:READ a:farb(n)=a:INK n,a:N
EXT:PEN 1:BORDER 0:farbe=1:/GPEN,1
:DATA 26,0,11,6                   (7667)
30 x=320:y=200:cha=0              (1E33)
40 modusf=4\modus:modi=3-modus    (2C69)
50 /IN:CALL &BB48                 (0FFA)
60 LOCATE#2,1,1:POKE &B08B,&8C:POK
E &B08C,&AE:POKE &B295,0:POKE &B54
1,&4C:POKE &B543,&9C:zon$="{"+CHR$
(zmo):PRINT#2,"{@":/PATTOFF       (65AA)
70 inki(0)=0:inki(1)=&F0:inki(2)=&
F:inki(3)=&FF:ORIGIN 0,0:GOSUB 212
0:RESTORE 80:FOR n=0 TO 17:READ a$
:LOCATE 2,1+n:PRINT a$:NEXT       (79AE)
80 DATA ZOOM CIRCLE,SAVE PATTERN,L
OAD BOX,TEXT STREAMS,MODE INK,LINE
 MERGE,FILL GPAPER,DRAW CLG,EXIT S
HRIKE,TURN TRIANGLE,GMOD TOOLS,BEN
D INVERT,CHANGE PEN,MAGNIFY,SAVE-P
ART,LOAD-PART,CHARACTERS,MIRROR   (C82C)
90 PRINT#2,CHR$(7):z=1:mz=1       (1D2D)
100 LOCATE mz,z:PRINT CHR$(32):IF
NOT INKEY(0)OR NOT INKEY(72)THEN z
=z-1                              (3BEC)
110 IF NOT INKEY(2)OR NOT INKEY(73
)THEN z=z+1                       (258E)
120 IF NOT INKEY(74)OR NOT INKEY(8
)THEN mz=mz-5                     (2732)
130 IF NOT INKEY(75)OR NOT INKEY(1
)THEN mz=mz+5                     (2763)
140 IF z=0 THEN z=18              (1737)
150 IF z=19 THEN z=1              (17AB)
160 mz=MIN(mz,6):mz=MAX(mz,1):IF m
z=6 THEN z=MIN(z,12)              (45E2)
170 IF NOT INKEY(18)OR NOT INKEY(7
6)THEN GOTO 190                   (20F6)
180 LOCATE mz,z:PRINT CHR$(243):WH
ILE INKEY$="":WEND:GOTO 100       (2A14)
190 IF mz=1 THEN 200 ELSE 210     (1B44)
200 ON z GOTO 220,390,1070,720,460
,480,690,1310,1610,1810,2200,2010,
2470,1360,1410,1470,1650,2090     (55F2)
210 ON z GOTO 500,800,1260,1860,96
0,1890,1280,1330,1620,1760,1900,20
00                                (3D25)
220 /OUT:GOSUB 2180:spalte=40*modu
s:/PART.IN,x,y,spalte,24,modus:WIN
DOW 1,spalte,1,25:CLS:/ZOOM.OUT,mo
dus:PRINT#2,CHR$(7):xset=x:yset=y:
cx=0:cy=17:xc=1:yc=24             (BFFD)
230 f1=TEST(cx,cy):f2=TESTR(4,-4):
f3=TESTR(2,-2):f4=TESTR(2,-2):f5=T
EST(cx+2,cy):f6=TEST(cx+4,cy):f7=T
EST(cx,cy-2):f8=TEST(cx,cy-4)     (A7A9)
240 PLOT cx,cy,farbe:PLOTR 4,-4,fa
rbe:PLOTR 2,-2,farbe:PLOTR 2,-2,fa
rbe:PLOT cx+2,cy,farbe:PLOT cx+4,c
y,farbe:PLOT cx,cy-2,farbe:PLOT cx
,cy-4,farbe:IF farbe=f1 THEN PLOT
cx,cy,(farbe+1)MOD modusf         (E8E3)
250 BORDER farb(farbe):WHILE INKEY
$="":WEND:PLOT cx,cy,f1:PLOTR 4,-4
,f2:PLOTR 2,-2,f3:PLOTR 2,-2,f4:PL
OT cx+2,cy,f5:PLOT cx+4,cy,f6:PLOT
 cx,cy-2,f7:PLOT cx,cy-4,f8       (B564)
260 IF INKEY(0)=0 OR INKEY(72)=0 T
HEN cy=cy+16:yc=yc-1:IF yc<1 THEN
yc=24:cy=17                       (5687)
270 IF INKEY(2)=0 OR INKEY(73)=0 T
HEN cy=cy-16:yc=yc+1:IF yc>24 THEN
 yc=1:cy=385                      (57C9)
280 IF INKEY(8)=0 OR INKEY(74)=0 T
HEN cx=cx-16\modus:xc=xc-1:IF xc<1
 THEN xc=spalte:cx=640-(16\modus) (7557)
290 IF NOT INKEY(1)OR NOT INKEY(75
)THEN cx=cx+16\modus:xc=xc+1:IF xc
>40*modus THEN xc=1:cx=1          (6532)
300 IF NOT INKEY(18)OR NOT INKEY(7
6)THEN GOSUB 350                  (20BD)
310 IF INKEY(0)=32 THEN farbe=farb
e+1                               (256C)
320 IF INKEY(2)=32 THEN farbe=farb
e-1                               (25A5)
330 farbe=ABS(farbe)MOD modusf:IF
NOT INKEY(66)THEN 360             (350F)
340 GOTO 230                      (0AE6)
350 LOCATE xc,yc:PEN farbe:PRINT C
HR$(143):RETURN                   (28E7)
360 PLOT cx,cy,f1:PLOTR 4,-4,f2:PL
OTR 2,-2,f3:PLOTR 2,-2,f4:PLOT cx+
2,cy,f5:PLOT cx+4,cy,f6:PLOT cx,cy
-2,f7:PLOT cx,cy-4,f8             (97A6)
370 /SHRIKE.IN,8,0,16,40*modus,24,
modus:/OUT:x=xset:y=yset          (4F54)
380 /PART.OUT,x,y,40*modus,24,modu
s:LOCATE#2,1,1:PRINT#2,CHR$(7):GOS
UB 2180:GOTO 380                  (4FC0)
390 WINDOW 1,40*modus,1,25:CLS:INK
 1,26:INK 0,26:CAT:a$="D":c$=" ":F
OR n=4 TO 37:LOCATE 1,n:/COPYCHR,@
a$:IF a$=" "THEN 410              (74DB)
400 a$="1":a$=" ":NEXT            (185C)
410 FOR a=1 TO 5:LOCATE a,n+1:/COP
YCHR,@c$:a$=a$+c$:NEXT:b=VAL(a$):P
EN 1:INK 1,0:IF b<17 THEN MODE mod
us:GOTO 450                       (754F)
420 IF b>17 THEN CLS:PRINT b;" K f
ree"                              (2394)
430 PRINT:PRINT:INPUT"Name";name$:
IF name$=""THEN MODE modus:/OUT:GO
TO 450                            (3D13)
440 MODE modus:/OUT:FOR n=0 TO 3:P
OKE &FFFC+n,farb(n):NEXT:POKE &FFF
B,modus:SAVE name$,b,&C000,&4000  (6316)
450 INK 1,farb(1):INK 0,farb(0):/O
UT:GOTO 30                        (2E14)
460 GOSUB 2150:IF modus=2 THEN mod
us=1 ELSE modus=2                 (32C1)
470 /OUT:/MODE,modus:GOTO 40      (2057)
480 /OUT                          (0ABE)
490 GOSUB 650:LOCATE#2,1,1:PRINT#2
,zon$:MOVE xset,yset:DRAW x,y,farb
e:xset=x:yset=y:PRINT#2,"{@":GOTO
490                               (6DBC)
500 CLS:PRINT" C ircle or":PRINT"
E llipse"                         (22F7)
510 a$=UPPER$(INKEY$):IF a$<>"C"AN
D a$<>"E"THEN 510                 (2A89)
520 /OUT:mit=0                    (1391)
530 GOSUB 650:GOSUB 540:GOTO 530  (16FC)
540 IF mit=0 THEN mx=x:my=y:mit=1:
RETURN                            (32BB)
550 IF a$="C"THEN GOSUB 630:mit=0:
RETURN                            (216B)
560 IF zahl=1 THEN rady=ABS(y-my):
GOSUB 570:mit=0:zahl=0:RETURN ELSE
 radx=ABS(x-mx):zahl=1:RETURN     (6AA5)
570 xa=x:ya=y:x=mx+radx:y=my:GOSUB
 650:LOCATE#2,1,1:PRINT#2,zon$:x=x
-mx:y=y-my:IF x THEN w!=ATN(y/x)EL
SE w!=90                          (97D0)
580 ORIGIN mx,my:ws!=SIN(w!):wc!=C
OS(w!):a=radx:b=rady:back=1       (544F)
590 x=a:GOSUB 610:PLOT x,y,farbe:F
OR x=a TO-a STEP-2:xx=x:ON back GO
SUB 610,620                       (5F0A)
600 DRAW x,y:x=xx:NEXT:IF back=2 T
HEN ORIGIN 0,0:x=xa:y=ya:PRINT#2,"
{@":RETURN ELSE back=2:GOTO 590   (653D)
610 y=SQR(b*b-(b*b*x*x)/(a*a)):t=x
*wc!-y*ws!:y=x*ws!+y*wc!:x=t:RETUR
N                                 (7BA1)
620 y=-SQR(b*b-(b*b*x*x)/(a*a)):t=
x*wc!-y*ws!:y=x*ws!+y*wc!:x=t:RETU
RN                                (7C65)
630 r=SQR((x-mx)^2+(y-my)^2):DEG:L
OCATE#2,1,1:PRINT#2,zon$:n=ROUND(P
I*SQR(r)+1):wsn!=SIN(360/n):wcs!=C
OS(360/n):xd=r:yd=0:MOVE mx+r,my:F
OR i=1 TO n                       (B5CD)
640 m=wcs!*xd-wsn!*yd:yd=wsn!*xd+w
cs!*yd:xd=m:DRAW mx+xd,my+yd,farbe
:NEXT:RAD:PRINT#2,"{@":RETURN     (80D4)
650 IF dar=1 THEN f1=TEST(x,y):f2=
TESTR(4,-4):f3=TESTR(2,-2):f4=TEST
R(2,-2):f5=TEST(x+2,y):f6=TEST(x+4
,y):f7=TEST(x,y-2):f8=TEST(x,y-4)E
LSE PRINT#2,"{A";:MOVE x,0:DRAW x,
400,farbe:MOVE 0,y:DRAW 640,y:GOTO
 670                              (EDFA)
660 modusf=4\modus:PLOT x,y,farbe:
PLOTR 4,-4,farbe:PLOTR 2,-2,farbe:
PLOTR 2,-2,farbe:PLOT x+2,y,farbe:
PLOT x+4,y,farbe:PLOT x,y-2,farbe:
PLOT x,y-4,farbe:IF farbe=f1 THEN
PLOT x,y,(farbe+1)MOD modusf      (F10F)
670 BORDER farb(farbe):WHILE INKEY
$="":r=1:WEND:IF dar=1 THEN PLOT x
,y,f1:PLOTR 4,-4,f2:PLOTR 2,-2,f3:
PLOTR 2,-2,f4:PLOT x+2,y,f5:PLOT x
+4,y,f6:PLOT x,y-2,f7:PLOT x,y-4,f
8 ELSE MOVE x,0:DRAW x,400:MOVE 0,
y:DRAW 640,y:PRINT#2,"{@";        (F477)
680 GOSUB 2340:IF s THEN s=0:RETUR
N ELSE 650                        (2364)
690 /OUT                          (0A63)
700 GOSUB 650:/GPEN,farbe:MOVE x,y
:IF(TEST(x,y)=gpap AND c<>91)OR TE
ST(x,y)=farbe THEN 700 ELSE POKE &
B295,&FF:POKE &B297,&A3:/PATTON,32
+c,inki(farbe),inki(gpap):/FILL,fa
rbe:/PATTOFF                      (CC38)
710 POKE &B295,0:GOTO 700         (1249)
720 CLS:PRINT"TEXT";:IF cha=1 THEN
 POKE &B295,255:POKE &B297,&A0:POK
E &B541,&9C                       (3939)
730 IF cha=2 THEN POKE &B295,255:P
OKE &B297,&A0:POKE &B543,&4C      (2EE8)
740 INPUT"";text$:IF text$=""THEN/
OUT:GOTO 50                       (29B5)
750 POKE &B295,0:POKE &B541,&4C:PO
KE &B543,&9C:INPUT"FAKTOR";f:IF f=
0 THEN f=1                        (41FD)
760 IF cha<>0 THEN POKE &B295,255 (1A91)
770 /OUT:WINDOW SWAP 0,1          (1227)
780 GOSUB 650:/GPEN,farbe:GOSUB 79
0:GOTO 780                        (26BF)
790 FOR n=1 TO LEN(text$):/CHAR,AS
C(MID$(text$,n,1)),f,f,x\modi+f*(n
-1)*8,y\2:NEXT:RETURN             (6A60)
800 CLS:POKE &B295,255:POKE &B297,
&A3                               (190C)
810 b=1:FOR a=1 TO 13:FOR n=1 TO 7
:LOCATE 2*n,a:PRINT CHR$(b+32);:b=
b+1:NEXT:NEXT:POKE &B295,0        (5B3F)
820 z=1:p=1:c=1                   (1968)
830 LOCATE 2*z-1,p:PRINT CHR$(243)
;:WHILE INKEY$="":WEND:LOCATE 2*z-
1,p:PRINT CHR$(32);:IF NOT INKEY(7
4)OR NOT INKEY(8)THEN z=z-1:c=c-1 (6FC7)
840 IF NOT INKEY(75)OR NOT INKEY(1
)THEN z=z+1:c=c+1                 (31D3)
850 IF NOT INKEY(76)OR NOT INKEY(1
8)THEN 910                        (1EAE)
860 IF c=0 THEN z=7:c=91:p=13     (2652)
870 IF z=8 THEN z=1:p=p+1         (2222)
880 IF c=92 THEN z=1:c=1:p=1      (2517)
890 IF z=0 THEN z=7:p=p-1         (22CE)
900 GOTO 830                      (0A14)
910 POKE &B295,255:FOR n=14 TO 16:
FOR a=2 TO 4:LOCATE a,n:PRINT CHR$
(c+32);:NEXT:NEXT                 (4885)
920 POKE &B295,0                  (0C4F)
930 PRINT:PRINT"REALLY(Y/N)"      (15E9)
940 a$=UPPER$(INKEY$):IF a$="Y"THE
N 60 ELSE IF a$<>"N"THEN 940      (33DA)
```

CPC Graphic Designer

```
950 GOTO 830                                    <0A78>
960 CLS:FOR n=1 TO modusf:LOCATE 2
,n:PEN n-1:PRINT CHR$(143):NEXT:a=
0:PEN 1                                         <43F3>
970 LOCATE 1,a+1:PRINT CHR$(243):W
HILE INKEY$="":WEND:LOCATE 1,a+1:P
RINT CHR$(32):IF NOT INKEY(1)OR NO
T INKEY(75)THEN GOSUB 1030                      <514B>
980 IF NOT INKEY(8)OR NOT INKEY(74
)THEN GOSUB 1050                                <1F75>
990 IF NOT INKEY(0)OR NOT INKEY(72
)THEN a=a-1:a=MAX(a,0)                          <35E8>
1000 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN a=a+1:a=MIN(a,modusf-1)                  <3F99>
1010 IF NOT INKEY(66)THEN 60                    <1480>
1020 GOTO 970                                   <0A63>
1030 farb(a)=farb(a)+1:IF farb(a)>
26 THEN farb(a)=0                               <4770>
1040 INK a,farb(a):RETURN                       <1B33>
1050 farb(a)=farb(a)-1:IF farb(a)<
0 THEN farb(a)=26                               <4780>
1060 INK a,farb(a):RETURN                       <1B5B>
1070 GOSUB 1080:GOTO 1210                       <1074>
1080 WINDOW 1,40,1,25:CLS:a=FRE(""
):INK 0,26:INK 1,26:PEN 1:CAT:l=1:
lll=0:lo=0                                      <4A52>
1090 FOR n=4 TO 25:a$=" ":LOCATE 1
,n:/COPYCHR,@a$:IF a$<>" "THEN GOS
UB 1240 ELSE IF lll=0 THEN lll=1:l
=21:GOTO 1090 ELSE GOTO 1110                    <7901>
1100 NEXT                                       <065F>
1110 CLS:lo=lo-1:IF lo<0 THEN 1230
                                                <248D>
1120 FOR n=0 TO lo:INK 1,0:PEN 1:P
RINT" "b$(n):NEXT:num=0                         <39DB>
1130 LOCATE 2,num+1:PRINT"#"b$(num
)"#":WHILE INKEY$="":WEND:LOCATE 2
,num+1:PRINT b$(num):IF NOT INKEY(
0)OR NOT INKEY(72)THEN num=num-1                <704A>
1140 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN num=num+1                                <29BE>
1150 IF NOT INKEY(66)THEN/OUT:GOTO
 1230                                           <1B2A>
1160 IF NOT INKEY(18)OR NOT INKEY(
76)THEN 1200                                    <1E7C>
1170 IF num<0 THEN num=lo                       <1EE7>
1180 IF num>lo THEN num=0                       <1EE4>
1190 GOTO 1130                                  <0AB8>
1200 RETURN                                     <068A>
1210 /OUT:MODE modus:a=FRE(""):LOA
D b$(num),&C000:IF PEEK(&FFFC)<>PE
EK(&FFFD)THEN FOR r=0 TO 3:farb(r)
=PEEK(&FFFC+r):NEXT:modus=PEEK(&FF
FB)                                             <820C>
1220 /IN                                        <0994>
1230 MODE modus:/OUT:FOR n=0 TO 3:
INK n,farb(n):NEXT:GOTO 30                      <3FE2>
1240 a$="":c$="K":FOR o=0 TO 16:LO
CATE l+o,n:/COPYCHR,@c$:a$=a$+c$:N
EXT:IF RIGHT$(a$,3)="17K"THEN b$(l
o)=LEFT$(a$,12):lo=lo+1                         <8FEF>
1250 RETURN                                     <06EE>
1260 /OUT                                       <0AD9>
1270 GOSUB 2260:LOCATE#2,1,1:PRINT
#2,zon$:MOVE xset,yset:DRAW xset,y
,farbe:DRAW x,y:DRAW x,yset:DRAW x
set,yset:PRINT#2,"{@":GOTO 1270                 <8348>
1280 CLS:PRINT"CODE"                            <0E73>
1290 a$=INKEY$:IF a$<"0"OR a$>"3"T
HEN 1290                                        <2612>
1300 gpap=VAL(a$):/GPAPER,gpap:GOT
O 60                                            <2C08>
1310 /OUT                                       <0A3D>
1320 GOSUB 650:PLOT x,y,farbe:GOTO
 1320                                           <257A>
1330 GOSUB 2150:CLS                             <0CFF>
1340 INPUT"PEN";a:IF a<0 OR a>modu
sf-1 THEN 1340                                  <3205>
1350 CLG a:GOTO 50                              <11AD>
1360 CLS                                        <06CF>
1370 INPUT"FACTOR(1-10)";f:IF f<1
OR f>10 THEN 1370                               <32A1>
1380 /OUT                                       <0AC9>
1390 GOSUB 2260:xd=x-xset+modi:yd=
y-yset+2:IF(xd*yd)/(2*(2\modus))>1
920 THEN 50                                     <64F8>
1400 /PART.IN,xset,yset,xd\modi,yd
\2,modus:LOCATE#2,1,1:PRINT#2,CHR$
(7):GOSUB 650:/PART.MAG,f,x,y,xd\m
odi,yd\2,modus:GOTO 1390                        <9473>
1410 /OUT                                       <0A06>
1420 GOSUB 2260:IF x<xset OR y<yse
t THEN 1420                                     <2E55>
1430 xd=x-xset+modi:yd=y-yset+2:PO
KE &55FE,xd:POKE &55FF,yd:IF(xd\mo
di)*(yd\2)>1920 THEN 30                         <717C>
1440 /PART.IN,xset,yset,xd\modi,yd
\2,modus:lae=(xd\modi)*(yd\2):/OUT
:GOSUB 2120:INPUT"NAME :";nam$:IF
nam$=""THEN/OUT:GOTO 1460                       <93D1>
1450 /OUT:SAVE nam$+".prt",b,&55FE
,lae+2                                          <2CB2>
1460 GOTO 30                                    <0A6B>
1470 DIM nama$(40):WINDOW 1,40,1,2
5:CLS:INK 0,26:INK 1,26:a$="*.prt"
:/DIR,@a$:d=1:x=1:r=1:y=4                       <6370>
1480 nama$(d)="":c$=" ":FOR a=x TO
 x+7:LOCATE a,y:/COPYCHR,@c$:nama$
(d)=nama$(d)+c$:NEXT:IF nama$(d)=S
PACE$(8)THEN 1500                               <8DB1>
1490 y=y+1:d=d+1:GOTO 1480                      <22E9>
1500 IF x=1 THEN x=16:y=4:GOTO 148
0                                               <24B7>
1510 CLS:INK 1,0:y=1:d=d-1:IF d=0
THEN 1600                                       <2ECA>
1520 FOR n=1 TO d:PRINT" "nama$(n)
:NEXT:num=1                                     <311D>
1530 LOCATE 2,num:PRINT"#"nama$(nu
m)"#":WHILE INKEY$="":WEND:LOCATE
2,num:PRINT nama$(num):IF NOT INKE
Y(0)OR NOT INKEY(72)THEN num=num-1
                                                <7230>
1540 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN num=num+1                                <29E1>
1550 IF NOT INKEY(66)THEN/OUT:GOTO
 1880                                           <1BAD>
1560 num=num MOD d+1:IF NOT INKEY(
18)OR NOT INKEY(76)THEN nam$=nama$
(num):GOTO 1580                                 <4E5D>
1570 GOTO 1530                                  <0A3A>
1580 ERASE nama$:num=FRE(""):LOAD
nam$+".prt",&55FE:CLOSEIN:MODE mod
us:/OUT:CALL &BB48                              <4952>
1590 GOSUB 650:xd=PEEK(&55FE):yd=P
EEK(&55FF):/PART.OUT,x,y,xd\modi,y
d\2,modus                                       <5A7B>
1600 GOTO 1590                                  <0A5C>
1610 GOSUB 2150:RUN                             <0C30>
1620 CLS:INPUT"FACTOR(1-10)",multi
:/OUT                                           <25D1>
1630 a=0:GOSUB 2260:xd=x-xset+modi
:yd=y-yset+2:IF(xd*yd)/(2*modi*mul
ti*multi)>1920 THEN 50                          <7825>
1640 /SHRIKE.IN,multi,xset,yset,(x
d\modi)\multi,(yd\2)\multi,modus:L
OCATE#2,1,1:PRINT#2,CHR$(7):GOSUB
650:/PART.OUT,x,y,(xd\modi)\multi,
(yd\2)\multi,modus:GOTO 1630                    <C62E>
1650 CLS:PRINT" ABC":POKE &B295,25
5:POKE &B297,&A0:PRINT" ABC"                    <298B>
1660 POKE &B297,&A0:PRINT" abc"                 <1618>
1670 POKE &B295,0                               <0C2E>
1680 PRINT" CHANGE":LOCATE 1,1:PRI
NT CHR$(243):z=1                                <254D>
1690 WHILE INKEY$="":WEND:LOCATE 1
,z:PRINT CHR$(32):IF NOT INKEY(0)O
R NOT INKEY(72)THEN IF z>1 THEN z=
z-1                                             <4CCC>
1700 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN IF z<4 THEN z=z+1                        <3099>
1710 IF NOT INKEY(18)OR NOT INKEY(
76)THEN 1730                                    <1E65>
1720 LOCATE 1,z:PRINT CHR$(243):GO
TO 1690                                         <1CAF>
1730 IF z<>4 THEN cha=z-1:GOTO 60                <23BC>
1740 INPUT"Nr. of set";num:IF num>
0 AND num<27 THEN nam$="G3"+CHR$(6
4+num)                                          <4C6B>
1750 a=FRE(""):LOAD nam$,&A070:GOT
O 60                                            <2367>
1760 /OUT:back=0                                <142A>
1770 GOSUB 650:PLOT x,y,farbe:IF b
ack=0 THEN back=1:x1=x:y1=y:GOTO 1
770                                             <5315>
1780 IF back=1 THEN back=2:x2=x:y2
=y:GOTO 1770                                    <38AD>
1790 LOCATE#2,1,1:PRINT#2,zon$:MOV
E x1,y1:DRAW x2,y2,farbe:DRAW x,y:
DRAW x1,y1                                      <5699>
1800 PRINT#2,"{@":back=0:GOTO 1770
                                                <1DF8>
1810 /OUT:DEG                                   <0C6D>
1820 GOSUB 2260:xd=x-xset+modi:yd=
y-yset+2:IF(xd*yd)/(4\modus))1920
THEN 1820                                       <60AC>
1830 /PART.IN,xset,yset,xd\modi,yd
\modi,modus:PRINT#2,CHR$(7):GOSUB
650:x1=x:y1=y:xa=ABS(xset-x):ya=AB
S(yset-y):ORIGIN x,y:x=0:y=0:PLOT
0,0,farbe:GOSUB 650:IF x THEN w=AT
N(y/x)ELSE w=90                                 <EC2E>
1840 pr=(xd\modi)*(yd\2):z=1:FOR a
=ya TO ya+yd-2 STEP 2:FOR b=xa TO
xa+xd-modi STEP modi:px=b*COS(w)-a
*SIN(w):py=b*SIN(w)+a*COS(w):PLOT
px,py,PEEK(&5000+pr-z):z=z+1                    <EA26>
1850 NEXT:NEXT:ORIGIN 0,0:x=x1:y=y
1:RAD:GOTO 1820                                 <2C34>
1860 /OUT                                       <0A8B>
1870 GOSUB 650:LOCATE#2,1,1:PRINT#
2,zon$:MOVE xset,yset:DRAW x,y,far
be:PRINT#2,"{@":GOTO 1870                       <53E6>
1880 /OUT:POKE 288,modus:FOR n=0 T
O 3:POKE 289+n,farb(n):NEXT:CLEAR:
DEFINT a-z:modus=PEEK(288):FOR n=0
 TO 3:farb(n)=PEEK(289+n):INK n,fa
rb(n):NEXT:farbe=1:ON ERROR GOTO 1
880:x=320:y=200:GOTO 30                         <C547>
1890 GOSUB 1080:MODE modus:/OUT:/M
ERGE,@b$(num):GOTO 1220                         <37C0>
1900 CLS:f=0:PRINT" S PRAY,":PRINT
" P EN OR":PRINT" M ULTISPRAY"                  <35E4>
1910 a$=UPPER$(INKEY$):IF a$="S"TH
EN rn=1 ELSE IF a$="P"THEN rn=0 EL
SE IF a$="M"THEN rn=1:f=1 ELSE GOT
O 1910                                          <6257>
1920 INPUT"WIDTH(1-5):",w:IF w<1 O
R w>5 THEN 1920                                 <309B>
1930 /OUT                                       <0A18>
1940 GOSUB 650:IF rn THEN 1980                  <18F9>
1950 LOCATE#2,1,1:PRINT#2,zon$:IF
gpap=0 THEN pap=1:IF farbe=1 THEN
pap=2 ELSE ELSE pap=0                           <5513>
1960 FOR n=1 TO w*2-1 STEP 2:MOVE
xset,yset+n-1:DRAW x+w-1,y+n-1,pap
:NEXT:POKE &B295,255:POKE &B297,&A
3:/PATTON,c+32,inki(farbe),inki(gp
ap)                                             <9FD3>
1970 FOR n=1 TO w*2-1 STEP 2:MOVE
xset,yset+n-1:DRAW x+w-1,y+n-1,far
be:NEXT:PRINT#2,"{@":xset=x:yset=y
:/PATTOFF:GOTO 1940                             <8D37>
1980 LOCATE#2,1,1:PRINT#2,zon$:FOR
 n=1 TO w^2+1:p=RND*(3*w):r=RND*(3
*w):IF f THEN fa=RND*modusf ELSE f
a=farbe:PLOT x+r*(3-modus),y+2*p,p
ap:POKE &B295,255:POKE &B297,&A3:/
PATTON,c+32,inki(farbe),inki(gpap)
                                                <EED6>
1990 PLOT x+r*(3-modus),y+p*2,fa:/
PATTOFF:NEXT:PRINT#2,"{@":GOTO 194
0                                               <4A2F>
2000 /OUT:/INVERT:GOTO 50                       <1971>
2010 CLS:PRINT" B END OR":PRINT" W
AVES ?"                                         <20E4>
2020 a$=UPPER$(INKEY$):IF a$="B"TH
EN a=1 ELSE IF a$<>"W"THEN 2020 EL
SE a=2                                          <40C8>
2030 /OUT                                       <0AE0>
```

```
2040 GOSUB 650:ORIGIN 0,200:FOR y=
-200 TO 200 STEP 2:/BEND.IN,y:ON a
 GOSUB 2070,2080                  (45CF)
2050 MOVE x+640,y:DRAW 640,y,farbe
:/BEND.OUT,x+640,y:MOVE 0,y:DRAW x
,y,farbe:IF NOT INKEY(66)THEN 2060
 ELSE NEXT                        (7450)
2060 ORIGIN 0,0:x=320:y=200:GOTO 2
040                               (214A)
2070 x=SQR(90000-y*y)-270:RETURN  (249C)
2080 DEG:x=20*SIN(y*5):RETURN     (1B10)
2090 /OUT:GOSUB 2260:xd=x-xset+mod
i:yd=y-yset+2:IF(xd*yd)\modi>1920
THEN 60                           (6116)
2100 /PART.IN,xset,yset,xd\modi,yd
\2,modus                          (3D8B)
2110 GOSUB 650:/MIRROR.OUT,x,y,xd\
modi,yd\2,modus:GOTO 2110         (466A)
2120 WINDOW 4,22,3,22:CLS:PEN 1:a$
=CHR$(150)+STRING$(17,154)+CHR$(15
6):b$=CHR$(149)+SPACE$(17)+CHR$(14
9):c$=CHR$(147)+STRING$(17,154)+CH
R$(153):PRINT a$;:FOR n=1 TO 18:PR
INT b$;:NEXT:PRINT c$;:WINDOW 5,19
,4,21:CLS:WINDOW#3,21,21,4,21     (AFC0)
2125 PRINT#3,"#GRAPHICS  DESIGNER#
";                                (2021)
2130 IF modus=1 THEN MOVE 319,352:
DRAWR 17,0,1:DRAWR 0,-289:DRAWR-17
,0:DRAWR 0,289 ELSE MOVE 159,352:D
RAWR 9,0,1:DRAWR 0,-289:DRAWR-9,0:
DRAWR 0,289                       (6971)
2140 RETURN                       (06E7)
2150 CLS:PRINT" REALLY (Y/N)"     (17BA)
2160 a$=UPPER$(INKEY$):IF a$="N"TH
EN 60 ELSE IF a$<>"Y"THEN 2160    (33D1)
2170 RETURN                       (0622)
2180 LOCATE#2,1,1:PRINT#2,"{A":MOV
E x-3+modus,y-2:DRAWR 0,50,1:DRAWR
 80+3-modus,0:DRAWR 0,-50:DRAWR-80
-3+modus,0:WHILE INKEY$="":r=1:WEN
D:MOVE x-3+modus,y-2:DRAWR 0,50,1:
DRAWR 80+3-modus,0:DRAWR 0,-50:DRA
WR-80-3+modus,0:PRINT#2,"{@"      (CB2F)
2185 GOSUB 2340                   (0A2C)
2190 IF s THEN s=0:RETURN ELSE 218
0                                 (1D28)
2200 CLS:PRINT" STANDARD":PRINT" X
OR":PRINT" AND":PRINT" OR":z=1    (3185)
2210 LOCATE 1,z:PRINT CHR$(243):WH
ILE INKEY$="":WEND:LOCATE 1,z:PRIN
T CHR$(32):IF NOT INKEY(0)OR NOT I
NKEY(72)THEN z=z-1                (53A9)
2220 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN z=z+1                      (2513)
2230 IF NOT INKEY(18)OR NOT INKEY(
76)THEN 2250                      (1EBF)
2240 z=MAX(1,z):z=MIN(4,z):GOTO 22
10                                (2A57)
2250 zmo=z-1:GOTO 60              (18AD)
2260 m=0:s=0                      (1298)
2270 IF m=0 THEN IF dar=1 THEN f1=
TEST(x,y):f2=TESTR(4,-4):f3=TESTR(
2,-2):f4=TESTR(2,-2):f5=TEST(x+2,y
):f6=TEST(x+4,y):f7=TEST(x,y-2):f8
=TEST(x,y-4)ELSE PRINT#2,"{A";:MOV
E x,0:DRAW x,400,farbe:MOVE 0,y:DR
AW 640,y:GOTO 2290 ELSE GOSUB 2320
:GOTO 2290                        (0751)
2280 modus=4\modus:PLOT x,y,farbe
:PLOTR 4,-4,farbe:PLOTR 2,-2,farbe
:PLOTR 2,-2,farbe:PLOT x+2,y,farbe
:PLOT x+4,y,farbe:PLOT x,y-2,farbe
:PLOT x,y-4,farbe:IF farbe=f1 THEN
 PLOT x,y,(farbe+1)MOD modus      (F1BD)
2290 BORDER farb(farbe)           (18A1)
2300 WHILE INKEY$="":r=1:WEND:IF m
=0 THEN IF dar=1 THEN PLOT x,y,f1:
PLOTR 4,-4,f2:PLOTR 2,-2,f3:PLOTR
2,-2,f4:PLOT x+2,y,f5:PLOT x+4,y,f
6:PLOT x,y-2,f7:PLOT x,y-4,f8 ELSE
 MOVE x,0:DRAW x,400:MOVE 0,y:DRAW
 640,y:PRINT#2,"{@";ELSE GOSUB 233
0                                 (F348)
2310 GOSUB 2340:IF s THEN s=0:IF m
 AND xset>0 AND yset>0 THEN m=0:RE
TURN ELSE xset=x:yset=y:m=1:GOTO 2
270 ELSE 2270                     (75F9)
2320 PRINT#2,"{A";:MOVE xset-3+mod
us,yset-2:DRAW xset-3+modus,y+2,fa
rbe:DRAW x+3-modus,y+2:DRAW x+3-mo
dus,yset-2:DRAW xset-3+modus,yset-
2:RETURN                          (A8D2)
2330 MOVE xset-3+modus,yset-2:DRAW
 xset-3+modus,y+2,farbe:DRAW x+3-m
odus,y+2:DRAW x+3-modus,yset-2:DRA
W xset-3+modus,yset-2:PRINT#2,"{@"
;:RETURN                           (A847)
2340 IF INKEY(0)=0 OR INKEY(72)=0
THEN y=y+2*r                      (2C3F)
2350 IF INKEY(2)=0 OR INKEY(73)=0
THEN y=y-2*r                      (2CE3)
2360 IF NOT INKEY(8)OR NOT INKEY(7
4)THEN x=x-(2\modus)*r            (35E6)
2370 IF NOT INKEY(1)OR NOT INKEY(7
5)THEN x=x+(2\modus)*r            (3540)
2380 y=MIN(y,399):y=MAX(y,0):x=MIN
(x,639):x=MAX(x,0):IF NOT INKEY(18
)OR NOT INKEY(76)THEN s=1:r=0     (6C73)
2390 IF NOT INKEY(66)THEN 50      (14FE)
2400 IF NOT INKEY(47)THEN xset=x:y
set=y:PLOT x,y,farbe              (3FEE)
2410 IF NOT INKEY(9)THEN/IN       (1340)
2420 IF NOT INKEY(79)THEN/OUT     (15BF)
2430 IF NOT INKEY(68)THEN IF dar=1
 THEN dar=0 ELSE dar=1            (328F)
2440 IF INKEY(0)=32 OR INKEY(72)=3
2 THEN farbe=farbe+1              (31C9)
2450 IF INKEY(2)=32 OR INKEY(73)=3
2 THEN farbe=farbe-1              (311A)
2460 farbe=ABS(farbe)MOD modusf:LO
CATE#2,1,1:r=r+1:RETURN           (3B94)
2470 /OUT                         (0A52)
2480 GOSUB 2260:IF x<xset OR y<yse
t THEN 2480 ELSE i1=farbe:xd=x-xse
t:yd=y-yset                       (6624)
2490 GOSUB 650:i2=farbe           (199C)
2500 /CHANGE,xset,yset,xd,yd,i1,i2
                                  (35BC)
2510 GOTO 2480                    (0A54)
```

```
1 '******************************* (2506)
2 '*       DATALADER G2.LAD       * (25E2)
3 '*        ERZEUGT G2.BIN        * (2510)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464          ME* (2593)
7 '******************************* (2512)
10 DATA 01,13,AF,21,0F,AF,CD,D1,BC
,3E,43A                           (287B)
20 DATA C9,32,00,AF,C9,00,00,00,00
,27,29A                           (28CE)
30 DATA AF,C3,42,AF,C3,4E,AF,C3,5A
,AF,5EF                           (28D5)
40 DATA C3,64,AF,C3,6E,AF,C3,80,AF
,49,5F1                           (2846)
50 DATA CE,4F,55,D4,47,50,45,CE,47
,50,487                           (28D0)
60 DATA 41,50,45,D2,43,4F,50,59,43
,48,36E                           (28A6)
70 DATA D2,46,49,4C,CC,00,21,00,C0
,11,36B                           (28CD)
80 DATA 00,60,01,FF,3F,ED,B0,C9,21
,00,426                           (2830)
90 DATA 60,11,00,C0,01,FF,3F,ED,B0
,C9,4D6                           (2834)
100 DATA FE,01,C0,DD,7E,00,CD,DE,B
B,C9,649                          (28CE)
110 DATA FE,01,C0,DD,7E,00,CD,E4,B
B,C9,64F                          (2804)
120 DATA FE,01,C0,DD,6E,00,DD,66,0
1,23,471                          (2871)
130 DATA 4E,23,46,CD,60,BB,02,C9,D
F,84,4CD                          (287E)
140 DATA AF,C9,87,AF,FE,7B,CD,F6,1
7,F5,6F6                          (283E)
150 DATA CD,FC,15,CD,27,18,CD,86,0
C,C1,50A                          (28F0)
160 DATA B8,C8,32,5F,B0,CD,1A,16,C
D,FF,58A                          (2838)
170 DATA 16,D2,15,B0,E5,FD,E1,D5,D
D,E1,703                          (28BB)
180 DATA CD,A9,0B,3A,5F,B0,47,FD,2
3,CD,4FE                          (28A1)
190 DATA 2D,0C,3A,34,B3,FD,95,38,0
5,78,3A1                          (2845)
200 DATA AE,A1,28,EF,FD,2B,CD,13,0
C,51,4CB                          (28E3)
210 DATA 59,E5,C5,CB,01,DC,05,0C,7
8,42,476                          (2891)
220 DATA AE,A1,57,20,0A,B0,28,07,D
D,2B,3B7                          (28EC)
230 DATA DF,25,B0,DD,23,C1,E1,E5,C
5,CB,6CB                          (2827)
240 DATA 09,DC,F9,0B,78,43,AE,A1,5
F,20,472                          (2868)
250 DATA 0A,B0,28,07,DD,23,DF,25,B
0,DD,47A                          (2812)
260 DATA 2B,C1,E1,CD,00,A0,00,00,0
0,00,33A                          (286B)
270 DATA 3A,36,B3,FD,95,30,0A,FD,2
B,CD,4E4                          (2866)
280 DATA 13,0C,78,AE,A1,28,B4,2A,6
0,B0,3FC                          (2810)
290 DATA 7D,B4,C8,2B,22,60,B0,DF,4
D,B0,532                          (2816)
300 DATA C3,9E,AF,28,B0,FD,D5,2A,8
9,AE,61B                          (2806)
310 DATA 01,03,00,CD,18,F6,38,17,E
B,22,33B                          (2842)
320 DATA 89,AE,DD,E5,D1,FD,E5,C1,2
B,72,70A                          (28CE)
330 DATA 2B,73,2B,71,2A,60,B0,23,2
2,60,319                          (2861)
340 DATA B0,D1,C9,50,B0,FD,2A,89,A
E,2B,5D3                          (287C)
350 DATA 56,2B,5E,2B,22,89,AE,6E,2
6,00,2F7                          (287D)
360 DATA C9,00,00,00,00,00,0C9    (1C3F)
370 dat=0:sz=0:dz=10              (1E30)
380 FOR adr=&AF00 TO &B063        (14DE)
390 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35C6)
400 sz=sz+byte                    (1808)
410 POKE adr,byte                 (14F4)
420 IF dat<10 AND adr<&B063 THEN 4
60                                (1EA3)
430 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2B89)
440 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30B3)
450 dz=dz+10:sz=0:dat=0           (24A5)
460 NEXT                          (065B)
470 SAVE"G2.BIN",B,&AF00,&163     (1B01)
```

```
1 '******************************* (2506)
2 '*       DATALADER G3A.LAD      * (2588)
3 '*        ERZEUGT G3A.BIN       * (2587)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464          ME* (2593)
7 '******************************* (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                           (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                           (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                           (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                           (28E4)
```

CPC Graphic Designer

```
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,00,70,80,80,80
,80,270                                 (2844)
80 DATA 80,70,00,0E,01,01,01,01,01
,0E,111                                 (28F8)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,00,00,10,00,00,00,7E,0
0,00,08E                                (2859)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
0,88,2F2                                (2857)
140 DATA 88,00,88,88,70,00,10,10,1
0,00,238                                (2851)
150 DATA 10,10,10,00,70,08,08,70,8
0,80,220                                (28F3)
160 DATA 70,00,70,08,08,70,08,08,7
0,00,1E0                                (28B6)
170 DATA 88,88,88,70,08,08,08,00,7
0,80,310                                (2893)
180 DATA 80,70,08,08,70,00,70,80,8
0,70,350                                (28E4)
190 DATA 88,88,70,00,70,08,08,00,0
8,08,210                                (283A)
200 DATA 08,00,70,88,88,70,88,88,7
0,00,378                                (283A)
210 DATA 70,88,88,70,08,08,70,00,0
0,00,270                                (28E0)
220 DATA 10,00,00,00,00,10,00,00,1
8,18,050                                (28DE)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,00,38,00,00,3
8,00,07C                                (2866)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,70,88,88,70,8
8,88,51A                                (2865)
280 DATA 88,00,F0,88,88,70,88,88,F
0,00,4F8                                (28D0)
290 DATA 70,80,80,80,80,80,70,00,F
0,88,4D8                                (2892)
300 DATA 88,00,88,88,F0,00,70,80,8
0,70,468                                (2816)
310 DATA 80,80,70,00,70,80,80,70,8
0,80,450                                (282F)
320 DATA 80,00,70,80,80,00,88,88,7
0,00,370                                (28F0)
330 DATA 88,88,88,70,88,88,88,00,1
0,10,3C0                                (2857)
340 DATA 10,00,10,10,10,00,10,10,1
0,00,070                                (285F)
350 DATA 10,10,60,00,A0,A0,A0,C0,A
0,A0,460                                (28E6)
360 DATA A0,00,80,80,80,00,80,80,7
0,00,390                                (28DB)
370 DATA 70,A8,A8,00,88,88,88,00,7
0,88,450                                (2893)
380 DATA 88,00,88,88,00,00,70,88,8
8,00,318                                (280E)
390 DATA 88,88,70,00,70,88,88,70,8
0,80,470                                (281F)
400 DATA 80,00,70,88,88,00,A8,A8,7
0,00,3C0                                (2821)
410 DATA 70,88,88,70,A0,A0,A0,00,7
0,80,4C0                                (28A9)
420 DATA 80,70,08,08,70,00,EE,10,1
0,00,27E                                (28BE)
430 DATA 10,10,10,00,88,88,88,00,8
8,88,2D8                                (2889)
440 DATA 70,00,90,90,90,00,90,90,6
0,00,3A0                                (2886)
450 DATA 88,88,88,00,A8,A8,70,00,9
0,90,478                                (28A6)
460 DATA 90,60,90,90,90,00,88,88,8
8,70,4A8                                (288C)
470 DATA 20,20,20,00,70,08,08,70,8
0,80,250                                (280B)
480 DATA 70,00,3C,30,30,30,30,30,3
C,00,1D8                                (2877)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 00,10,08,08,14,24,3A,41,0
0,40,113                                (28F5)
540 DATA 5C,62,41,41,62,1C,00,00,0
E,31,1FD                                (282C)
550 DATA 40,40,30,0F,C0,30,0C,32,4
1,41,26F                                (282D)
560 DATA 21,1E,00,00,3E,42,44,3F,2
0,1F,181                                (2843)
570 DATA 00,18,24,22,20,38,20,20,0
0,1C,112                                (2898)
580 DATA 22,40,40,3E,02,0C,40,40,5
C,62,22C                                (28CA)
590 DATA 42,44,48,48,00,18,00,10,0
8,08,14E                                (28D0)
600 DATA 08,10,00,00,30,08,04,04,0
4,38,094                                (2892)
610 DATA 00,40,24,28,30,28,24,42,0
0,00,14A                                (28E9)
620 DATA 40,20,20,20,42,3C,00,36,4
9,49,1E6                                (286B)
630 DATA 49,49,49,2A,00,42,22,32,2
A,26,1EB                                (284D)
640 DATA 22,21,00,00,1C,22,41,41,4
2,3C,181                                (28A0)
650 DATA 00,5C,62,41,41,62,5C,40,0
0,1D,25B                                (28A6)
660 DATA 23,41,41,23,1D,01,00,5C,6
2,42,1E6                                (2870)
670 DATA 7C,48,44,40,00,1E,20,1E,0
1,41,1E6                                (28F9)
680 DATA 21,1E,00,7F,08,08,10,10,0
8,04,0FA                                (28CC)
690 DATA 00,71,11,21,21,41,43,3D,0
0,44,1C9                                (28DE)
700 DATA 42,42,42,44,58,60,00,49,2
5,25,255                                (28FF)
710 DATA 49,49,4B,35,00,41,63,14,0
8,14,1E6                                (28BD)
720 DATA 22,63,00,42,21,11,0A,04,0
4,18,123                                (28CC)
730 DATA 00,3F,42,04,08,10,21,7E,0
E,18,162                                (28C5)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                (28B5)
770 DATA 00,00,00,00,00,00,00,00,0
0,000                                   (256F)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3A.BIN",B,&A000,&300          (1CBE)
```

```
1 '****************************** (2506)
2 '*      DATALADER G3B.LAD       * (259A)
3 '*       ERZEUGT G3B.BIN        * (2593)
4 '*    WIRD BENOETIGT FUER       * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*          CPC 464          ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,06
,06,00C                                 (2816)
20 DATA 0C,0C,18,00,00,18,1B,1B,36
,00,0B4                                 (28E4)
30 DATA 00,00,00,00,1B,1B,7F,36,FE
,6C,255                                 (289F)
40 DATA 6C,6C,06,0F,2C,1E,1A,7C,7C
,18,261                                 (2824)
50 DATA 00,31,66,0C,30,66,66,C6,0E
,1B,28E                                 (2849)
60 DATA 1C,3B,DC,CC,CC,76,06,06,18
,00,365                                 (286C)
70 DATA 00,00,00,00,03,06,18,18,30
,18,081                                 (283C)
80 DATA 18,0C,0C,06,06,06,0C,18,18
,30,0AE                                 (28CC)
90 DATA 00,19,1E,7F,3C,66,66,00,00
,06,1C4                                 (2885)
100 DATA 0C,3F,18,18,18,00,00,00,0
0,00,093                                (2832)
110 DATA 00,18,18,18,00,00,00,3F,0
0,00,087                                (2825)
120 DATA 00,00,00,00,00,00,00,18,1
8,18,048                                (280A)
130 DATA 01,03,0C,18,60,C0,C0,80,1
F,31,2D8                                (2842)
140 DATA 67,6B,E6,C6,C6,7C,06,0E,0
C,0C,3EC                                (2830)
150 DATA 18,18,18,7E,0F,19,03,1E,6
0,66,1D5                                (2868)
160 DATA 66,7E,0F,19,03,0E,06,66,6
6,3C,22B                                (2812)
170 DATA 07,0F,36,66,FE,0C,0C,1E,1
F,18,21D                                (28CD)
180 DATA 30,3E,06,66,66,3C,0F,19,3
0,3E,212                                (283F)
190 DATA 66,66,66,3C,1F,19,03,06,1
8,18,1DF                                (283A)
200 DATA 18,18,0F,19,33,1E,66,66,6
6,3C,217                                (28EE)
210 DATA 0F,19,33,1F,06,66,66,3C,0
0,00,188                                (284A)
220 DATA 0C,0C,00,18,18,18,00,00,0
C,0C,078                                (28CC)
230 DATA 00,18,18,18,03,06,18,30,3
0,18,0E1                                (28FD)
240 DATA 18,0C,00,00,3F,00,00,7E,7
E,00,15F                                (2885)
250 DATA 18,0C,0C,06,18,30,30,60,0
F,19,136                                (287B)
260 DATA 33,06,18,00,00,18,1F,31,6
F,6F,197                                (28E1)
270 DATA DE,C0,C0,7C,06,0F,33,33,7
E,66,439                                (28DC)
280 DATA 66,66,3F,19,33,3E,66,66,6
6,FC,3C3                                (28DC)
290 DATA 0F,19,60,60,C0,66,66,3C,3
E,1B,309                                (28B5)
300 DATA 33,33,66,6C,6C,F8,3F,18,3
4,3C,363                                (28F2)
310 DATA 68,62,62,FE,3F,18,34,3C,6
8,60,3B9                                (282F)
320 DATA 60,F0,0F,19,60,60,CE,66,6
6,3E,410                                (2864)
330 DATA 19,19,33,3F,66,66,66,66,1
F,06,261                                (28E0)
340 DATA 0C,0C,18,18,18,7E,07,03,0
6,06,0F4                                (28A4)
350 DATA CC,CC,CC,78,39,19,36,3C,6
C,66,472                                (28C9)
360 DATA 66,E6,3C,18,30,30,62,66,6
6,FE,42C                                (2829)
370 DATA 31,3B,7F,7F,D6,C6,C6,C6,3
1,39,4FC                                (281E)
380 DATA 7B,6F,CE,C6,C6,C6,0E,1B,6
3,63,4F9                                (2898)
```

```
390 DATA C6,6C,6C,38,3F,19,33,3E,6
0,60,35F                                (28B7)
400 DATA 60,F0,0E,1B,63,63,DA,CC,C
C,76,527                                (281A)
410 DATA 3F,19,33,3E,6C,66,66,E6,0
F,19,30F                                (28F5)
420 DATA 30,1E,06,66,66,3C,1F,16,0
C,0C,1A9                                (2853)
430 DATA 18,18,18,3C,19,19,33,33,6
6,66,1E8                                (2873)
440 DATA 66,3C,19,19,33,33,66,3C,3
C,18,230                                (2882)
450 DATA 31,31,63,6B,FE,EE,EE,C6,3
1,1B,51C                                (28EC)
460 DATA 1C,1C,6C,C6,C6,C6,19,19,3
3,1E,379                                (2842)
470 DATA 18,18,18,3C,3F,31,46,0C,3
2,66,1DE                                (288C)
480 DATA 66,FE,0F,0C,18,18,30,30,3
0,3C,27B                                (28FD)
490 DATA 30,18,18,0C,0C,06,06,02,0
F,03,098                                (28CA)
500 DATA 06,06,0C,0C,0C,3C,06,0F,3
F,0C,0CC                                (2890)
510 DATA 18,18,18,18,00,00,00,00,0
0,00,060                                (282E)
520 DATA 00,00,0C,06,06,00,00,00,0
0,00,018                                (285B)
530 DATA 00,00,3C,06,7C,CC,CC,76,3
8,18,31C                                (2859)
540 DATA 3E,33,66,66,66,DC,00,00,1
E,33,2D0                                (28FF)
550 DATA 60,66,66,3C,07,03,3E,66,C
C,CC,3AE                                (28BF)
560 DATA CC,76,00,00,1E,33,7E,60,6
0,3C,30D                                (2842)
570 DATA 07,0D,18,3C,30,30,30,78,0
0,00,170                                (2833)
580 DATA 1F,33,66,3E,3E,06,38,18,3
6,3B,1FB                                (28BA)
590 DATA 66,66,66,E6,06,00,1C,0C,1
8,18,276                                (2887)
600 DATA 18,3C,01,00,07,03,06,66,6
6,66,197                                (2885)
610 DATA 38,18,33,36,78,6C,6C,E6,0
E,06,303                                (285D)
620 DATA 0C,0C,18,18,18,3C,00,00,3
6,7F,151                                (287F)
630 DATA D6,D6,D6,C6,00,00,6E,33,6
6,66,4B5                                (285E)
640 DATA 66,66,00,00,1E,33,66,66,6
6,3C,28B                                (285A)
650 DATA 00,00,6E,33,66,7C,7C,60,0
0,00,25F                                (2826)
660 DATA 3B,66,CC,7C,7C,0C,00,00,6
E,3B,31A                                (28F3)
670 DATA 60,60,60,F0,00,00,1E,30,3
C,06,2A0                                (289B)
680 DATA 06,7C,0C,0C,3E,18,30,36,3
6,1C,1A8                                (289B)
690 DATA 00,00,33,33,66,66,66,3E,0
0,00,1D6                                (2898)
700 DATA 33,33,66,3C,3C,18,00,00,6
3,6B,22A                                (282B)
710 DATA D6,FE,FE,6C,00,00,63,36,3
8,6C,47B                                (28D9)
720 DATA 6C,C6,00,00,33,33,66,3E,3
E,06,280                                (28A6)
730 DATA 00,00,3F,26,18,32,32,7E,0
3,06,168                                (289C)
740 DATA 0C,38,18,18,18,0E,06,06,0
C,0C,0BE                                (2849)
750 DATA 18,18,18,18,1C,06,0C,07,1
8,18,0C5                                (28D5)
760 DATA 18,70,1D,37,00,00,00,00,0
0,00,0DC                                (287E)
770 DATA 00,36,36,07,00,00,7E,19,0
0,10A                                   (2572)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3B.BIN",B,&A000,&300          (1C2D)
```

```
1 '******************************* (2506)
2 '*       DATALADER G3C.LAD       * (25AA)
3 '*        ERZEUGT G3C.BIN        * (25B2)
4 '*    WIRD BENOETIGT FUER        * (25CD)
5 '*   CPC GRAPHICS DESIGNER       * (25E5)
6 '*          CPC 464          ME* (2593)
7 '******************************* (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
C,86,2FC                                (2825)
140 DATA 8A,92,A2,C2,7C,00,08,18,0
8,08,32C                                (286D)
150 DATA 08,08,3E,00,3C,42,02,3C,4
0,42,18C                                (28DB)
160 DATA 7E,00,3C,42,02,1C,02,42,3
C,00,19A                                (287E)
170 DATA 0C,14,24,44,FE,04,1E,00,7
E,42,268                                (281D)
180 DATA 40,7C,02,42,3C,00,3C,42,4
0,7C,276                                (2835)
190 DATA 42,42,3C,00,7E,42,02,04,0
8,08,196                                (2881)
200 DATA 08,00,3C,42,42,3C,42,42,3
C,00,1C4                                (28A7)
210 DATA 3C,42,42,3E,02,42,3C,00,0
0,00,17E                                (2864)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,18,24,42,42,7
E,42,39A                                (28DA)
280 DATA 42,00,FC,42,42,7C,42,42,F
C,00,3BE                                (28E4)
290 DATA 3C,42,80,80,80,42,3C,00,F
8,44,3B8                                (2831)
300 DATA 42,42,42,44,F8,00,FE,42,4
8,78,402                                (2855)
310 DATA 48,42,FE,00,FE,42,48,78,4
8,40,410                                (28E5)
320 DATA E0,00,3C,42,80,80,8E,42,3
E,00,36C                                (28EC)
330 DATA 42,42,42,7E,42,42,42,00,7
C,10,296                                (284C)
340 DATA 10,10,10,10,7C,00,1E,04,0
4,04,0E6                                (280E)
350 DATA 84,84,78,00,C2,42,44,78,4
4,42,3C6                                (2837)
360 DATA C2,00,F0,40,40,40,42,42,F
E,00,3F4                                (28D4)
370 DATA 82,C6,AA,92,82,82,82,00,8
2,C2,54E                                (281D)
380 DATA A2,92,8A,86,82,00,38,44,8
2,82,446                                (287A)
390 DATA 82,44,38,00,FC,42,42,7C,4
0,40,37A                                (28C3)
400 DATA F0,00,38,44,82,82,92,8C,7
4,00,402                                (28E2)
410 DATA FC,42,42,7C,48,44,E2,00,3
C,42,3E8                                (282E)
420 DATA 40,3C,02,42,3C,00,7C,54,1
0,10,1EC                                (28B7)
430 DATA 10,10,38,00,42,42,42,42,4
2,42,1E4                                (2879)
440 DATA 3C,00,42,42,42,42,42,24,1
8,00,1C2                                (2884)
450 DATA 82,82,82,92,AA,C6,82,00,8
2,44,4D0                                (28DB)
460 DATA 38,38,44,82,82,00,44,44,4
4,28,2AC                                (284E)
470 DATA 10,10,38,00,FE,84,88,10,2
2,42,2D6                                (2861)
480 DATA FE,00,3C,30,30,30,30,30,3
C,00,266                                (28BA)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 10,38,38,64,7C,C2,C2,00,7
8,6C,3C8                                (2884)
540 DATA 6C,78,6C,6C,78,00,1C,36,6
0,60,346                                (2822)
550 DATA 60,36,1C,00,70,68,6C,6C,6
C,68,336                                (2822)
560 DATA 70,00,7C,60,60,78,60,60,7
C,00,360                                (28C9)
570 DATA 7C,60,60,78,60,60,60,00,1
C,36,326                                (2821)
580 DATA 60,60,6E,36,1C,00,62,62,6
2,7E,324                                (282F)
590 DATA 62,62,62,00,18,18,18,18,1
8,18,1B6                                (283D)
600 DATA 18,00,7E,06,06,06,66,6E,3
C,00,1B8                                (287D)
610 DATA 62,66,6C,78,6C,66,62,00,6
0,60,3A0                                (28FD)
620 DATA 60,60,60,60,7C,00,83,C7,E
F,DB,510                                (289D)
630 DATA C3,C3,C3,00,66,66,76,76,6
E,6E,4DD                                (2884)
640 DATA 66,00,38,6C,C6,C6,C6,6C,3
8,00,400                                (2859)
650 DATA 7C,66,66,66,7C,60,60,00,3
8,6C,38E                                (280D)
660 DATA C6,C6,CE,6E,3B,00,7C,66,6
6,66,4B1                                (2885)
670 DATA 7C,66,66,00,38,6C,60,38,0
C,6C,2FC                                (28F4)
680 DATA 38,00,7E,18,18,18,18,18,1
8,00,146                                (28C7)
690 DATA 6C,6C,6C,6C,6C,6C,38,00,6
C,6C,398                                (28CA)
700 DATA 6C,6C,6C,38,10,00,C6,C6,C
6,D6,4B4                                (28DF)
710 DATA D6,FE,6C,00,C6,6C,38,38,6
C,C6,514                                (283A)
720 DATA 82,00,6C,6C,6C,38,18,18,1
8,00,246                                (2880)
```

CPC Graphic Designer

```
730 DATA 7E,06,0C,18,30,60,7E,00,0
E,18,1DC                                (2811)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                                   (2511)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3C.BIN",B,&A000,&300          (1CDE)
```

```
1 '****************************** (2506)
2 '*      DATALADER G3D.LAD     * (257A)
3 '*       ERZEUGT G3D.BIN      * (25D1)
4 '*    WIRD BENOETIGT FUER     * (25CD)
5 '*  CPC GRAPHICS DESIGNER     * (25E5)
6 '*          CPC 464         ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
C,4C,2C2                                (2804)
140 DATA 4C,54,74,64,7C,00,10,10,1
0,30,254                                (2871)
150 DATA 30,30,30,00,3E,22,02,3E,3
0,32,192                                (286C)
160 DATA 3E,00,3E,22,02,1E,06,26,3
E,00,128                                (2846)
170 DATA C8,C8,C8,88,FC,08,08,00,7
E,42,4AC                                (283F)
180 DATA 40,7E,06,46,7E,00,7E,42,4
0,7E,306                                (284C)
190 DATA 62,62,7E,00,3C,04,04,0C,0
C,0C,1AA                                (285A)
200 DATA 0C,00,7C,44,44,7C,64,64,7
C,00,2D0                                (2831)
210 DATA 7C,44,44,7C,0C,4C,7C,00,0
0,00,254                                (2882)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,7C,44,44,7C,6
4,64,462                                (2883)
280 DATA 64,00,7C,44,44,78,64,64,7
C,00,324                                (284F)
290 DATA 7C,44,40,60,60,64,7C,00,7
8,44,35C                                (289D)
300 DATA 44,64,64,64,78,00,7C,40,4
0,7C,360                                (28EB)
310 DATA 60,60,7C,00,7C,40,40,7C,6
0,60,374                                (2819)
320 DATA 60,00,7C,44,40,6C,64,64,7
C,00,310                                (285B)
330 DATA 44,44,44,7C,64,64,64,00,1
0,10,294                                (28CB)
340 DATA 10,18,18,18,18,00,7C,04,0
4,0C,100                                (287C)
350 DATA 0C,4C,7C,00,48,48,48,78,6
4,64,2EC                                (2811)
360 DATA 64,00,40,40,40,60,60,60,7
C,00,2C0                                (284E)
370 DATA FE,92,92,D2,D2,D2,D2,00,7
C,44,62A                                (28F2)
380 DATA 44,64,64,64,64,00,7E,46,4
6,42,320                                (284A)
390 DATA 42,42,7E,00,7C,44,44,7C,6
0,60,342                                (2877)
400 DATA 60,00,7E,46,46,42,42,4E,7
F,00,2BB                                (2837)
410 DATA 7C,44,44,7E,62,62,62,00,7
C,44,368                                (2866)
420 DATA 40,7C,0C,4C,7C,00,7C,10,1
0,18,244                                (28F7)
430 DATA 18,18,18,00,44,44,44,64,6
4,64,240                                (28C0)
440 DATA 7C,00,62,62,62,66,24,24,3
C,00,28C                                (28B4)
450 DATA D2,D2,D2,D2,92,92,FE,00,3
2,32,5CE                                (2864)
460 DATA 32,1C,22,22,22,00,64,64,6
C,68,250                                (28E5)
470 DATA 38,10,10,00,7C,44,04,7C,6
0,64,25C                                (2896)
480 DATA 7C,00,3C,30,30,30,30,30,3
C,00,1E4                                (28E5)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 00,7C,44,40,7C,64,7C,00,4
0,40,2DC                                (2802)
540 DATA 7C,64,64,64,7C,00,00,7E,4
2,60,344                                (28F2)
550 DATA 60,62,7E,00,04,04,7C,4C,4
C,4C,2A8                                (2803)
560 DATA 7C,00,00,7C,44,7C,60,64,7
C,00,2F8                                (2840)
570 DATA 38,20,70,20,30,30,30,00,0
0,7C,1F4                                (28A1)
580 DATA 44,64,64,7C,04,7C,40,40,7
C,44,348                                (28DA)
590 DATA 64,64,64,00,10,00,10,10,1
8,18,18C                                (28B6)
600 DATA 18,00,04,00,04,04,0C,0C,2
C,3C,0A4                                (28B7)
610 DATA 40,40,48,48,70,68,68,00,1
0,10,270                                (2865)
620 DATA 10,18,18,18,18,00,00,00,F
E,92,200                                (28BC)
630 DATA D2,D2,D2,00,00,00,7C,44,6
4,64,3FE                                (2885)
640 DATA 64,00,00,00,7C,44,64,64,7
C,00,268                                (28EA)
650 DATA 00,00,7C,44,44,7C,60,60,0
0,00,240                                (28D9)
660 DATA 7C,44,44,7C,0C,0C,00,00,7
C,44,258                                (2887)
670 DATA 60,60,60,00,00,00,3C,20,3
C,0C,1C4                                (2857)
680 DATA 3C,00,20,70,20,30,30,30,3
8,00,1B4                                (28B9)
690 DATA 00,00,44,44,64,64,7C,00,0
0,00,1CC                                (28DF)
700 DATA 62,62,66,24,18,00,00,00,9
2,92,28A                                (2858)
710 DATA D2,D2,FE,00,00,00,44,6C,3
8,6C,3F6                                (2809)
720 DATA 64,00,00,00,24,24,24,3C,0
C,3C,154                                (28C0)
730 DATA 00,00,7C,04,7C,40,7C,00,0
E,18,1DE                                (28B1)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                                   (2511)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3D.BIN",B,&A000,&300          (1C0D)
```

```
1 '****************************** (2506)
2 '*      DATALADER G3E.LAD     * (254A)
3 '*       ERZEUGT G3E.BIN      * (25F1)
4 '*    WIRD BENOETIGT FUER     * (25CD)
5 '*  CPC GRAPHICS DESIGNER     * (25E5)
6 '*          CPC 464         ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,0
0,38,232                                (2895)
140 DATA 21,21,09,31,03,3E,00,04,0
4,04,0C9                                (280B)
150 DATA 04,04,00,3F,00,18,31,03,1
E,10,0C1                                (28F0)
160 DATA 01,3F,00,18,31,03,08,01,0
3,1E,0B6                                (28B7)
170 DATA 00,02,12,32,00,73,00,0F,0
0,1D,0E5                                (283F)
180 DATA 11,00,38,01,03,1E,00,18,1
3,00,096                                (288B)
```

```
190 DATA 18,11,03,1E,00,19,31,03,0
6,04,0A1                                (280F)
200 DATA 04,0C,00,18,11,03,18,11,0
3,1E,086                                (28AD)
210 DATA 00,18,11,01,19,01,03,1E,0
0,00,065                                (289E)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,00,00,1C,11,0
1,1D,265                                (2830)
280 DATA 11,33,00,18,11,03,1A,11,0
3,7E,11C                                (28A8)
290 DATA 00,18,33,20,20,10,03,1E,0
0,10,0CC                                (2851)
300 DATA 10,11,11,11,02,7C,00,1D,1
1,04,0F3                                (28CD)
310 DATA 1C,14,01,7F,00,1D,11,04,1
C,14,112                                (2878)
320 DATA 08,78,00,19,33,20,20,15,0
1,1F,141                                (28B0)
330 DATA 00,11,11,01,19,11,11,33,0
0,07,098                                (28D8)
340 DATA 04,04,04,04,00,3F,00,03,0
2,02,056                                (285C)
350 DATA 02,22,06,3C,10,11,13,06,1
0,10,0C0                                (2860)
360 DATA 11,73,00,18,10,10,10,11,0
1,7F,15D                                (28B2)
370 DATA 00,01,01,01,29,29,21,63,0
0,01,0DA                                (2888)
380 DATA 01,21,21,21,21,63,00,10,3
0,21,149                                (284E)
390 DATA 21,03,06,1C,00,18,11,03,1
E,10,0A0                                (28A0)
400 DATA 00,78,00,10,30,21,21,21,0
2,3B,158                                (2872)
410 DATA 00,18,11,03,16,10,11,73,0
0,18,0EE                                (28CB)
420 DATA 13,00,18,01,03,1E,00,25,2
5,04,09B                                (280F)
430 DATA 04,04,00,1E,00,11,11,11,1
1,11,07B                                (28BB)
440 DATA 03,1E,00,11,11,11,11,03,0
6,0C,07A                                (284D)
450 DATA 00,21,21,21,01,11,31,63,0
0,03,10C                                (28E6)
460 DATA 06,04,10,30,21,63,00,11,1
1,03,0F3                                (28C4)
470 DATA 06,04,00,1E,00,39,63,46,0
C,19,12F                                (28FB)
480 DATA 01,7F,3C,30,30,30,30,30,3
C,00,1E8                                (28F4)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 18,24,5A,42,5A,5A,24,00,7
C,82,2AE                                (28CE)
540 DATA 5A,44,5A,82,7C,00,1C,22,4
C,50,2D0                                (287C)
550 DATA 4C,22,1C,00,78,84,52,5A,5
2,84,308                                (281A)
560 DATA 78,00,7C,82,5C,48,5C,82,7
C,00,374                                (288D)
570 DATA 7C,82,5C,48,50,88,70,00,3
C,42,368                                (2884)
580 DATA 9C,A4,BA,82,7C,00,24,5A,5
A,42,412                                (2851)
590 DATA 5A,5A,24,00,38,44,28,28,2
8,44,210                                (28B0)
600 DATA 38,00,1C,22,14,54,B4,84,7
8,00,28E                                (2899)
610 DATA 6C,92,44,48,44,92,6C,00,7
0,88,3C4                                (285B)
620 DATA 50,54,5A,82,7C,00,44,AA,9
2,82,3FE                                (283E)
630 DATA AA,BA,44,00,24,5A,4A,42,5
2,5A,35E                                (2881)
640 DATA 24,00,3C,42,99,99,99,42,3
C,00,2EB                                (28A6)
650 DATA 7C,82,42,5C,50,88,70,00,3
C,42,362                                (2805)
660 DATA 99,99,99,9D,41,3F,7C,82,4
2,44,46C                                (2889)
670 DATA 52,8A,76,00,3E,41,5E,41,3
D,41,2EE                                (2899)
680 DATA 3E,00,7C,82,6C,28,28,44,3
8,00,274                                (283F)
690 DATA 24,5A,5A,5A,5A,42,3C,00,4
4,AA,2F8                                (286A)
700 DATA AA,92,44,28,10,00,44,BA,A
A,82,3E2                                (280E)
710 DATA 92,AA,44,00,66,99,42,24,4
2,99,3C0                                (2847)
720 DATA 66,00,44,AA,92,44,28,28,1
0,00,28A                                (28DD)
730 DATA 7E,81,72,24,4E,81,7E,00,0
E,18,308                                (2814)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                                   (2511)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3E.BIN",B,&A000,&300          (1CFE)
```

```
1 '******************************** (2506)
2 '*       DATALADER G3F.LAD      * (25D9)
3 '*        ERZEUGT G3F.BIN       * (25DE)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*          CPC 464          ME* (2593)
7 '******************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
C,C6,33C                                (2844)
140 DATA CE,D6,E6,C6,7C,00,18,38,1
8,18,44C                                (2870)
150 DATA 18,18,7E,00,3C,66,06,3C,6
0,66,258                                (28C6)
160 DATA 7E,00,3C,66,06,1C,06,66,3
C,00,1EA                                (28B9)
170 DATA 1C,3C,6C,CC,FE,0C,1E,00,7
E,62,398                                (28B9)
180 DATA 60,7C,06,66,3C,00,3C,66,6
0,7C,302                                (284F)
190 DATA 66,66,3C,00,7E,66,06,0C,1
8,18,22E                                (2822)
200 DATA 18,00,3C,66,66,3C,66,66,3
C,00,264                                (28FF)
210 DATA 3C,66,66,3E,06,66,3C,00,0
0,00,1EE                                (28D1)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,00,1E,21,42,7
E,84,39D                                (2814)
280 DATA 84,00,00,1E,11,22,3C,44,F
8,00,24D                                (2814)
290 DATA 00,1E,22,40,40,84,78,00,0
0,1E,1DA                                (28F4)
300 DATA 11,22,22,44,F8,00,00,1F,1
1,20,1E1                                (285E)
310 DATA 38,42,FE,00,00,1F,11,20,3
8,40,240                                (2875)
320 DATA F0,00,00,1E,22,40,4C,84,7
8,00,2B8                                (2878)
330 DATA 00,22,22,44,7C,88,88,00,0
0,3E,252                                (284B)
340 DATA 08,10,10,20,F8,00,00,1F,0
2,04,165                                (2841)
350 DATA 44,88,78,00,00,32,14,28,3
0,48,22A                                (28C1)
360 DATA C8,00,00,20,20,40,40,80,F
C,00,304                                (2867)
370 DATA 00,22,36,5C,44,88,88,00,0
0,24,22C                                (2839)
380 DATA 24,58,58,90,90,00,00,1C,2
2,44,276                                (2807)
390 DATA 44,88,70,00,00,3E,11,22,3
C,40,229                                (2811)
400 DATA F0,00,00,1C,22,44,54,88,7
8,00,2C6                                (2888)
410 DATA 00,3E,11,22,3C,44,F4,00,0
0,1C,201                                (28A8)
420 DATA 24,40,38,88,70,00,00,FE,1
0,20,2C2                                (28E0)
430 DATA 20,40,40,00,00,22,22,44,4
4,88,1F4                                (28CD)
440 DATA 70,00,00,22,22,44,84,98,6
0,00,274                                (2828)
450 DATA 00,22,22,44,54,A8,D8,00,0
0,44,2A0                                (288C)
460 DATA 48,30,50,88,88,00,00,44,4
4,88,2E8                                (284F)
470 DATA 70,20,F8,00,00,7E,84,08,1
0,62,304                                (2875)
480 DATA FC,00,3C,30,30,30,30,30,3
C,00,264                                (28BE)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 00,00,3C,66,7E,66,66,00,0
0,00,1EC                                (2816)
```

CPC Graphic Designer

```
540 DATA 7C,66,7C,66,7C,00,00,00,3
E,60,2DE                          (28EA)
550 DATA 60,60,3E,00,00,00,7C,66,6
6,66,2AC                          (28C6)
560 DATA 7C,00,00,00,7E,60,78,60,7
E,00,2B0                          (28D8)
570 DATA 00,00,7E,60,7C,60,60,00,0
0,00,21A                          (2892)
580 DATA 3E,60,6E,66,3E,00,00,00,6
6,66,27C                          (2861)
590 DATA 7E,66,66,00,00,00,18,18,1
8,18,1AA                          (28A7)
600 DATA 18,00,00,00,06,06,66,66,3
C,00,12C                          (2850)
610 DATA 00,00,66,6C,78,6C,66,00,0
0,00,21C                          (287C)
620 DATA 60,60,60,60,7E,00,00,00,6
6,7E,2E2                          (28F3)
630 DATA 66,66,66,00,00,00,66,76,7
E,6E,2FA                          (2859)
640 DATA 66,00,00,00,3C,66,66,66,3
C,00,210                          (289E)
650 DATA 00,00,7C,66,7C,60,60,00,0
0,00,21E                          (28CB)
660 DATA 3E,63,6B,66,3B,00,00,00,7
C,66,28F                          (28F5)
670 DATA 7C,6C,66,00,00,00,3E,60,3
C,06,22E                          (285F)
680 DATA 7C,00,00,00,7E,18,18,18,1
8,00,15A                          (28D1)
690 DATA 00,00,66,66,66,66,3C,00,0
0,00,1D4                          (288D)
700 DATA 66,66,66,3C,18,00,00,00,6
6,66,252                          (28E5)
710 DATA 66,7E,66,00,00,00,66,3C,1
8,3C,240                          (28A0)
720 DATA 66,00,00,00,66,66,3C,18,1
8,00,19E                          (28FF)
730 DATA 00,00,7E,0C,18,30,7E,00,0
E,18,176                          (2801)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                          (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                          (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                          (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                             (2511)
780 dat=0:sz=0:dz=10              (1E65)
790 FOR adr=&A000 TO &A300        (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35FB)
810 sz=sz+byte                    (183D)
820 POKE adr,byte                 (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30EA)
860 dz=dz+10:sz=0:dat=0           (24DC)
870 NEXT                          (0692)
880 SAVE"G3F.BIN",B,&A000,&300    (1C6E)
```

```
1 '*******************************  (2506)
2 '*       DATALADER G3G.LAD     *  (25E9)
3 '*        ERZEUGT G3G.BIN      *  (252E)
4 '*    WIRD BENOETIGT FUER      *  (25CD)
5 '*   CPC GRAPHICS DESIGNER     *  (25E5)
6 '*           CPC 464        ME*  (2593)
7 '*******************************  (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                           (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                           (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                           (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                           (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,28
,6C,39A                           (288A)
60 DATA 28,66,CC,CC,66,00,18,18,30
,00,2EC                           (2805)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                           (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                           (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,18
,18,273                           (2882)
100 DATA 00,66,00,18,18,00,00,00,0
0,00,096                          (2820)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                          (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                          (285D)
130 DATA 06,0C,18,30,60,C0,80,00,6
C,C6,32C                          (281A)
140 DATA C6,C6,C6,C6,6C,00,18,58,1
8,18,424                          (2861)
150 DATA 18,18,5C,00,2C,66,06,2C,6
0,66,216                          (28E3)
160 DATA 6E,00,2C,66,06,0C,06,66,2
C,00,1AA                          (28DB)
170 DATA 0C,2C,6C,CC,EE,0C,0E,00,6
E,62,348                          (28F3)
180 DATA 60,6C,06,66,2C,00,2C,66,6
0,6C,2C2                          (2894)
190 DATA 66,66,2C,00,76,66,06,0C,1
8,18,216                          (2865)
200 DATA 18,00,2C,66,66,2C,66,66,2
C,00,234                          (285E)
210 DATA 34,66,66,36,06,66,34,00,0
0,00,1D6                          (286B)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                          (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                          (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                          (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                          (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                          (2813)
270 DATA DE,C0,7C,00,08,2C,66,66,6
E,66,3EE                          (28CF)
280 DATA 66,00,EC,66,64,6C,66,66,E
C,00,440                          (285B)
290 DATA 36,62,C0,C0,C0,62,36,00,E
8,6C,4C4                          (286D)
300 DATA 66,66,66,6C,E8,00,EE,62,6
4,6C,4A6                          (28F1)
310 DATA 64,62,EE,00,EE,62,64,6C,6
4,60,498                          (28AD)
320 DATA E0,00,2C,66,C0,C0,CE,66,2
E,00,454                          (287A)
330 DATA 66,66,66,6E,66,66,66,00,7
4,30,376                          (2807)
340 DATA 30,30,30,30,74,00,0E,0C,0
C,0C,166                          (285C)
350 DATA CC,CC,68,00,E6,66,6C,68,6
C,66,4F2                          (2895)
360 DATA E6,00,E0,60,60,60,62,66,E
E,00,49C                          (28A5)
370 DATA C6,CE,DE,DE,D6,C6,C6,00,C
6,E6,75E                          (2899)
380 DATA E6,CE,CE,C6,C6,00,28,44,C
6,C6,606                          (28FA)
390 DATA C6,44,28,00,EC,66,66,6C,6
0,60,416                          (2827)
400 DATA E0,00,6C,C6,C6,C6,CA,CC,6
6,00,59A                          (289E)
410 DATA EC,66,66,60,6C,66,E6,00,3
6,62,468                          (2824)
420 DATA 70,3C,0E,46,6C,00,F6,B2,3
0,30,374                          (28FD)
430 DATA 30,30,74,00,66,66,66,66,6
6,66,338                          (28DE)
440 DATA 2C,00,66,66,66,66,66,2C,0
8,00,25E                          (2822)
450 DATA C6,C6,C6,C6,EE,EE,C6,00,C
6,6C,6EC                          (2861)
460 DATA 28,28,6C,C6,C6,00,66,66,6
6,30,3AA                          (28ED)
470 DATA 18,18,3C,00,F2,C6,8C,18,3
2,66,360                          (2846)
480 DATA CE,00,3C,30,30,30,30,30,3
C,00,236                          (2898)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                          (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                          (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                          (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                          (28F1)
530 DATA 00,00,68,0C,6C,CC,6E,00,E
0,60,35A                          (280E)
540 DATA 6C,66,66,66,EC,00,00,00,3
6,62,322                          (285B)
550 DATA 60,62,36,00,0C,0C,6C,CC,C
C,CC,3E0                          (28EC)
560 DATA 6E,00,00,00,2C,66,6C,60,2
C,00,1F8                          (28B6)
570 DATA 16,32,30,74,30,30,74,00,0
0,00,1C0                          (286E)
580 DATA 36,66,66,36,06,74,E0,60,6
C,66,3C4                          (280C)
590 DATA 66,66,E6,00,18,00,58,18,1
8,18,26A                          (28DA)
600 DATA 5C,00,06,00,06,06,06,66,6
6,34,174                          (28D9)
610 DATA E0,60,66,6C,68,6C,E6,00,7
0,30,46C                          (28AB)
620 DATA 30,30,30,30,74,00,00,00,4
C,DE,25E                          (28C6)
630 DATA D6,C6,C6,00,00,00,EC,66,6
6,66,480                          (285B)
640 DATA 66,00,00,00,2C,66,66,66,2
C,00,1F0                          (28FE)
650 DATA 00,00,CC,66,66,6C,60,E0,0
0,00,344                          (288F)
660 DATA 66,CC,CC,6C,0C,0E,00,00,6
C,66,356                          (285A)
670 DATA 60,60,60,00,00,00,2C,60,2
C,06,1DE                          (28EF)
680 DATA 6C,00,60,60,6C,60,60,66,0
C,00,2CA                          (28AA)
690 DATA 00,00,66,66,66,66,36,00,0
0,00,1CE                          (2848)
700 DATA 66,66,66,2C,08,00,00,00,C
6,D6,302                          (2836)
710 DATA D6,DE,4C,00,00,00,C6,6C,2
8,6C,3C6                          (2834)
720 DATA C6,00,00,00,66,66,66,36,0
6,7C,2B0                          (285B)
730 DATA 00,00,66,4C,18,32,66,00,0
C,18,186                          (288D)
740 DATA 18,30,18,18,0C,00,18,18,1
8,00,0CC                          (28A5)
750 DATA 18,18,18,00,30,18,18,0C,1
8,18,0E4                          (28B3)
760 DATA 30,00,76,DC,00,00,00,00,0
0,00,182                          (281B)
770 DATA 2C,66,60,E8,60,66,EE,00,0
0,38E                             (25BD)
780 dat=0:sz=0:dz=10              (1E65)
790 FOR adr=&A000 TO &A300        (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35FB)
810 sz=sz+byte                    (183D)
820 POKE adr,byte                 (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30EA)
860 dz=dz+10:sz=0:dat=0           (24DC)
870 NEXT                          (0692)
880 SAVE"G3G.BIN",B,&A000,&300    (1C38)
```

```
1 '****************************** (2506)
2 '*       DATALADER G3H.LAD      * (25B9)
3 '*        ERZEUGT G3H.BIN       * (254E)
4 '*    WIRD BENOETIGT FUER       * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464         ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                           (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                           (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                           (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                           (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                           (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                           (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                           (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                           (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                           (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                          (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                          (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                          (285D)
130 DATA 06,0C,18,30,60,C0,80,00,1
C,3A,250                          (2854)
140 DATA 66,6E,76,66,5C,38,08,18,3
8,18,2B4                          (28A2)
150 DATA 18,18,3E,7C,1C,3A,46,1C,3
8,60,23A                          (28F3)
160 DATA 7E,7C,1C,3A,46,0C,1A,26,5
C,38,276                          (28F6)
170 DATA 0C,1C,2C,4C,FE,FC,0E,1C,3
E,7E,380                          (284A)
180 DATA 64,7C,7A,26,7C,38,1C,3A,6
4,7C,36A                          (28A4)
190 DATA 7A,66,7C,38,3E,7E,46,04,0
8,18,2BA                          (2848)
200 DATA 18,10,1C,3A,66,5C,3A,66,5
C,38,274                          (28BD)
210 DATA 1C,3A,66,5E,3E,26,5C,38,0
0,00,212                          (28E3)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                          (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                          (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                          (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                          (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                          (2813)
270 DATA DE,C0,7C,00,08,1C,3E,66,7
E,7E,3DE                          (282F)
280 DATA 66,64,3C,7A,46,44,7A,46,4
4,78,386                          (281C)
290 DATA 1C,3A,64,60,60,62,5C,38,3
C,7A,326                          (2895)
300 DATA 66,66,66,66,7C,78,3E,7C,6
0,7C,422                          (2856)
310 DATA 78,60,7E,7C,3E,7C,60,7C,7
8,60,440                          (28F1)
320 DATA 60,60,1C,3A,64,60,66,6E,5
E,3C,348                          (2818)
330 DATA 22,66,66,7E,7E,66,66,44,3
E,7C,3B4                          (28A7)
340 DATA 18,18,18,18,3E,7C,0E,1C,0
C,0C,15C                          (2879)
350 DATA 2C,6C,58,30,22,66,64,78,7
4,6A,362                          (2877)
360 DATA 66,44,20,60,60,60,60,60,7
E,7C,3A4                          (287B)
370 DATA 42,E6,EE,FE,F6,C6,C6,84,2
2,66,6A2                          (28A3)
380 DATA 76,6E,76,6E,66,44,1C,3A,6
6,66,394                          (28AB)
390 DATA 66,66,5C,38,3C,7A,66,7C,7
8,60,3D0                          (280F)
400 DATA 60,40,1C,3A,66,66,6E,76,5
E,3C,340                          (2848)
410 DATA 3C,7A,66,7C,7C,6A,66,44,1
E,3E,384                          (28A0)
420 DATA 34,3E,3E,16,3E,3C,3E,7C,1
8,18,22A                          (2873)
430 DATA 18,18,18,10,22,66,66,66,6
6,66,278                          (2814)
440 DATA 5C,38,22,66,66,66,66,54,2
8,10,2DA                          (2864)
450 DATA 22,66,66,66,6E,76,6E,44,2
2,66,372                          (287F)
460 DATA 54,28,14,2A,66,44,22,66,6
6,5C,2AE                          (2875)
470 DATA 38,18,18,10,3E,7E,04,08,1
0,20,170                          (28BD)
480 DATA 7E,7C,3C,30,30,30,30,30,3
C,00,262                          (28EE)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                          (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                          (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                          (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                          (28F1)
530 DATA 00,00,1C,26,66,A7,1A,00,2
0,50,1D9                          (28D3)
540 DATA 50,50,56,69,B0,00,00,00,3
C,66,2B1                          (28F2)
550 DATA 60,E7,3C,00,0C,0C,7C,CC,C
C,CD,47C                          (283B)
560 DATA 76,00,00,00,3C,66,7C,E1,3
E,00,2B3                          (28C9)
570 DATA 30,30,48,50,70,A7,78,20,0
0,3C,2E3                          (28EA)
580 DATA 66,66,7E,8F,06,06,60,60,6
8,7C,389                          (28F7)
590 DATA 74,67,66,00,30,00,30,30,7
8,CF,318                          (28B4)
600 DATA 00,00,10,00,10,34,5B,90,3
0,10,17F                          (28DB)
610 DATA 60,60,66,6C,78,ED,66,00,2
0,50,3CD                          (28AD)
620 DATA 50,20,60,A7,18,00,00,00,6
C,FE,2F9                          (2855)
630 DATA D6,D7,C6,00,00,00,3C,66,6
6,E7,462                          (28E7)
640 DATA 66,00,00,00,3C,66,E7,66,3
C,00,291                          (28A2)
650 DATA 00,00,5C,66,66,FD,60,60,0
0,00,2E5                          (28E9)
660 DATA 74,CC,CC,7F,0C,0C,00,00,5
C,76,375                          (288D)
670 DATA E2,E1,60,00,00,00,3E,60,3
C,86,383                          (28D1)
680 DATA 7C,00,30,30,7C,70,B0,37,1
C,00,2CB                          (28C9)
690 DATA 00,00,66,66,E6,67,3E,00,0
0,00,257                          (2889)
700 DATA 66,66,E6,3D,18,00,00,00,C
6,D6,3A3                          (2805)
710 DATA D6,FF,6C,00,00,00,C6,6C,3
8,6D,418                          (2854)
720 DATA C6,00,00,00,66,66,66,BF,0
6,7C,339                          (28B2)
730 DATA 00,00,7E,0C,18,B1,7E,00,0
E,18,1F7                          (289C)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                          (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                          (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                          (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                             (2511)
780 dat=0:sz=0:dz=10              (1E65)
790 FOR adr=&A000 TO &A300        (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35FB)
810 sz=sz+byte                    (183D)
820 POKE adr,byte                 (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30EA)
860 dz=dz+10:sz=0:dat=0           (24DC)
870 NEXT                          (0692)
880 SAVE"G3H.BIN",B,&A000,&300    (1CC8)
```

```
1 '****************************** (2506)
2 '*       DATALADER G3I.LAD      * (250A)
3 '*        ERZEUGT G3I.BIN       * (2574)
4 '*    WIRD BENOETIGT FUER       * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464         ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                           (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                           (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                           (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                           (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                           (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                           (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                           (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                           (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                           (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                          (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                          (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                          (285D)
130 DATA 06,0C,18,30,60,C0,80,00,3
C,66,29C                          (288D)
140 DATA 42,42,42,66,3C,00,18,18,1
8,18,1C8                          (2872)
150 DATA 18,18,18,00,3C,66,06,3C,6
0,60,1EC                          (281B)
160 DATA 7E,00,3C,66,06,1C,06,66,3
C,00,1EA                          (28B9)
170 DATA 1C,3C,6C,CC,FE,0C,1E,00,7
E,60,396                          (2833)
180 DATA 60,7C,06,66,3C,00,3C,66,6
0,7C,302                          (284F)
190 DATA 66,66,3C,00,7E,06,0C,0C,1
8,18,1D4                          (28D5)
200 DATA 30,00,3C,66,66,3C,66,66,3
C,00,27C                          (2854)
210 DATA 3C,66,66,3E,06,66,3C,00,0
0,00,1EE                          (28D1)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                          (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                          (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                          (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                          (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                          (2813)
270 DATA DE,C0,7C,00,18,18,3C,24,6
6,7E,38E                          (2895)
280 DATA 66,00,7C,66,62,7C,62,66,7
C,00,36A                          (2841)
290 DATA 3C,66,40,40,40,66,3C,00,7
8,4C,2C8                          (2855)
300 DATA 44,44,44,4C,78,00,7E,60,6
0,78,346                          (28E1)
310 DATA 60,60,7E,00,7E,60,60,7C,6
0,60,3B8                          (2821)
```

CPC Graphic Designer

```
320 DATA 60,00,3C,66,40,4E,46,66,3
C,00,278                                (2815)
330 DATA 66,66,66,7E,66,66,66,00,1
8,18,312                                (28C7)
340 DATA 18,18,18,18,18,00,06,06,0
6,06,090                                (28FA)
350 DATA 06,CE,7C,00,63,66,6C,78,6
C,66,3CF                                (284D)
360 DATA 63,00,60,60,60,60,60,60,7
E,00,321                                (284C)
370 DATA C6,EE,FE,D6,C6,C6,C6,00,4
6,66,686                                (28DB)
380 DATA 76,7E,6E,66,62,00,3C,66,4
2,42,350                                (2845)
390 DATA 42,66,3C,00,7C,62,62,7C,6
0,60,360                                (28BC)
400 DATA 60,00,3C,66,42,42,4A,66,3
E,00,274                                (284A)
410 DATA 7C,62,62,7C,64,66,62,00,3
C,66,38A                                (28E3)
420 DATA 60,3C,06,66,3C,00,7E,18,1
8,18,20A                                (2815)
430 DATA 18,18,18,00,66,66,66,66,6
6,7E,2C4                                (2819)
440 DATA 3C,00,66,66,66,24,3C,18,1
8,00,1FE                                (288B)
450 DATA C6,C6,D6,FE,EE,C6,82,00,C
3,66,6BF                                (28CA)
460 DATA 3C,18,3C,66,C3,00,66,66,7
E,3C,33F                                (28E2)
470 DATA 18,18,18,00,FE,FE,0C,18,3
0,7E,316                                (285E)
480 DATA FE,00,3C,30,30,30,30,30,3
C,00,266                                (28BA)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 7C,EE,EE,FE,EE,EE,EE,00,F
C,EE,80A                                (2874)
540 DATA EE,FC,EE,EE,FC,00,7C,EE,E
0,E0,7EC                                (28A9)
550 DATA E0,EE,7C,00,FC,EE,EE,EE,E
E,EE,7EC                                (286F)
560 DATA FC,00,FE,E0,E0,FC,E0,E0,F
E,00,774                                (2852)
570 DATA FE,E0,E0,FC,E0,E0,E0,00,7
C,EE,7C4                                (28EB)
580 DATA E0,E0,EE,EE,7E,00,EE,EE,E
E,FE,7E2                                (28B8)
590 DATA EE,EE,EE,00,38,38,38,38,3
8,38,41A                                (28AD)
600 DATA 38,00,0E,0E,0E,0E,EE,EE,7
C,00,2C8                                (2878)
610 DATA EE,EE,FC,F8,FC,EE,EE,00,E
0,E0,868                                (2885)
620 DATA E0,E0,E0,FE,FE,00,C6,EE,F
E,FE,84C                                (2826)
630 DATA FE,EE,EE,00,CE,EE,FE,FE,F
E,EE,87E                                (2855)
640 DATA E6,00,7C,EE,EE,EE,EE,EE,7
C,00,684                                (2843)
650 DATA FC,EE,EE,FC,E0,E0,E0,00,7
C,EE,7DE                                (2837)
660 DATA EE,EE,FE,EE,7F,00,FC,EE,E
E,FC,81B                                (2862)
670 DATA EE,EE,EE,00,7C,EE,E0,7C,0
E,EE,68C                                (287D)
680 DATA 7C,00,FE,FE,38,38,38,38,3
8,00,390                                (285B)
690 DATA EE,EE,EE,EE,EE,EE,7C,00,E
E,EE,7EC                                (284A)
700 DATA EE,EE,EE,7C,38,00,EE,EE,E
E,FE,746                                (288E)
710 DATA FE,EE,C6,00,EE,EE,7C,38,7
C,EE,6AC                                (2813)
720 DATA EE,00,EE,EE,EE,EE,7C,38,38,3
8,00,4DC                                (288E)
730 DATA FE,FE,1C,38,70,FE,FE,00,0
E,18,4E2                                (286F)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                                   (2511)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3I.BIN",B,&A000,&300          (1C58)
```

```
1 '*******************************  (2506)
2 '*       DATALADER G3J.LAD      *  (251B)
3 '*        ERZEUGT G3J.BIN       *  (25A4)
4 '*    WIRD BENOETIGT FUER       *  (25CD)
5 '*   CPC GRAPHICS DESIGNER      *  (25E5)
6 '*          CPC 464          ME*  (2593)
7 '*******************************  (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,0
0,FC,2F6                                (2894)
140 DATA CC,CC,CC,CC,FC,00,00,30,F
0,30,57C                                (2845)
150 DATA 30,30,FC,00,00,FC,0C,0C,F
C,C0,42C                                (2879)
160 DATA FC,00,00,FC,CC,3C,0C,CC,F
C,00,4D4                                (28A3)
170 DATA 00,C0,CC,CC,FC,0C,0C,00,0
0,FC,468                                (2887)
180 DATA C0,FC,0C,CC,FC,00,00,FC,C
0,FC,648                                (2894)
190 DATA CC,CC,FC,00,00,FC,0C,0C,3
0,30,408                                (2892)
200 DATA 30,00,00,FC,CC,FC,CC,CC,F
C,00,588                                (2870)
210 DATA 00,FC,CC,CC,FC,0C,FC,00,0
0,00,498                                (2857)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,00,FC,CC,CC,F
C,CC,676                                (28DB)
280 DATA CC,00,00,FC,CC,F0,CC,CC,F
C,00,618                                (2899)
290 DATA 00,FC,CC,C0,C0,CC,FC,00,0
0,F0,600                                (28C4)
300 DATA CC,CC,CC,CC,F0,00,00,FC,C
0,F0,6CC                                (2806)
310 DATA C0,C0,FC,00,00,FC,C0,F0,C
0,C0,6A8                                (28DD)
320 DATA C0,00,00,FC,C0,C0,CC,CC,F
C,00,5D0                                (2823)
330 DATA 00,CC,CC,FC,CC,CC,CC,00,0
0,30,528                                (287F)
340 DATA 30,30,30,30,30,00,00,0C,0
C,0C,114                                (2873)
350 DATA CC,CC,78,00,00,CC,CC,F0,C
C,CC,630                                (283B)
360 DATA CC,00,00,C0,C0,C0,C0,C0,F
C,00,588                                (2879)
370 DATA 00,CC,FC,CC,CC,CC,CC,00,0
0,F0,5E8                                (282E)
380 DATA CC,CC,CC,CC,CC,00,00,FC,C
C,CC,690                                (2862)
390 DATA CC,CC,FC,00,00,FC,CC,CC,F
C,C0,6E4                                (28BA)
400 DATA C0,00,00,FC,CC,CC,CC,FC,F
F,00,61B                                (289F)
410 DATA 00,FC,CC,CC,F0,CC,CC,00,0
0,FC,618                                (2866)
420 DATA C0,FC,0C,0C,FC,00,00,FC,3
0,30,42C                                (28C4)
430 DATA 30,30,30,00,00,CC,CC,CC,C
C,CC,48C                                (288B)
440 DATA FC,00,00,CC,CC,CC,CC,30,3
0,00,48C                                (28BC)
450 DATA 00,CC,CC,CC,CC,FC,CC,00,0
0,CC,5C4                                (28EB)
460 DATA CC,30,30,CC,CC,00,00,CC,C
C,FC,558                                (289C)
470 DATA 30,30,30,00,00,FC,0C,30,3
0,C0,2B8                                (2852)
480 DATA FC,00,3C,30,30,30,30,30,3
C,00,264                                (28BE)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                                (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                                (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                                (28F1)
530 DATA 00,00,FC,0C,FC,CC,FC,00,0
0,C0,48C                                (289B)
540 DATA C0,FC,CC,CC,FC,00,00,00,F
C,C0,60C                                (28BD)
550 DATA C0,C0,FC,00,00,0C,0C,FC,C
C,CC,528                                (288C)
560 DATA FC,00,00,00,FC,CC,FC,C0,F
C,00,57C                                (2883)
570 DATA 00,3C,C0,F0,C0,C0,C0,00,0
0,00,42C                                (285B)
580 DATA FC,CC,CC,FC,0C,FC,00,C0,C
0,FC,714                                (284A)
590 DATA CC,CC,CC,00,00,30,00,30,3
0,30,324                                (2834)
600 DATA 30,00,00,0C,00,0C,0C,CC,C
C,FC,2E8                                (28B2)
610 DATA 00,C0,CC,CC,F0,CC,CC,00,0
0,C0,5A0                                (28A6)
620 DATA C0,C0,C0,C0,30,00,00,00,C
C,FC,4F8                                (2856)
630 DATA CC,CC,CC,00,00,00,F0,CC,C
C,CC,5B8                                (2822)
640 DATA CC,00,00,00,FC,CC,CC,CC,F
C,00,528                                (2828)
650 DATA 00,00,FC,CC,CC,FC,C0,C0,0
0,00,510                                (285A)
660 DATA FC,CC,CC,FC,0C,0C,00,00,3
C,C0,4A4                                (28C7)
```

```
670 DATA C0,C0,C0,00,00,00,FC,C0,F
C,0C,504                                <28E7>
680 DATA FC,00,00,30,FC,30,30,30,3
C,00,2F4                                <28F9>
690 DATA 00,00,CC,CC,CC,CC,FC,00,0
0,00,42C                                <2854>
700 DATA CC,CC,CC,30,30,00,00,00,C
C,CC,45C                                <28CC>
710 DATA CC,FC,CC,00,00,00,CC,CC,3
0,CC,528                                <285D>
720 DATA CC,00,00,00,CC,CC,CC,FC,0
C,FC,534                                <2835>
730 DATA 00,00,FC,0C,30,C0,FC,00,0
E,18,31A                                <28D1>
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                                <282A>
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                                <281A>
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                                <28B5>
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                                   <2511>
780 dat=0:sz=0:dz=10                    <1E65>
790 FOR adr=&A000 TO &A300              <1400>
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              <35FB>
810 sz=sz+byte                          <183D>
820 POKE adr,byte                       <142A>
830 IF dat<10 AND adr<&A300 THEN 8
70                                      <1EEA>
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  <2BC0>
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        <30EA>
860 dz=dz+10:sz=0:dat=0                 <24DC>
870 NEXT                                <0692>
880 SAVE"G3J.BIN",B,&A000,&300          <1CA8>
```

```
1 '******************************* <2506>
2 '*       DATALADER G3K.LAD     * <252B>
3 '*        ERZEUGT G3K.BIN      * <2544>
4 '*    WIRD BENOETIGT FUER      * <25CD>
5 '*   CPC GRAPHICS DESIGNER     * <25E5>
6 '*          CPC 464         ME* <2593>
7 '******************************* <2512>
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 <28EA>
20 DATA 18,18,18,00,18,00,36,36,36
,00,102                                 <287B>
30 DATA 00,00,00,00,36,36,7F,36,7F
,36,1D6                                 <28CD>
40 DATA 36,00,18,7C,1A,3C,58,3E,18
,00,1CE                                 <281A>
50 DATA 00,63,33,18,0C,66,63,00,1C
,36,1D5                                 <28DA>
60 DATA 1C,6E,3B,33,6E,00,18,18,0C
,00,1A2                                 <28ED>
70 DATA 00,00,00,00,30,18,0C,0C,0C
,18,084                                 <2889>
80 DATA 30,00,0C,18,30,30,30,18,0C
,00,108                                 <28B0>
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 <28D8>
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                <28B8>
110 DATA 00,18,18,0C,00,00,00,7E,0
0,00,0BA                                <282A>
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                <285D>
130 DATA 60,30,18,0C,06,03,01,00,3
E,63,15F                                <28D0>
140 DATA 73,6B,67,63,3E,00,18,1C,1
8,18,24A                                <28AA>
150 DATA 18,18,7E,00,3C,66,60,3C,0
6,66,258                                <28CF>
160 DATA 7E,00,3C,66,60,38,60,66,3
C,00,2BA                                <28C9>
170 DATA 38,3C,36,33,7F,30,78,00,7
E,46,2C8                                <2802>
180 DATA 06,3E,60,66,3C,00,3C,66,0
6,3E,22C                                <28E6>
190 DATA 66,66,3C,00,7E,66,60,30,1
8,18,2AC                                <2884>
200 DATA 18,00,3C,66,66,3C,66,66,3
C,00,264                                <28FF>
210 DATA 3C,66,66,7C,60,66,3C,00,0
0,00,286                                <28B9>
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                <2810>
230 DATA 00,18,18,0C,30,18,0C,06,0
C,18,0BA                                <285A>
240 DATA 30,00,00,00,7E,00,00,7E,0
0,00,12C                                <281A>
250 DATA 06,0C,18,30,18,0C,06,00,3
C,66,126                                <28C5>
260 DATA 66,30,18,00,18,00,3E,63,7
B,7B,25D                                <28C4>
270 DATA 7B,03,3E,00,18,3C,66,66,7
E,66,2C0                                <28A3>
280 DATA 66,00,3F,66,66,3E,66,66,3
F,00,2BA                                <2891>
290 DATA 3C,66,03,03,03,66,3C,00,1
F,36,1A2                                <2879>
300 DATA 66,66,66,36,1F,00,7F,46,1
6,1E,280                                <28AC>
310 DATA 16,46,7F,00,7F,46,16,1E,1
6,06,1F0                                <2857>
320 DATA 0F,00,3C,66,03,03,73,66,7
C,00,20C                                <288D>
330 DATA 66,66,66,7E,66,66,66,00,7
E,18,378                                <284C>
340 DATA 18,18,18,18,7E,00,78,30,3
0,30,1E6                                <2846>
350 DATA 33,33,1E,00,67,66,36,1E,3
6,66,241                                <2834>
360 DATA 67,00,0F,06,06,06,46,66,7
F,00,1B3                                <28D9>
370 DATA 63,77,7F,7F,6B,63,63,00,6
3,67,3D3                                <2888>
380 DATA 6F,7B,73,63,63,00,1C,36,6
3,63,33B                                <28DB>
390 DATA 63,36,1C,00,3F,66,66,3E,0
6,06,20A                                <281D>
400 DATA 0F,00,1C,36,63,63,5B,33,6
E,00,223                                <2877>
410 DATA 3F,66,66,3E,36,66,67,00,3
C,66,2EE                                <28D3>
420 DATA 06,3C,60,66,3C,00,7E,5A,1
8,18,24C                                <28D1>
430 DATA 18,18,3C,00,66,66,66,66,6
6,66,2D0                                <281E>
440 DATA 3C,00,66,66,66,66,66,3C,1
8,00,28E                                <281C>
450 DATA 63,63,63,6B,7F,77,63,00,6
3,36,386                                <28C2>
460 DATA 1C,1C,36,63,63,00,66,66,6
6,3C,2A2                                <2832>
470 DATA 18,18,3C,00,7F,63,31,18,4
C,66,249                                <289B>
480 DATA 7F,00,3C,0C,0C,0C,0C,0C,3
C,00,133                                <28E8>
490 DATA 03,06,0C,18,30,60,40,00,3
C,30,169                                <281E>
500 DATA 30,30,30,30,3C,00,18,3C,7
E,18,1E6                                <28C6>
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                                <2844>
520 DATA 00,FF,0C,18,30,00,00,00,0
0,00,153                                <2855>
530 DATA 00,00,1E,30,3E,33,6E,00,0
7,06,13A                                <2839>
540 DATA 3E,66,66,66,3B,00,00,00,3
C,66,24D                                <28B8>
550 DATA 06,66,3C,00,38,30,3E,33,3
3,33,1E7                                <289E>
560 DATA 6E,00,00,00,3C,66,7E,06,3
C,00,1D0                                <2895>
570 DATA 38,6C,0C,1E,0C,0C,1E,00,0
0,00,104                                <2890>
580 DATA 7C,66,66,7C,60,3E,07,06,3
6,6E,313                                <286A>
590 DATA 66,66,67,00,18,00,1C,18,1
8,18,1AF                                <28B5>
600 DATA 3C,00,60,00,70,60,60,66,6
6,3C,2D4                                <2880>
610 DATA 07,06,66,36,1E,36,67,00,1
C,18,198                                <280E>
620 DATA 18,18,18,18,3C,00,00,00,3
6,7F,151                                <2888>
630 DATA 6B,6B,63,00,00,00,3B,66,6
6,66,2A6                                <285C>
640 DATA 66,00,00,00,3C,66,66,66,3
C,00,210                                <289E>
650 DATA 00,00,3B,66,66,3E,06,0F,0
0,00,15A                                <28E7>
660 DATA 6E,33,33,3E,30,78,00,00,3
B,6E,263                                <28B6>
670 DATA 06,06,0F,00,00,00,3C,06,3
C,60,0F9                                <2860>
680 DATA 3E,00,0C,0C,3E,0C,0C,6C,3
8,00,150                                <28E5>
690 DATA 00,00,66,66,66,66,7C,00,0
0,00,214                                <28E6>
700 DATA 66,66,66,3C,18,00,00,00,6
3,6B,254                                <28F0>
710 DATA 6B,7F,36,00,00,00,63,36,1
C,36,20B                                <28D9>
720 DATA 63,00,00,00,66,66,66,7C,6
0,3E,2AF                                <281B>
730 DATA 00,00,7E,32,18,4C,7E,00,7
0,18,21A                                <285B>
740 DATA 18,0E,18,18,70,00,18,18,1
8,18,126                                <284F>
750 DATA 18,18,18,00,0E,18,18,70,1
8,18,126                                <2820>
760 DATA 0E,00,6E,3B,00,00,00,00,0
0,00,0B7                                <283F>
770 DATA 33,CC,33,CC,33,CC,33,CC,0
0,3FC                                   <255F>
780 dat=0:sz=0:dz=10                    <1E65>
790 FOR adr=&A000 TO &A300              <1400>
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              <35FB>
810 sz=sz+byte                          <183D>
820 POKE adr,byte                       <142A>
830 IF dat<10 AND adr<&A300 THEN 8
70                                      <1EEA>
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  <2BC0>
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        <30EA>
860 dz=dz+10:sz=0:dat=0                 <24DC>
870 NEXT                                <0692>
880 SAVE"G3K.BIN",B,&A000,&300          <1C78>
```

```
1 '******************************* <2506>
2 '*       DATALADER G3N.LAD     * <255B>
3 '*        ERZEUGT G3N.BIN      * <2525>
4 '*    WIRD BENOETIGT FUER      * <25CD>
5 '*   CPC GRAPHICS DESIGNER     * <25E5>
6 '*          CPC 464         ME* <2593>
7 '******************************* <2512>
10 DATA 00,00,00,00,00,00,00,00,00
,10,010                                 <28A0>
20 DATA 38,38,10,10,00,10,00,36,36
,00,10C                                 <285D>
30 DATA 00,00,00,00,00,28,44,FE,44
,FE,2AC                                 <282E>
40 DATA 44,28,00,28,7E,A8,FE,2A,FC
,28,406                                 <2882>
50 DATA 00,00,46,CC,18,30,66,C4,00
,30,2B4                                 <2832>
60 DATA 68,30,60,D4,C8,74,00,0C,18
,00,32C                                 <284F>
70 DATA 00,00,00,00,00,30,60,60,60
,60,1B0                                 <28FC>
80 DATA 60,30,00,18,0C,0C,0C,0C,0C
,18,0FC                                 <2863>
90 DATA 00,10,54,38,FE,38,54,10,00
,00,236                                 <2818>
100 DATA 08,10,7C,10,20,00,00,00,0
0,00,0C4                                <28B9>
110 DATA 00,00,0C,18,00,00,00,3E,7
C,00,0DE                                <28DB>
```

CPC Graphic Designer

```
120 DATA 00,00,00,00,00,00,00,00,1
C,1C,038                                (287E)
130 DATA 00,00,06,0C,18,30,60,C0,0
0,7C,1F6                                (2877)
140 DATA EE,C6,C6,C6,EE,7C,00,18,3
8,38,532                                (28A4)
150 DATA 18,18,3E,7C,00,7C,CE,9C,3
8,70,378                                (28CD)
160 DATA E6,FE,00,7E,CC,98,3C,0E,C
E,7C,55A                                (28B0)
170 DATA 00,60,C8,D8,7E,18,18,10,0
0,FC,3BA                                (28E0)
180 DATA 66,60,7C,0E,CE,7C,00,3C,6
0,DC,412                                (284A)
190 DATA F6,C2,E6,7C,00,7E,E6,0E,1
C,38,4E0                                (2829)
200 DATA 30,30,00,7C,EE,C6,7C,C6,E
E,7C,53C                                (28F0)
210 DATA 00,7C,CE,86,DE,76,0C,78,0
0,00,3A8                                (2881)
220 DATA 00,30,30,00,30,30,00,00,0
0,18,0D8                                (28D4)
230 DATA 18,00,18,30,00,10,30,60,C
0,60,220                                (28C4)
240 DATA 30,10,00,00,3C,78,00,3C,7
8,00,1A8                                (28D9)
250 DATA 00,10,18,0C,06,0C,18,10,0
0,7C,0EA                                (2818)
260 DATA C6,46,1C,30,00,30,00,7C,C
6,9A,364                                (2828)
270 DATA AA,DC,C0,7C,00,3C,66,C6,F
E,C6,5EE                                (28E8)
280 DATA E6,66,00,D8,EC,CC,FC,C6,E
6,DC,760                                (2893)
290 DATA 00,38,6C,C6,C0,C2,EE,7C,0
0,DC,532                                (28F6)
300 DATA E6,C6,C6,C6,CC,F8,00,DC,E
6,60,71E                                (28A5)
310 DATA 7C,60,E6,DC,00,EE,72,60,7
C,60,53A                                (282A)
320 DATA E0,C0,00,3C,66,C0,CE,C4,E
C,78,5F8                                (28FB)
330 DATA 00,CC,C6,C6,FE,C6,C6,66,0
0,7E,5C6                                (28A6)
340 DATA 98,30,30,30,1A,FC,00,3E,0
C,0C,294                                (2872)
350 DATA E6,66,C6,7C,00,C0,66,6C,7
8,78,510                                (285B)
360 DATA EC,C6,00,C0,E0,60,60,60,F
C,C2,630                                (28E5)
370 DATA 00,66,FE,D6,D6,C6,E6,66,0
0,CC,5EE                                (2851)
380 DATA E6,E6,D6,CE,CE,66,00,38,6
C,C6,60E                                (2894)
390 DATA C6,C6,EE,7C,00,DC,E6,C6,C
E,F8,744                                (2829)
400 DATA C0,C0,00,38,6C,C6,D6,CC,E
C,76,5EE                                (28BF)
410 DATA 00,DC,E6,C6,EC,D8,CC,C6,0
0,3C,61A                                (287F)
420 DATA 66,62,3C,46,C6,7C,00,FE,B
A,18,45C                                (287C)
430 DATA 18,18,38,30,00,E6,66,C6,C
6,C6,436                                (28C0)
440 DATA C6,7C,00,CE,CC,C6,C6,C6,6
C,38,5D2                                (28B9)
450 DATA 00,CC,CE,C6,D6,D6,FE,6C,0
0,EE,664                                (2826)
460 DATA C6,6C,38,6C,C6,EE,00,CC,C
6,66,582                                (2884)
470 DATA 3C,18,30,60,00,76,8C,18,3
0,60,28E                                (286C)
480 DATA C2,BC,00,38,60,60,60,60,6
0,38,3CE                                (28C7)
490 DATA 00,00,C0,60,30,18,0C,06,0
0,38,1B2                                (28DE)
500 DATA 0C,0C,0C,0C,0C,38,00,10,3
8,7C,138                                (2849)
510 DATA D6,10,10,10,00,00,00,00,0
0,00,106                                (2836)
520 DATA 7E,FC,00,30,18,00,00,00,0
0,00,1C2                                (283C)
530 DATA 00,00,78,8C,CC,CC,76,00,4
0,C0,412                                (2863)
540 DATA DC,E2,C6,C6,FC,00,00,00,7
C,E6,5A8                                (28C3)
550 DATA C0,C6,7C,00,04,06,76,8E,C
E,C6,4A4                                (28A2)
560 DATA 7E,00,00,00,7C,C6,FC,C0,7
C,00,3F8                                (2874)
570 DATA 00,3C,66,60,F0,60,60,C0,0
0,00,372                                (2861)
580 DATA 78,C6,CE,76,06,3C,40,C0,D
C,E2,582                                (2836)
590 DATA C6,C6,C4,00,00,18,00,18,1
8,18,2B0                                (28BD)
600 DATA 1C,08,00,18,00,18,18,18,9
8,70,18C                                (28F8)
610 DATA 00,40,C8,D0,E0,D8,CC,00,0
0,20,47C                                (28E0)
620 DATA 30,30,30,30,38,10,00,00,A
8,FC,2AC                                (28E4)
630 DATA D2,D6,D4,00,00,00,BC,EE,C
6,C6,5B2                                (28D4)
640 DATA 44,00,00,00,7C,8E,C6,E2,7
C,00,372                                (285C)
650 DATA 00,00,BC,C6,C2,E6,DC,40,0
0,00,446                                (287D)
660 DATA 74,CC,8C,DC,6C,06,00,00,5
C,E6,45C                                (2886)
670 DATA C2,C0,40,00,00,00,3C,62,3
C,86,322                                (28C0)
680 DATA 7C,00,10,30,7C,30,30,34,1
8,00,1E4                                (28FA)
690 DATA 00,00,42,C6,C6,EE,7A,00,0
0,00,336                                (2864)
700 DATA 86,C6,CC,6C,30,00,00,00,5
2,D6,3DC                                (28CC)
710 DATA D6,FE,6C,00,00,00,8C,58,3
0,68,3BC                                (284C)
720 DATA C4,00,00,00,44,C6,EE,7A,0
6,7C,3B8                                (2818)
730 DATA 00,00,7A,8E,30,C6,BE,00,0
0,38,2F4                                (2834)
740 DATA 60,60,E0,60,60,38,00,08,1
8,18,2D0                                (28FD)
750 DATA 18,18,18,10,00,38,0C,0C,0
E,0C,0C2                                (2813)
760 DATA 0C,38,00,36,6C,00,00,00,0
0,00,0E6                                (2812)
770 DATA 00,3C,66,F8,60,F8,60,FE,0
0,450                                   (2531)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3N.BIN",B,&A000,&300          (1CE9)
```

```
1 '*****************************  (2506)
2 '*      DATALADER G3L.LAD      *  (25FA)
3 '*       ERZEUGT G3L.BIN       *  (25E3)
4 '*    WIRD BENOETIGT FUER      *  (25CD)
5 '*   CPC GRAPHICS DESIGNER     *  (25E5)
6 '*          CPC 464          ME*  (2593)
7 '*****************************  (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                                 (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                                 (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                                 (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                                 (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                                 (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                                 (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                                 (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                                 (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                                 (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                                (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                                (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                                (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
4,44,2B2                                (2822)
140 DATA 44,00,44,44,74,00,04,04,0
4,00,14C                                (2866)
150 DATA 04,04,04,00,74,04,04,38,4
0,40,140                                (287B)
160 DATA 70,00,74,04,04,38,04,04,7
4,00,1A0                                (28B3)
170 DATA 04,44,44,38,04,04,04,00,7
0,40,180                                (280A)
180 DATA 40,38,04,04,74,00,70,40,4
0,38,21C                                (28FA)
190 DATA 44,44,74,00,74,44,44,00,0
4,04,200                                (28F9)
200 DATA 04,00,74,44,44,38,44,44,7
4,00,234                                (2870)
210 DATA 74,44,44,38,04,04,74,00,0
0,00,1B0                                (283F)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                                (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                                (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                                (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                                (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                                (2813)
270 DATA DE,C0,7C,00,7E,81,9D,9D,9
D,81,571                                (289E)
280 DATA 9D,E7,7E,81,9D,9D,82,9D,8
1,7E,5DB                                (2891)
290 DATA 7F,C1,9E,9F,9F,9E,C1,7F,F
C,82,678                                (284E)
300 DATA 9D,9D,9D,9D,82,FC,FF,81,9
F,82,693                                (28F3)
310 DATA 9E,9E,81,FF,FF,81,9E,82,9
E,90,68A                                (2804)
320 DATA 90,E0,3E,C3,9F,9F,91,9D,C
3,7E,61E                                (2806)
330 DATA FE,9A,9A,82,9A,9A,9A,FE,7
E,42,640                                (2878)
340 DATA 66,24,24,66,42,7E,07,05,0
5,FD,2E2                                (2863)
350 DATA 9D,9D,81,7F,F7,99,93,87,8
7,93,5FE                                (2879)
360 DATA 99,F7,F0,90,90,90,90,9F,8
1,FF,6DF                                (2876)
370 DATA F3,9D,89,81,95,9D,95,F7,F
3,9D,6E8                                (28A7)
380 DATA 8D,85,91,99,95,F3,3C,42,9
D,9D,57C                                (282F)
390 DATA 9D,9D,42,3C,FE,83,9D,9D,8
3,9E,594                                (28E2)
400 DATA 90,F0,3C,42,9D,9D,9D,99,4
1,3E,4ED                                (2882)
410 DATA FE,83,9D,9D,83,9B,9D,F7,7
E,C2,6AD                                (2801)
420 DATA 9E,C2,79,99,C3,7E,FF,81,E
7,24,63E                                (289B)
430 DATA 24,24,24,3C,E3,9D,9D,9D,9
D,9D,49C                                (283A)
440 DATA C3,7E,E3,9D,9D,9D,9D,CB,6
6,3C,605                                (289C)
450 DATA E3,9D,9D,9D,95,81,9D,E3,E
3,9D,6D0                                (284D)
460 DATA 89,42,42,89,9D,E3,E3,9D,9
D,89,5BC                                (2886)
```

```
470 DATA 42,24,24,3C,FF,81,71,22,4
4,8E,3AB                          (28FC)
480 DATA 81,FF,3C,30,30,30,30,30,3
C,00,2E8                          (2891)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                          (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                          (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                          (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                          (28F1)
530 DATA 74,44,44,38,44,44,04,00,7
0,44,274                          (28AD)
540 DATA 44,38,44,44,70,00,78,40,4
0,00,26C                          (287F)
550 DATA 40,40,78,00,5C,44,44,00,4
4,44,264                          (2835)
560 DATA 5C,00,78,40,40,30,40,40,7
8,00,27C                          (2809)
570 DATA 78,40,40,30,40,40,00,00,5
C,40,244                          (28B0)
580 DATA 40,00,44,44,5C,00,48,48,4
8,30,22C                          (2857)
590 DATA 48,48,48,00,40,40,40,00,4
0,40,218                          (285F)
600 DATA 40,00,04,04,04,00,44,44,7
4,00,148                          (28BE)
610 DATA 40,48,48,30,48,48,40,00,4
0,40,250                          (2839)
620 DATA 40,00,40,40,5C,00,74,54,5
4,00,238                          (2869)
630 DATA 44,44,04,00,78,44,44,00,4
4,44,214                          (2844)
640 DATA 04,00,74,44,44,00,44,44,7
4,00,1FC                          (2833)
650 DATA 5C,44,44,38,40,40,40,00,7
4,44,294                          (2801)
660 DATA 44,00,44,5E,74,00,5C,44,4
4,38,276                          (28CE)
670 DATA 44,44,40,00,5C,40,40,38,0
4,04,1E4                          (28B0)
680 DATA 74,00,7C,00,10,10,00,10,1
0,00,130                          (28EE)
690 DATA 04,44,44,00,44,44,74,00,4
8,48,218                          (28EE)
700 DATA 48,00,48,50,40,00,04,44,4
4,00,1AC                          (281F)
710 DATA 54,54,74,00,00,44,44,38,4
4,44,264                          (28A5)
720 DATA 00,00,04,44,44,38,10,10,0
0,00,0E4                          (28FB)
730 DATA 74,04,04,10,20,00,70,00,0
E,18,142                          (2869)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                          (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                          (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                          (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                             (2511)
780 dat=0:sz=0:dz=10              (1E65)
790 FOR adr=&A000 TO &A300        (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35FB)
810 sz=sz+byte                    (183D)
820 POKE adr,byte                 (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30EA)
860 dz=dz+10:sz=0:dat=0           (24DC)
870 NEXT                          (0692)
880 SAVE"G3L.BIN",B,&A000,&300    (1C88)
```

```
1 '******************************* (2506)
2 '*       DATALADER G3M.LAD       * (25CA)
3 '*        ERZEUGT G3M.BIN        * (2505)
4 '*    WIRD BENOETIGT FUER        * (25CD)
5 '*   CPC GRAPHICS DESIGNER       * (25E5)
6 '*          CPC 464          ME* (2573)
7 '******************************* (2512)
10 DATA 00,00,00,00,00,00,00,00,18
,18,030                           (28EA)
20 DATA 18,18,18,00,18,00,6C,6C,6C
,00,1A4                           (282A)
30 DATA 00,00,00,00,6C,6C,FE,6C,FE
,6C,3AC                           (2889)
40 DATA 6C,00,18,3E,58,3C,1A,7C,18
,00,204                           (28E4)
50 DATA 00,C6,CC,18,30,66,C6,00,38
,6C,3AA                           (28D6)
60 DATA 38,76,DC,CC,76,00,18,18,30
,00,32C                           (2843)
70 DATA 00,00,00,00,0C,18,30,30,30
,18,0CC                           (28B3)
80 DATA 0C,00,30,18,0C,0C,0C,18,30
,00,0C0                           (28B7)
90 DATA 00,66,3C,FF,3C,66,00,00,00
,18,25B                           (28D8)
100 DATA 18,7E,18,18,00,00,00,00,0
0,00,0C6                          (28B8)
110 DATA 00,18,18,30,00,00,00,7E,0
0,00,0DE                          (2877)
120 DATA 00,00,00,00,00,00,00,18,1
8,00,030                          (285D)
130 DATA 06,0C,18,30,60,C0,80,00,7
C,C6,33C                          (2844)
140 DATA CE,D6,E6,C6,7C,00,18,38,1
8,18,44C                          (2870)
150 DATA 18,18,7E,00,3C,66,06,3C,6
0,66,258                          (28C6)
160 DATA 7E,00,3C,66,06,1C,06,66,3
C,00,1EA                          (28B9)
170 DATA 1C,3C,6C,CC,FE,0C,1E,00,7
E,62,398                          (28B9)
180 DATA 60,7C,06,66,3C,00,3C,66,6
0,7C,302                          (284F)
190 DATA 66,66,3C,00,7E,66,06,0C,1
8,18,22E                          (2822)
200 DATA 18,00,3C,66,66,3C,66,66,3
C,00,264                          (28FF)
210 DATA 3C,66,66,3E,06,66,3C,00,0
0,00,1EE                          (28D1)
220 DATA 18,18,00,18,18,00,00,00,1
8,18,090                          (2810)
230 DATA 00,18,18,30,0C,18,30,60,3
0,18,15C                          (287A)
240 DATA 0C,00,00,00,7E,00,00,7E,0
0,00,108                          (289F)
250 DATA 60,30,18,0C,18,30,60,00,3
C,66,1FE                          (28D3)
260 DATA 66,0C,18,00,18,00,7C,C6,D
E,DE,3A0                          (2813)
270 DATA DE,C0,7C,00,0C,1E,36,66,7
E,66,3C4                          (2884)
280 DATA 67,00,3E,63,63,7E,63,63,3
E,00,2ED                          (2861)
290 DATA 3E,63,60,60,60,63,3E,00,3
C,66,304                          (2886)
300 DATA 63,63,63,66,3C,00,3E,63,6
0,78,344                          (282A)
310 DATA 60,63,3E,00,3E,63,60,78,6
0,60,33A                          (2816)
320 DATA 20,00,3E,63,60,60,66,63,3
E,00,288                          (28E9)
330 DATA 22,63,63,7F,63,63,22,00,3
C,18,2A3                          (28DC)
340 DATA 18,18,18,18,3C,00,02,03,0
3,03,0A7                          (2864)
350 DATA 03,63,3E,00,23,66,6C,78,7
C,66,2F3                          (287A)
360 DATA 23,00,20,60,60,60,61,63,3
F,00,266                          (28CB)
370 DATA 22,77,6B,6B,63,63,22,00,3
2,73,2FC                          (28F7)
380 DATA 6B,6B,6B,67,26,00,3E,63,6
3,63,335                          (28DE)
390 DATA 63,63,3E,00,3E,63,63,7E,6
0,60,346                          (28CF)
400 DATA 20,00,3E,63,63,63,6B,65,3
A,00,291                          (2855)
410 DATA 3E,63,63,7E,6C,66,23,00,3
E,63,318                          (2886)
420 DATA 60,3E,03,63,3E,00,7E,5A,1
8,18,24A                          (28B0)
430 DATA 18,18,3C,00,22,63,63,63,6
3,63,27D                          (2814)
440 DATA 3E,00,22,63,63,63,63,36,1
C,00,23E                          (28D8)
450 DATA 22,63,63,63,6B,6B,36,00,6
3,36,2F0                          (2895)
460 DATA 1C,1C,1C,36,63,00,24,66,6
6,3C,219                          (2854)
470 DATA 18,18,3C,00,7F,43,04,1E,0
8,31,189                          (287A)
480 DATA 7F,00,3C,30,30,30,30,30,3
C,00,1E7                          (2877)
490 DATA C0,60,30,18,0C,06,02,00,3
C,0C,1C4                          (2836)
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7
E,18,156                          (28F6)
510 DATA 18,18,18,00,00,00,00,00,0
0,00,048                          (2844)
520 DATA 00,FF,30,18,0C,00,00,00,0
0,00,153                          (28F1)
530 DATA 00,00,78,0C,7C,CC,76,00,E
0,60,382                          (28EF)
540 DATA 7C,66,66,66,DC,00,00,00,3
C,66,32C                          (28D7)
550 DATA 60,66,3C,00,1C,0C,7C,CC,C
C,CC,40A                          (2837)
560 DATA 76,00,00,00,3C,66,7E,60,3
C,00,232                          (28F3)
570 DATA 1C,36,30,78,30,30,78,00,0
0,00,1D2                          (2811)
580 DATA 3E,66,66,3E,06,7C,E0,60,6
C,76,3EC                          (28C7)
590 DATA 66,66,E6,00,18,00,38,18,1
8,18,24A                          (28BC)
600 DATA 3C,00,06,00,0E,06,06,66,6
6,3C,164                          (2828)
610 DATA E0,60,66,6C,78,6C,E6,00,3
8,18,42C                          (284B)
620 DATA 18,18,18,18,3C,00,00,00,6
C,FE,206                          (28FB)
630 DATA D6,D6,C6,00,00,00,DC,66,6
6,66,480                          (28D7)
640 DATA 66,00,00,00,3C,66,66,66,3
C,00,210                          (289E)
650 DATA 00,00,DC,66,66,7C,60,F0,0
0,00,374                          (2880)
660 DATA 76,CC,CC,7C,0C,1E,00,00,D
C,76,406                          (28B9)
670 DATA 60,60,F0,00,00,00,3C,60,3
C,06,28E                          (284D)
680 DATA 7C,00,30,30,7C,30,30,36,1
C,00,20A                          (2893)
690 DATA 00,00,66,66,66,66,3E,00,0
0,00,1D6                          (28AC)
700 DATA 66,66,66,3C,18,00,00,00,C
6,D6,322                          (289E)
710 DATA D6,FE,6C,00,00,00,C6,6C,3
8,6C,416                          (2876)
720 DATA C6,00,00,00,66,66,66,3E,0
6,7C,2B8                          (28A4)
730 DATA 00,00,7E,4C,18,32,7E,00,0
E,18,1B8                          (28F3)
740 DATA 18,70,18,18,0E,00,18,18,1
8,18,126                          (282A)
750 DATA 18,18,18,00,70,18,18,0E,1
8,18,126                          (281A)
760 DATA 70,00,76,DC,00,00,00,00,0
0,00,1C2                          (28B5)
770 DATA CC,33,CC,33,CC,33,CC,33,0
0,3FC                             (2511)
780 dat=0:sz=0:dz=10              (1E65)
790 FOR adr=&A000 TO &A300        (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35FB)
810 sz=sz+byte                    (183D)
```

CPC Graphic Designer

```
820 POKE adr,byte                          (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                         (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                     (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                           (30EA)
860 dz=dz+10:sz=0:dat=0                    (24DC)
870 NEXT                                   (0692)
880 SAVE"G3M.BIN",B,&A000,&300             (1C98)
```

```
1 '****************************** (2506)
2 '*       DATALADER G30.LAD      * (256B)
3 '*        ERZEUGT G30.BIN       * (25C4)
4 '*    WIRD BENOETIGT FUER       * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464          ME* (2593)
7 '****************************** (2512)
10 DATA 00,00,00,00,00,00,00,00,3C
,3C,078                                    (2888)
20 DATA 3C,3C,3C,00,3C,3C,77,77,77
,00,291                                    (2822)
30 DATA 00,00,00,00,36,7F,7F,36,7F
,7F,268                                    (2894)
40 DATA 36,00,1C,7F,7D,3E,5F,7F,1C
,00,286                                    (28D1)
50 DATA 7B,77,6E,5C,3B,77,6F,00,7C
,5D,3B6                                    (2894)
60 DATA 6F,7E,7E,77,7B,00,0E,1C,38
,00,2BF                                    (2847)
70 DATA 00,00,00,00,0E,1C,38,38,38
,1C,0EE                                    (28D7)
80 DATA 0E,00,38,1C,0E,0E,0E,1C,38
,00,0E0                                    (284C)
90 DATA 08,6B,3E,1C,3E,6B,08,00,1C
,1C,1B6                                    (28C3)
100 DATA 7F,7F,7F,1C,1C,00,00,00,0
0,00,1B5                                   (282E)
110 DATA 00,18,18,30,00,00,00,7F,7
F,00,15E                                   (2887)
120 DATA 00,00,00,00,00,00,1C,1C,1
C,00,054                                   (288B)
130 DATA 03,07,0E,1C,38,70,60,00,7
F,6F,22A                                   (28F2)
140 DATA 5F,7F,7D,7B,7F,00,7E,3E,3
E,3E,38D                                   (2854)
150 DATA 3E,3E,7F,00,7F,5F,3E,7D,7
B,77,386                                   (28E4)
160 DATA 7F,00,7F,3F,5F,7F,3F,5F,7
F,00,338                                   (28EF)
170 DATA 7E,5E,6F,77,7F,3E,3E,00,7
F,7D,3B9                                   (2885)
180 DATA 7E,3F,5F,6F,7F,00,7F,7D,7
E,6F,3F3                                   (2827)
190 DATA 77,7B,7F,00,7F,7F,7F,0F,1
7,1F,333                                   (288F)
200 DATA 1F,00,7F,5F,6F,7F,7B,7D,7
F,00,362                                   (283B)
210 DATA 7F,5F,6F,7F,3F,5F,7F,00,0
0,00,2E9                                   (2893)
220 DATA 1C,1C,00,1C,1C,00,00,00,1
C,1C,0A8                                   (28CB)
230 DATA 00,1C,1C,38,0C,1C,38,70,3
8,1C,194                                   (2893)
240 DATA 0C,00,00,00,3E,3E,00,3E,3
E,00,104                                   (2835)
250 DATA 30,38,1C,0E,1C,38,30,00,7
F,6F,204                                   (2872)
260 DATA 5F,3E,1C,00,1C,00,7F,73,6
D,6B,29F                                   (28A3)
270 DATA 6F,70,7F,00,7F,77,7B,7F,6
F,77,434                                   (288A)
280 DATA 7B,00,7F,6F,77,7E,6F,77,7
F,00,3C3                                   (2846)
290 DATA 7F,7F,7B,7D,7E,7F,7F,00,7
E,7F,46F                                   (28D3)
300 DATA 6F,77,7B,7F,7E,00,7F,7D,7
E,7F,457                                   (289B)
310 DATA 7D,7E,7F,00,7F,7D,7E,7F,7
F,7E,470                                   (2894)
320 DATA 7E,00,7F,7F,7E,7D,73,7F,7
F,00,3E8                                   (2837)
330 DATA 7B,77,7F,7F,7B,77,6F,00,7
F,7F,44F                                   (2885)
340 DATA 7F,1C,7F,7F,7F,00,7F,7F,3
F,5F,3B4                                   (28D4)
350 DATA 6F,7F,7E,00,6F,5F,7F,7C,7
F,5F,413                                   (2882)
360 DATA 6F,00,7C,7C,7B,77,7F,7F,7
F,00,3D6                                   (28CA)
370 DATA 6F,77,7B,7F,7F,5B,6D,00,7
B,7D,41F                                   (2872)
380 DATA 7F,7F,7F,5F,6F,00,7F,7F,6
F,77,42F                                   (28AD)
390 DATA 7B,7F,7F,00,7F,6F,77,7F,7
F,7C,458                                   (287E)
400 DATA 7C,00,7F,7F,77,77,7E,7F,7
B,00,3E0                                   (2858)
410 DATA 7F,7F,67,7F,76,7B,7D,00,7
F,7B,44C                                   (28F3)
420 DATA 7D,3E,5F,6F,7F,00,7F,7F,5
D,3E,3A1                                   (28B0)
430 DATA 7F,7F,7F,00,5F,6F,77,7B,7
F,7F,43B                                   (281D)
440 DATA 7F,00,5F,6F,77,7B,7F,7F,1
C,00,359                                   (289D)
450 DATA 5B,6D,7F,7F,7F,77,73,00,7
7,7F,425                                   (2865)
460 DATA 7F,3E,7F,7F,77,00,6F,77,7
B,3F,3D2                                   (28FE)
470 DATA 5F,7F,7F,00,7F,5F,3F,7F,7
E,7D,3F4                                   (285B)
480 DATA 7F,00,7E,7E,78,78,78,7E,7
E,00,3DF                                   (286A)
490 DATA 60,70,38,1C,0E,07,03,00,7
E,7E,238                                   (28DF)
500 DATA 1E,1E,1E,7E,7E,00,1C,3E,7
F,7F,2AE                                   (2817)
510 DATA 1C,1C,1C,00,00,00,00,00,0
0,00,054                                   (28B5)
520 DATA 7F,7F,38,1C,0E,00,00,00,0
0,00,160                                   (2869)
530 DATA 00,00,7E,3E,6E,76,7E,00,7
0,70,2FE                                   (2823)
540 DATA 7E,6E,76,7A,7E,00,00,00,7
E,76,34E                                   (28CD)
550 DATA 7A,7C,7E,00,0E,0E,7E,5E,6
E,76,350                                   (2841)
560 DATA 7E,00,00,00,7E,5E,6E,7C,7
E,00,2C2                                   (28B1)
570 DATA 1E,1C,1A,3E,3E,3C,3C,00,0
0,00,148                                   (28C0)
580 DATA 7E,6E,76,3E,5E,7E,70,70,7
E,7E,458                                   (28DD)
590 DATA 6E,76,7A,00,38,00,78,78,3
8,3C,2FA                                   (2883)
600 DATA 3C,00,1E,00,1E,2E,36,3E,3
E,1C,174                                   (28AA)
610 DATA 60,62,66,6C,7C,76,7A,00,3
C,1C,358                                   (2878)
620 DATA 1C,1C,1C,1C,1C,00,00,00,6
E,76,170                                   (28FC)
630 DATA 7E,56,6A,00,00,00,7F,7F,6
F,77,322                                   (28BA)
640 DATA 7B,00,00,00,7F,6F,77,7B,7
F,00,2DA                                   (28B8)
650 DATA 00,00,7E,6E,76,7E,70,70,0
0,00,2C0                                   (2849)
660 DATA 7C,6C,74,7C,0E,0E,00,00,3
E,36,268                                   (2864)
670 DATA 3A,38,38,00,00,00,7E,7A,3
C,5E,23C                                   (28A7)
680 DATA 7E,00,00,18,3C,3C,38,38,1
C,00,19A                                   (2887)
690 DATA 00,00,6E,76,7A,7E,3E,00,0
0,00,21A                                   (2817)
700 DATA 6E,76,7E,3C,18,00,00,00,5
6,6A,276                                   (287F)
710 DATA 7E,6E,76,00,00,00,6E,76,3
C,6E,2F0                                   (2817)
720 DATA 76,00,00,00,6E,76,3E,5E,6
E,3C,2A0                                   (284F)
730 DATA 00,00,7E,6C,5A,36,7E,00,1
E,1C,232                                   (2830)
740 DATA 7C,60,7C,1C,1E,00,3C,3C,3
C,3C,282                                   (2832)
750 DATA 3C,3C,3C,00,78,38,3E,06,3
E,38,21E                                   (2871)
760 DATA 78,00,00,7B,7F,6F,00,00,0
0,00,1E1                                   (28B7)
770 DATA 7F,7B,7D,3E,7F,3C,7F,00,0
0,2EF                                      (259C)
780 dat=0:sz=0:dz=10                       (1E65)
790 FOR adr=&A000 TO &A300                 (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                                 (35FB)
810 sz=sz+byte                             (183D)
820 POKE adr,byte                          (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                         (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                     (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                           (30EA)
860 dz=dz+10:sz=0:dat=0                    (24DC)
870 NEXT                                   (0692)
880 SAVE"G30.BIN",B,&A000,&300             (1CB7)
```

```
1 '****************************** (2506)
2 '*     DATALADER G3PAT.LAD     * (25FA)
3 '*      ERZEUGT G3PAT.BIN      * (255B)
4 '*    WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER     * (25E5)
6 '*           CPC 464          ME* (2593)
7 '****************************** (2512)
10 DATA FF,FF,FF,FF,FF,FF,FF,FF,18
,2C,83C                                    (280E)
20 DATA 4E,8F,FF,7F,3F,1F,44,66,33
,11,3A7                                    (2874)
30 DATA 11,33,66,44,72,FB,1B,67,E6
,D8,49B                                    (285F)
40 DATA DF,4E,FF,05,05,05,05,05,FF
,05,349                                    (288C)
50 DATA 66,99,3C,EF,FF,3C,99,66,77
,BB,596                                    (28F0)
60 DATA DD,EE,11,22,44,88,66,99,99
,66,4C8                                    (28E9)
70 DATA 66,99,99,66,00,7E,7E,62,62
,62,420                                    (28EC)
80 DATA 7E,00,18,1C,1E,1A,7A,B8,70
,00,28C                                    (287D)
90 DATA 80,31,00,89,04,20,22,84,C3
,FF,3C6                                    (2815)
100 DATA 66,4A,5A,66,FF,C3,26,27,C
3,00,442                                   (2840)
110 DATA 62,72,3C,00,81,7E,5A,6E,7
E,5A,3AF                                   (2889)
120 DATA 7E,81,78,12,BA,EE,BA,90,3
C,00,4B7                                   (288D)
130 DATA C3,E7,7E,3C,3C,7E,E7,C3,C
3,81,60C                                   (28C3)
140 DATA 18,2C,3C,18,81,C2,00,78,5
A,7A,327                                   (28E7)
150 DATA 7A,02,1E,00,00,38,4C,5C,7
D,39,230                                   (2849)
160 DATA 03,0E,00,1F,3F,7F,47,47,4
6,7C,23E                                   (28EE)
170 DATA FF,81,BD,A5,AD,AD,A1,BF,F
F,C1,75C                                   (28C4)
180 DATA E1,F1,F9,FD,FF,FF,3C,42,9
9,A5,782                                   (287A)
190 DATA A5,99,42,3C,54,F8,F8,70,2
A,1F,4B9                                   (28F7)
200 DATA 1F,0E,C3,E7,6E,1C,38,76,E
7,C3,4B9                                   (28B9)
210 DATA 22,22,44,44,88,88,88,77,A
A,AA,42F                                   (2895)
220 DATA AA,AA,AA,AA,AA,AA,FF,00,F
F,00,5FA                                   (281E)
230 DATA FF,00,FF,00,FF,AA,FF,AA,F
F,AA,6F9                                   (2859)
240 DATA FF,AA,0F,0F,0F,0F,F0,F0,F
0,F0,5A5                                   (285A)
```

```
250 DATA 33,33,CC,CC,33,33,CC,CC,5
5,AA,4FB                                (2879)
260 DATA 55,AA,55,AA,55,AA,FF,81,B
D,A5,5DF                                (28E9)
270 DATA A5,BD,81,FF,18,18,18,FF,F
F,18,540                                (282F)
280 DATA 18,18,03,07,0E,1C,38,70,E
0,C0,2AC                                (2857)
290 DATA C0,E0,70,38,1C,0E,07,03,1
0,38,2C4                                (28DD)
300 DATA 7C,FE,7C,38,10,00,99,5A,2
4,C3,418                                (28BA)
310 DATA C3,24,5A,99,22,FF,22,22,2
2,FF,460                                (28BC)
320 DATA 22,22,44,88,11,22,44,88,1
1,22,242                                (28E7)
330 DATA 22,11,88,44,22,11,88,44,8
8,11,297                                (28BD)
340 DATA 11,88,44,22,22,44,FF,F0,8
0,80,454                                (2885)
350 DATA FF,0F,08,08,00,00,6C,92,8
2,44,2E2                                (286F)
360 DATA 28,10,FF,80,80,80,FF,08,0
8,08,3CE                                (2873)
370 DATA F9,09,09,F9,9F,90,90,9F,8
8,FF,5E9                                (2824)
380 DATA 22,FF,88,FF,22,FF,30,48,8
4,03,4C8                                (2804)
390 DATA 30,48,84,03,70,E0,C1,83,0
7,0E,3A8                                (2859)
400 DATA 1C,38,00,40,00,04,00,40,0
0,04,0DC                                (28C2)
410 DATA 25,29,49,4A,52,92,94,A4,5
5,55,3A7                                (2801)
420 DATA 55,55,AA,AA,AA,AA,18,3C,6
6,66,472                                (2889)
430 DATA 7E,66,66,00,FC,66,66,7C,6
6,66,45A                                (28A9)
440 DATA FC,00,3C,66,C0,C0,C0,66,3
C,00,480                                (285B)
450 DATA F8,6C,66,66,66,6C,F8,00,F
E,62,55A                                (280F)
460 DATA 68,78,68,62,FE,00,FE,62,6
8,78,4E8                                (288E)
470 DATA 68,60,F0,00,3C,66,C0,C0,C
E,66,50E                                (2833)
480 DATA 3E,00,66,66,66,7E,66,66,6
6,00,320                                (28B8)
490 DATA 7E,18,18,18,18,18,7E,00,3
C,66,216                                (2849)
500 DATA 60,3C,06,66,3C,00,1E,0C,0
C,0C,186                                (2843)
510 DATA CC,CC,78,00,E6,66,6C,78,6
C,66,512                                (288E)
520 DATA E6,00,F0,60,60,60,62,66,F
E,00,4BC                                (28A9)
530 DATA C6,EE,FE,FE,D6,C6,C6,00,C
6,E6,7BE                                (28B9)
540 DATA F6,DE,CE,C6,C6,00,38,6C,C
6,C6,65E                                (287D)
550 DATA C6,6C,38,00,FC,66,66,7C,6
0,60,46E                                (285C)
560 DATA F0,00,38,6C,C6,C6,DA,CC,7
6,00,53C                                (285D)
570 DATA FC,66,66,7C,6C,66,E6,00,7
E,5A,4D4                                (28EA)
580 DATA 18,18,18,18,3C,00,66,66,6
6,66,234                                (28DF)
590 DATA 66,66,3C,00,66,66,66,66,6
6,3C,342                                (2818)
600 DATA 18,00,C6,C6,C6,D6,FE,EE,C
6,00,5F2                                (289C)
610 DATA C6,6C,38,38,6C,C6,C6,00,6
6,66,466                                (2886)
620 DATA 66,3C,18,18,3C,00,FE,C6,8
C,18,376                                (2826)
630 DATA 32,66,FE,00,44,AA,44,00,4
4,AA,3B6                                (28F6)
640 DATA 44,00,AA,E3,AA,3E,AA,E3,A
A,3E,52E                                (2884)
650 DATA 00,3C,72,7A,7E,7E,3C,00,7
0,32,302                                (286C)
660 DATA 07,37,23,00,23,72,34,34,3
4,76,208                                (287B)
670 DATA 7E,34,34,34,7E,97,CB,A5,9
3,89,4BB                                (28AF)
680 DATA C5,7E,7E,C1,83,9F,95,9D,B
3,7E,607                                (286E)
690 DATA 2A,C9,08,FF,00,C9,2A,2A,6
6,9B,418                                (2819)
700 DATA BF,BF,DF,6E,3C,18,FF,E3,C
5,F9,6BF                                (28ED)
710 DATA 89,8B,8F,FF,66,C3,AD,38,1
C,B5,581                                (282E)
720 DATA C3,66,91,42,24,90,09,24,4
2,89,3A8                                (2898)
730 DATA 38,74,EE,47,83,C5,EE,5C,F
F,FF,671                                (28F0)
740 DATA FF,FF,FF,FF,FF,FF,FF,54,A
8,50,845                                (288F)
750 DATA A0,40,80,00,AA,55,2A,15,0
A,05,2AD                                (28BC)
760 DATA 02,01,01,02,05,0A,15,2A,5
5,AA,153                                (28EA)
770 DATA 00,80,40,A0,50,A8,54,AA,0
0,356                                   (2598)
780 dat=0:sz=0:dz=10                    (1E65)
790 FOR adr=&A000 TO &A300              (1400)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35FB)
810 sz=sz+byte                          (183D)
820 POKE adr,byte                       (142A)
830 IF dat<10 AND adr<&A300 THEN 8
70                                      (1EEA)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30EA)
860 dz=dz+10:sz=0:dat=0                 (24DC)
870 NEXT                                (0692)
880 SAVE"G3PAT.BIN",B,&A000,&300        (1E44)
```

```
1 '******************************* (2506)
2 '*       DATALADER G4.LAD       * (2522)
3 '*        ERZEUGT G4.BIN        * (259A)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464         ME* (2593)
7 '******************************* (2512)
10 DATA 00,01,14,5F,21,10,5F,CD,D1
,BC,35E                                 (28DC)
20 DATA 3E,C9,32,00,5F,C9,00,00,00
,00,261                                 (28F8)
30 DATA 1C,5F,C3,00,BE,C3,28,5F,4D
,45,3D8                                 (28E8)
40 DATA 52,47,C5,43,48,41,4E,47,C5
,00,384                                 (281E)
50 DATA FE,06,C0,DD,4E,04,DD,46,06
,C5,4E1                                 (2890)
60 DATA 06,00,DD,6E,08,DD,66,09,DD
,5E,3E0                                 (2834)
70 DATA 0A,DD,56,0B,09,C1,C5,48,06
,00,325                                 (28F0)
80 DATA EB,09,EB,C1,C5,E5,C5,D5,CD
,F0,7A1                                 (288D)
90 DATA BB,D1,C1,E1,E5,C5,D5,DD,BE
,02,74A                                 (2812)
100 DATA CC,79,5F,CD,DE,BB,D1,C1,E
1,E5,762                                (28B3)
110 DATA C5,D5,CD,EA,BB,D1,C1,E1,1
B,10,6AA                                (283E)
120 DATA DC,C1,0D,1AA                   (13C6)
130 dat=0:sz=0:dz=10                    (1E4F)
140 FOR adr=&5F00 TO &5F70              (1445)
150 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                              (35E5)
160 sz=sz+byte                          (1827)
170 POKE adr,byte                       (1412)
180 IF dat<10 AND adr<&5F70 THEN 2
20                                      (1E59)
190 READ chksum$:chksum=VAL("&"+ch
ksum$)                                  (2BA8)
200 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                        (30D2)
210 dz=dz+10:sz=0:dat=0                 (24C4)
220 NEXT                                (067A)
230 SAVE"G4.BIN",B,&5F00,&70            (1BD4)
```

```
1 '******************************* (2506)
2 '*       DATALADER G5.LAD       * (25C1)
3 '*        ERZEUGT G5.BIN        * (25DD)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464         ME* (2593)
7 '******************************* (2512)
10 DATA 01,93,54,21,8F,54,CD,D1,BC
,3E,484                                 (2881)
20 DATA C9,32,80,54,C9,00,00,00,00
,A1,339                                 (286C)
30 DATA 54,C3,BB,54,C3,73,55,C3,A0
,55,569                                 (28F3)
40 DATA C3,E7,55,4D,4F,44,C5,42,45
,4E,479                                 (2883)
50 DATA 44,2E,49,CE,42,45,4E,44,2E
,4F,31F                                 (28E2)
60 DATA 55,D4,49,4E,56,45,52,D4,00
,FE,47F                                 (2840)
70 DATA 01,C0,DD,7E,00,FE,01,CA,24
,55,45E                                 (2815)
80 DATA 21,00,C0,11,00,60,01,FF,3F
,ED,37E                                 (2860)
90 DATA B0,3E,02,CD,0E,BC,21,00,60
,11,319                                 (28E8)
100 DATA 00,C0,01,FF,3F,ED,B0,21,0
0,C0,47D                                (281C)
110 DATA 7E,47,EE,0F,A0,47,EE,70,A
0,4F,4F6                                (28E1)
120 DATA CB,3F,B1,57,78,EE,B0,A0,C
B,3F,5D2                                (2882)
130 DATA 4F,CB,3F,B1,B2,57,78,EE,D
0,A0,5E9                                (28CA)
140 DATA CB,3F,CB,3F,4F,CB,3F,B1,B
2,57,527                                (2811)
150 DATA 78,EE,E0,A0,CB,3F,CB,3F,C
B,3F,604                                (281D)
160 DATA 4F,CB,3F,B1,B2,77,23,3E,0
0,BC,450                                (28A3)
170 DATA C8,C3,E4,54,21,00,C0,11,0
0,60,415                                (2811)
180 DATA 01,FF,3F,ED,B0,3E,01,CD,0
E,BC,4B2                                (2802)
190 DATA 21,00,60,11,00,C0,01,FF,3
F,ED,37E                                (28E0)
200 DATA B0,21,00,C0,7E,47,EE,55,A
0,47,480                                (28DF)
210 DATA EE,7F,A0,57,78,CB,27,4F,E
E,BF,5CA                                (28C1)
220 DATA A1,B2,57,78,EE,F7,A0,CB,2
7,CB,664                                (2844)
230 DATA 27,B2,57,78,EE,FD,A0,CB,2
7,CB,5F0                                (284B)
240 DATA 27,CB,27,B2,77,23,3E,00,B
C,C8,427                                (287F)
250 DATA C3,42,55,FE,01,C0,DD,66,0
1,DD,53A                                (2800)
260 DATA 6E,00,11,80,02,01,00,56,E
5,D5,312                                (28EB)
270 DATA C5,CD,F0,BB,C1,D1,E1,02,0
3,1B,5D0                                (28E6)
280 DATA 3E,00,BB,CA,99,55,3E,00,C
3,82,434                                (2828)
290 DATA 55,3E,00,BA,C8,C3,94,55,F
E,02,4C1                                (286B)
300 DATA C0,DD,56,03,DD,5E,02,DD,6
6,01,477                                (282C)
310 DATA DD,6E,00,01,80,02,C5,01,0
0,56,2EA                                (2869)
320 DATA ED,43,00,59,C1,C5,ED,4B,0
0,59,4A0                                (28A8)
330 DATA 0A,CD,DE,BB,E5,D5,CD,EA,B
B,D1,76D                                (28FA)
340 DATA E1,ED,4B,00,59,03,ED,43,0
0,59,3FE                                (28D1)
350 DATA C1,0B,1B,3E,00,B9,CA,E0,5
```

```
5,C3,4A0                                    (28A4)
360 DATA BB,55,3E,00,B8,C8,C3,DD,5
5,21,4E4                                    (28AD)
370 DATA 00,C0,7E,EE,FF,77,23,3E,0
0,BC,4BF                                    (28A0)
380 DATA C8,C3,EA,55,00,00,00,00,0
0,00,2CA                                    (28F4)
390 DATA 00,00,00,00,00,000                 (1901)
400 dat=0:sz=0:dz=10                        (1E6C)
410 FOR adr=&5480 TO &5600                  (1444)
420 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                                  (3501)
430 sz=sz+byte                              (1844)
440 POKE adr,byte                           (142F)
450 IF dat<10 AND adr<&5600 THEN 4
90                                          (1EB2)
460 READ chksum$:chksum=VAL("&"+ch
ksum$)                                      (2BC5)
470 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                            (30EF)
480 dz=dz+10:sz=0:dat=0                     (24E1)
490 NEXT                                    (0697)
500 SAVE"G5.BIN",B,&5480,&180               (1B89)
```

```
1 '******************************* (2506)
2 '*       DATALADER G6.LAD       * (2562)
3 '*         ERZEUGT G6.BIN       * (2517)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464          ME* (2593)
7 '******************************* (2512)
10 DATA DF,04,A0,C9,07,A0,FE,D5,F5
,E5,6A0                                     (28A1)
20 DATA C5,11,5A,A0,CB,45,28,01,13
,CB,3E7                                     (28F9)
30 DATA 5C,28,02,13,13,CB,64,28,04
,13,21A                                     (2883)
40 DATA 13,13,13,CB,6C,28,08,13,13
,13,1D9                                     (28A2)
50 DATA 13,13,13,13,13,1A,47,CD,68
,0C,201                                     (286A)
60 DATA C1,E1,F1,D1,C9,3A,6A,A0,CD
,A5,6E3                                     (2855)
70 DATA BB,D0,11,5A,A0,CD,53,BC,21
,5A,4ED                                     (280F)
80 DATA A0,06,10,7E,4F,3A,6B,A0,77
,3A,379                                     (2858)
90 DATA 6C,A0,AE,A1,AE,77,23,10,F0
,C9,56C                                     (2845)
100 DATA 0F,1F,9F,DF,9F,7F,8F,359 (1F99)
110 dat=0:sz=0:dz=10                        (1E27)
120 FOR adr=&A000 TO &A060                  (14B1)
130 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                                  (35BD)
140 sz=sz+byte                              (18FE)
150 POKE adr,byte                           (14EB)
160 IF dat<10 AND adr<&A060 THEN 2
00                                          (1ED8)
170 READ chksum$:chksum=VAL("&"+ch
ksum$)                                      (2B80)
180 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                            (30AA)
190 dz=dz+10:sz=0:dat=0                     (249C)
200 NEXT                                    (0652)
210 SAVE"G6.BIN",B,&A000,&60                (1B62)
```

```
1 '******************************* (2506)
2 '*       DATALADER G7.LAD       * (257C)
3 '*         ERZEUGT G7.BIN       * (2557)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*   CPC GRAPHICS DESIGNER      * (25E5)
6 '*           CPC 464          ME* (2593)
7 '******************************* (2512)
10 DATA 00,01,84,51,21,80,51,CD,D1
,BC,422                                     (2870)
20 DATA 3E,C9,32,70,51,C9,00,00,00
,00,2C3                                     (2872)
30 DATA 9E,51,C3,DE,51,C3,2C,52,C3
,5D,542                                     (286F)
40 DATA 53,C3,71,52,C3,00,54,C3,C6
,52,4CB                                     (28C4)
50 DATA C3,25,53,C3,4D,53,50,41,52
,54,3D5                                     (2832)
60 DATA 2E,49,CE,5A,4F,4F,4D,2E,4F
,55,35C                                     (287D)
70 DATA D4,50,41,52,54,2E,4D,41,C7
,50,3DE                                     (28DD)
80 DATA 41,52,54,2E,4F,55,D4,53,48
,52,37A                                     (2881)
90 DATA 49,4B,45,2E,49,CE,4D,49,52
,52,358                                     (28D0)
100 DATA 4F,52,2E,4F,55,D4,50,41,5
4,54,380                                    (28EA)
110 DATA 4F,CE,50,41,54,54,4F,46,C
6,00,3B1                                    (2893)
120 DATA FE,05,C0,21,00,56,22,2A,5
2,DD,3B5                                    (28F5)
130 DATA 46,02,DD,4E,04,C5,DD,6E,0
6,DD,46A                                    (28AA)
140 DATA 66,07,DD,5E,08,DD,56,09,1
B,2B,332                                    (28B0)
150 DATA 78,06,00,EB,09,4F,EB,09,0
9,3E,2FC                                    (2878)
160 DATA 02,DD,BE,00,CA,14,52,C1,C
5,06,459                                    (286D)
170 DATA 00,EB,09,EB,CD,F0,BB,2A,2
A,52,4FD                                    (282A)
180 DATA 77,23,22,2A,52,C1,0D,3E,0
0,B9,2FD                                    (2889)
190 DATA C2,ED,51,10,C1,C9,00,00,F
E,01,499                                    (28DD)
200 DATA C0,21,00,56,22,2A,52,06,2
8,3E,241                                    (2839)
210 DATA 02,DD,BE,00,CC,6E,52,78,3
2,6D,440                                    (28E2)
220 DATA 52,06,18,21,6D,52,4E,C5,3
E,19,2BA                                    (28B0)
230 DATA 90,6F,61,CD,75,BB,2A,2A,5
2,7E,481                                    (28F4)
240 DATA 23,22,2A,52,CD,90,BB,3E,8
F,CD,473                                    (2826)
250 DATA 5A,BB,C1,0D,3E,00,B9,C2,4
9,52,437                                    (285E)
260 DATA 10,D9,C9,00,CB,20,C9,FE,0
5,C0,529                                    (28AD)
270 DATA 21,00,56,22,2A,52,DD,46,0
2,DD,317                                    (28E0)
280 DATA 4E,04,C5,DD,6E,06,DD,66,0
7,DD,48F                                    (2813)
290 DATA 5E,08,DD,56,09,1B,2B,78,0
6,00,266                                    (2837)
300 DATA EB,09,4F,EB,09,09,3E,02,D
D,BE,41B                                    (2879)
310 DATA 00,CA,A7,52,C1,C5,06,00,E
B,09,443                                    (2833)
320 DATA EB,ED,4B,2A,52,0A,03,ED,4
3,2A,406                                    (2882)
330 DATA 52,E5,D5,CD,DE,BB,D1,E1,C
D,EA,7DB                                    (28B1)
340 DATA BB,C1,0D,3E,00,B9,C2,80,5
2,10,424                                    (288D)
350 DATA B8,C9,FE,05,C0,21,00,56,2
2,2A,407                                    (280E)
360 DATA 52,DD,46,02,DD,4E,04,C5,D
D,6E,4B6                                    (285D)
370 DATA 06,DD,66,07,DD,5E,08,DD,5
6,09,3CF                                    (28C1)
380 DATA 1B,2B,DD,7E,04,91,4F,78,0
6,00,303                                    (2801)
390 DATA EB,09,4F,EB,09,09,3E,02,D
D,BE,41B                                    (282E)
400 DATA 00,CA,06,53,C1,C5,06,00,D
D,7E,40A                                    (2882)
410 DATA 04,91,4F,EB,09,EB,ED,4B,2
A,52,477                                    (2861)
420 DATA 0A,03,ED,43,2A,52,E5,D5,C
D,DE,51E                                    (2871)
430 DATA BB,D1,E1,CD,EA,BB,C1,0D,3
E,00,5EB                                    (28A8)
440 DATA B9,C2,D5,52,10,AE,C9,FE,0
3,C0,5EA                                    (28EF)
450 DATA DD,7E,00,32,6B,A0,DD,7E,0
2,32,427                                    (2898)
460 DATA 6C,A0,DD,7E,04,32,6A,A0,C
D,37,4AB                                    (289C)
470 DATA A0,3E,C3,32,E8,BD,3E,00,3
2,E9,4D1                                    (28EA)
480 DATA BD,3E,A0,32,EA,BD,C9,3E,C
3,32,570                                    (280C)
490 DATA E8,BD,3E,68,32,E9,BD,3E,0
C,32,49F                                    (28E9)
500 DATA EA,BD,C9,FE,06,C0,21,00,5
6,22,4CD                                    (289B)
510 DATA 2A,52,DD,46,02,DD,4E,04,C
5,DD,472                                    (28EA)
520 DATA 6E,06,DD,66,07,DD,5E,08,D
D,56,434                                    (2803)
530 DATA 09,1B,2B,DD,7E,0A,06,00,E
B,09,2AE                                    (280C)
540 DATA 3D,FE,00,C2,81,53,C1,C5,4
8,06,4A5                                    (28D3)
550 DATA 00,EB,DD,7E,0A,09,09,3D,F
E,00,39D                                    (28C0)
560 DATA C2,91,53,3E,02,DD,BE,00,C
A,B1,4FC                                    (2838)
570 DATA 53,C1,C5,06,00,EB,DD,7E,0
A,09,438                                    (2838)
580 DATA 3D,FE,00,C2,A9,53,EB,ED,4
B,2A,546                                    (28DC)
590 DATA 52,0A,03,ED,43,2A,52,E5,D
5,CD,492                                    (288D)
600 DATA DE,BB,D1,E1,DD,7E,0A,87,F
5,E5,711                                    (2891)
610 DATA D5,CD,C0,BB,D1,E1,DD,4E,0
A,06,60A                                    (28D9)
620 DATA 00,D5,EB,ED,42,3E,02,DD,B
E,00,4CA                                    (284C)
630 DATA CA,E1,53,ED,42,EB,E5,CD,F
6,BB,77B                                    (283D)
640 DATA E1,D1,2B,F1,3D,FE,00,C2,C
6,53,5E4                                    (28F2)
650 DATA C1,0D,3E,00,B9,C2,6C,53,0
5,3E,389                                    (286B)
660 DATA 00,B8,C2,69,53,C9,FE,06,C
0,21,4E4                                    (28D6)
670 DATA 00,56,22,2A,52,DD,46,02,D
D,4E,344                                    (2828)
680 DATA 04,C5,DD,6E,06,DD,66,07,D
D,5E,49F                                    (2896)
690 DATA 08,DD,56,09,1B,2B,DD,7E,0
A,06,2F5                                    (2833)
700 DATA 00,EB,09,3D,FE,00,C2,24,5
4,C1,42A                                    (288A)
710 DATA C5,48,06,00,EB,DD,7E,0A,0
9,09,375                                    (2846)
720 DATA 3D,FE,00,C2,34,54,3E,02,D
D,BE,460                                    (287B)
730 DATA 00,CA,54,54,C1,C5,06,00,E
B,DD,4C6                                    (28E6)
740 DATA 7E,0A,09,3D,FE,00,C2,4C,5
4,EB,419                                    (28ED)
750 DATA CD,F0,BB,ED,4B,2A,52,02,0
3,ED,51E                                    (28F0)
760 DATA 43,2A,52,C1,0D,3E,00,B9,C
2,0F,355                                    (2864)
770 DATA 54,05,3E,00,B8,C2,0C,54,C
9,00,33A                                    (2818)
780 dat=0:sz=0:dz=10                        (1E65)
790 FOR adr=&5170 TO &5471                  (14F3)
800 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                                  (35FB)
810 sz=sz+byte                              (183D)
820 POKE adr,byte                           (142A)
830 IF dat<10 AND adr<&5471 THEN 8
70                                          (1E7D)
840 READ chksum$:chksum=VAL("&"+ch
ksum$)                                      (2BC0)
850 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                            (30EA)
860 dz=dz+10:sz=0:dat=0                     (24DC)
870 NEXT                                    (0692)
880 SAVE"G7.BIN",B,&5170,&301               (1BE3)
```

```
1 '****************************** (2506)
2 '*       DATALADER G8.LAD       * (259C)
3 '*        ERZEUGT G8.BIN        * (25A3)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*    CPC GRAPHICS DESIGNER     * (25E5)
6 '*           CPC 464          ME* (2593)
7 '****************************** (2512)
10 DATA 21,00,00,22,0F,5E,01,11,5E
,21,141                           (28E6)
20 DATA 19,5E,C3,D1,BC,00,00,15,5E
,18,352                           (2848)
30 DATA 0D,43,48,41,D2,00,A6,11,5E
,00,2C0                           (28FB)
40 DATA 00,00,00,00,FE,05,C0,DD,7E
,08,326                           (28FD)
50 DATA DD,6E,00,DD,66,01,DD,5E,02
,DD,4A9                           (28C7)
60 DATA 56,03,FD,21,1D,5E,F5,3E,08
,FD,42A                           (28E9)
70 DATA 77,00,FD,77,01,DD,7E,04,FD
,77,4BF                           (283F)
80 DATA 02,DD,7E,06,FD,77,03,AF,FD
,77,4FD                           (28DE)
90 DATA 04,F1,DF,56,5E,C9,59,5E,FE
,F5,5FB                           (28C3)
100 DATA CD,1D,BC,F1,E5,CD,A5,BB,E
B,E1,775                          (28A0)
110 DATA E5,1A,47,CB,00,30,1C,C5,3
A,38,394                          (2899)
120 DATA B3,47,CD,E8,BD,00,00,00,C
1,CB,4F8                          (2848)
130 DATA 09,DC,20,BC,FD,7E,03,3D,F
D,77,4F0                          (28C1)
140 DATA 03,20,E6,18,0E,CB,09,DC,2
0,BC,3BB                          (28C1)
150 DATA FD,7E,03,3D,FD,77,03,20,F
2,DD,521                          (2846)
160 DATA 7E,06,FD,77,03,FD,7E,00,3
D,FD,4B0                          (2809)
170 DATA 77,00,20,C3,3E,08,FD,77,0
0,E1,3F5                          (288C)
180 DATA CD,26,BC,FD,7E,04,00,00,0
0,3C,36A                          (2878)
190 DATA FD,77,04,00,C3,C3,5E,FD,7
7,04,4D4                          (283A)
200 DATA CB,01,DC,23,BC,FD,7E,02,3
D,FD,53E                          (2833)
210 DATA 77,02,20,98,13,DD,7E,04,F
D,77,417                          (2840)
220 DATA 02,FD,7E,01,3D,FD,77,01,2
0,88,3D8                          (28E4)
230 DATA C9,00,00,00,00,0C9       (1944)
240 dat=0:sz=0:dz=10              (1E2A)
250 FOR adr=&5E00 TO &5EE0        (1463)
260 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                        (35C1)
270 sz=sz+byte                    (1803)
280 POKE adr,byte                 (14EF)
290 IF dat<10 AND adr<&5EE0 THEN 3
30                                (1EA6)
300 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2B86)
310 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (30B0)
320 dz=dz+10:sz=0:dat=0           (24A2)
330 NEXT                          (0658)
340 SAVE"G8.BIN",B,&5E00,&E0      (1BB3)
```

```
1 '****************************** (2506)
2 '*       DATALADER G9.LAD       * (253C)
3 '*        ERZEUGT G9.BIN        * (25E2)
4 '*     WIRD BENOETIGT FUER      * (25CD)
5 '*    CPC GRAPHICS DESIGNER     * (25E5)
6 '*           CPC 464          ME* (2593)
7 '****************************** (2512)
10 DATA DD,6E,00,DD,66,01,46,23,5E
,23,379                           (28DD)
20 DATA 56,EB,11,DF,48,CD,77,BC,21
,00,49A                           (2881)
30 DATA C0,E5,CD,80,BC,D2,24,BE,E1
,AE,6F1                           (2827)
40 DATA 77,23,E5,C3,16,BE,E1,CD,7A
,BC,5FA                           (289B)
50 DATA C9,00,0C9                 (10F0)
60 dat=0:sz=0:dz=10               (1EC2)
70 FOR adr=&BE00 TO &BE29         (1497)
80 READ byte$:byte=VAL("&"+byte$):
dat=dat+1                         (3559)
90 sz=sz+byte                     (189A)
100 POKE adr,byte                 (1487)
110 IF dat<10 AND adr<&BE29 THEN 1
50                                (1ECE)
120 READ chksum$:chksum=VAL("&"+ch
ksum$)                            (2B1C)
130 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                  (3046)
140 dz=dz+10:sz=0:dat=0           (2438)
150 NEXT                          (06EF)
160 SAVE"G9.BIN",B,&BE00,&29      (1BBF)
```

```
1 '****************************** (2506)
2 '*       SLIDE SHOW zum         * (25FB)
3 '*   CPC GRAPHICS DESIGNER      * (25E1)
4 '*           von                * (25D9)
5 '*       JOERG SCHWIEDER        * (256C)
6 '*           fuer               * (2597)
7 '*         CPC  WELT            * (2570)
8 '*          CPC 464           ME* (251C)
9 '****************************** (2516)
10 MODE 1                         (07BD)
15 INK 2,11                       (0A80)
20 MEMORY &3FFF                   (0930)
30 LOAD"s2",&AF00:CALL &AF00      (135B)
40 DIM name$(10):DIM num(10)      (1D2F)
50 MODE 1:LOCATE 11,12:PRINT"INSER
T SOURCE DISC":LOCATE 12,25:PRINT"
<PRESS ANY KEY>":CALL &BB06       (4321)
60 GOSUB 380                      (092C)
70 MODE 1                         (0734)
80 RESTORE                        (06BF)
90 FOR adr=&A000 TO &A005         (14A0)
100 READ c:POKE adr,c             (17CD)
110 NEXT                          (069F)
120 CALL &A000,&C0                (0D41)
130 adr(0)=&40:adr(1)=&C0:addr(0)=
&4000:addr(1)=&C000:ad=1          (461B)
140 DATA &dd,&7e,&00,&c3,&08,&bc  (1E5B)
150 INK 0,26:INK 1,0:INK 2,11:INK
3,6:PEN 1                         (1D47)
160 BORDER 0                      (078D)
170 inki(0)=26:inki(1)=0:inki(2)=1
1:inki(3)=6                       (3A81)
180 IF b=0 THEN 50                (1091)
190 PRINT:PRINT:PRINT:PRINT"
    S L I D E S H O W             (29D5)
200 PRINT:PRINT"      (C) 1987 by
Joerg Schwieder                   (2ADC)
210 FOR a=1 TO b                  (11A4)
220 IF ad=1 THEN ad=0 ELSE ad=1   (1EDA)
230 LOAD name$(a),addr(ad)        (2202)
240 modus=PEEK(addr(ad)+&3FFB)    (24B4)
250 FOR i=0 TO 3:inki(i)=PEEK(addr
(ad)+&3FFC+i):NEXT                (3A8F)
260 CALL &BB06                    (0956)
270 FOR n=40 TO 0 STEP-1          (12A7)
280 OUT &BC00,1:OUT &BD00,n       (152C)
290 NEXT                          (0608)
300 MODE modus                    (0E1F)
310 CALL &A000,adr(ad)            (17E4)
320 FOR i=0 TO 3:INK i,inki(i):NEX
T                                 (24B8)
330 FOR n=1 TO 40                 (0F79)
340 OUT &BC00,1:OUT &BD00,n       (15A4)
350 NEXT                          (0680)
360 NEXT                          (0694)
370 CALL &BB06:GOTO 50            (0E4D)
380 INK 0,26:INK 1,26             (1050)
390 MODE 2:CAT                    (09DA)
400 y=4:x=1:b=0                   (194E)
410 nam$="":a$=" "                (1783)
420 FOR n=1 TO 17                 (0FD1)
430 LOCATE x+n-1,y:/COPYCHR,@a$   (2651)
440 nam$=nam$+a$                  (1788)
450 NEXT                          (0647)
460 IF RIGHT$(nam$,3)="17K"THEN b=
b+1:name$(b)=LEFT$(nam$,12):y=y+1:
GOTO 410                          (5180)
470 IF LEFT$(nam$,1)=" "AND x<61 T
HEN x=x+20:y=4:GOTO 410           (3751)
480 IF LEFT$(nam$,1)<>" "THEN y=y+
1:GOTO 410                        (27C1)
490 MODE 1                        (077E)
500 INK 1,0                       (09C7)
510 FOR a=1 TO b                  (11FD)
520 LOCATE 2,a:PRINT name$(a)     (1BDF)
530 NEXT                          (06E8)
540 z=1                           (0B12)
550 LOCATE 2,z:PRINT CHR$(24)name$
(z)CHR$(24)                       (27F5)
560 WHILE INKEY$="":WEND          (0D20)
570 IF num(z)=0 THEN LOCATE 2,z:PR
INT name$(z)ELSE PAPER 2:LOCATE 2,
z:PRINT name$(z):PAPER 0          (49CE)
580 IF NOT INKEY(0)AND z>1 THEN z=
z-1                               (1F2A)
590 IF NOT INKEY(2)AND z<b THEN z=
z+1                               (22AE)
600 IF NOT INKEY(9)THEN IF num(z)=
1 THEN num(z)=0 ELSE num(z)=1     (3B30)
610 IF NOT INKEY(18)THEN 630      (11A0)
620 GOTO 550                      (0908)
630 z=0                           (0BA2)
640 FOR z=1 TO b                  (1134)
650 IF num(z)=1 THEN z=z+1:name$(z
)=name$(z)                        (3C71)
660 NEXT                          (06ED)
670 b=z:RETURN                    (10F3)
```

```
1 '****************************** (2506)
2 '*       CONVERTER fuer         * (25BB)
3 '*           s1.cvt             * (258D)
4 '*            von               * (25D9)
5 '*       JOERG SCHWIEDER        * (256C)
6 '*           fuer               * (2597)
7 '*         CPC  WELT            * (2570)
8 '*          CPC 464           ME* (251C)
9 '****************************** (2516)
10 DATA CD,4D,90,21,4A,90,11,80,BC
,06,3F8                           (2855)
20 DATA 03,7E,12,23,13,10,FA,C9,C5
,E5,446                           (2853)
30 DATA D5,CD,30,90,D1,E1,C1,F5,FE
,A3,76B                           (285B)
40 DATA CA,28,90,FE,7B,CA,2C,90,F1
,C9,63B                           (283D)
50 DATA F1,3E,18,C9,F1,3E,17,C9,21
,47,487                           (280A)
60 DATA 90,11,80,BC,06,03,7E,12,23
,13,2AC                           (2820)
70 DATA 10,FA,CD,80,BC,F5,CD,03,90
,F1,659                           (2856)
80 DATA C9,DF,8B,A8,C3,12,90,21,65
,90,556                           (2894)
90 DATA 11,04,AC,06,03,7E,12,13,23
,10,1A0                           (284B)
100 DATA FA,FE,17,C0,E1,C3,64,C0,C
```

Alle Programme auf Disc erhältlich!

Bestell-Coupon Seite 81

CPC Graphic Designer

```
3,64,6BE                                  (28FD)
110 DATA C0,C3,5B,90,00,26E               (19BB)
120 dat=0:sz=0:dz=10                      (1E3B)
130 FOR adr=&9000 TO &9068                (14C9)
140 READ byte$:byte=VAL("&"+byte$)
:dat=dat+1                                (35D1)
150 sz=sz+byte                            (1813)
160 POKE adr,byte                         (14FF)
170 IF dat<10 AND adr<&9068 THEN 2
10                                        (1E3D)
180 READ chksum$:chksum=VAL("&"+ch
ksum$)                                    (2B94)
190 IF chksum<>sz THEN PRINT"FEHLE
R IN ZEILE :";dz                          (30BE)
200 dz=dz+10:sz=0:dat=0                   (24B0)
210 NEXT                                  (0666)
230 MODE 2:PRINT:PRINT"  BITTE DIS
KETTE MIT DER ASCII-DATEI 'G1.CVT'
 EINLEGEN UND TASTE DRUECKEN"             (56BC)
240 CALL &BB06                            (092D)
250 PRINT:PRINT" LAUFFAEHIGES PROG
RAMM MIT SAVE'G1.BAS ABSPEICERN." (3C67)
255 POKE &AC00,255                        (0CC7)
260 CALL &9000:LOAD"g1.cvt"               (1320)
```

```
10 '****************************** (24C3)
20 '*       FONT DESIGNER        * (241D)
30 '* FUER CPC GRAPHICS DESIGNER* (24A3)
40 '*           von              * (240D)
50 '*       JOERG SCHWIEDER      * (2469)
60 '*           fuer             * (2410)
70 '*         CPC WELT           * (243E)
80 '*         CPC  464        ME* (24A1)
90 '****************************** (2462)
200 'HAUPTPROGRAMM                        (148A)
210 '                                     (07AE)
220 SYMBOL AFTER 256                      (0A41)
230 OPENOUT"dummy":MEMORY HIMEM-1:
CLOSEOUT                                  (1527)
240 SYMBOL AFTER 32                       (091B)
250 POKE &BDEE,&C9                        (0D61)
260 POKE &B295,0                          (0BD6)
270 MODE 1                                (07C6)
280 INK 0,26:BORDER 26:INK 1,0:INK
 2,13                                     (1996)
290 RESTORE 320:FOR n=1 TO 4              (1353)
300 READ funk$(n)                         (13B9)
310 NEXT                                  (0630)
320 DATA "EDIT SYMBOL"                    (14D5)
330 DATA "SAVE SET   "                    (1449)
340 DATA "LOAD SET   "                    (142E)
350 DATA "DELETE SET "                    (144D)
360 x1=13:x2=21:y1=4:y2=6:GOSUB 61
0:PRINT:PRINT CHR$(24)"M E N U"CHR
$(24)                                     (4412)
370 x1=11:x2=23:y1=8:y2=16                (279D)
380 GOSUB 610                             (09E2)
390 a=2                                   (0B5A)
400 FOR n=1 TO 4                          (0EA8)
410 LOCATE 2,n*2:PRINT funk$(n):NE
XT                                        (1FE2)
420 LOCATE 2,a:PRINT CHR$(24)+funk
$(a\2);CHR$(24);                          (2C77)
430 WHILE INKEY$="":WEND                  (0D1C)
440 LOCATE 2,a:PRINT funk$(a/2):SO
UND 1,45,3,15                             (28A2)
450 IF NOT INKEY(0)OR NOT INKEY(72
)THEN a=a-2:IF a<2 THEN a=8               (2F36)
460 IF NOT INKEY(2)OR NOT INKEY(73
)THEN a=a+2:IF a>8 THEN a=2               (2F7E)
470 IF NOT INKEY(18)OR NOT INKEY(7
6)THEN 490                                (1949)
480 GOTO 420                              (09D7)
490 a=a\2                                 (10F4)
500 ON a GOSUB 840,1410,1550,1600         (1AE1)
510 GOTO 260                              (0917)
590 'UNTERPROGRAMM WINDOWS                (1CD9)
600 '                                     (07BD)
610 WINDOW x1,x2,y1,y2                    (1D63)
620 x1=x1*16-17                           (1664)
630 x2=x2*16+1                            (1580)
640 y1=400-(y1*16-17)                     (1C97)
650 y2=400-(y2*16+1)                      (1B18)
660 MOVE x1,y1:DRAW x2,y1,1:DRAW x
2,y2:DRAW x1,y2:DRAW x1,y1                (4782)
670 x1=x1+2:x2=x2+2                       (2028)
680 y1=y1-2:y2=y2-2                       (2066)
690 MOVE x1+2,y2:DRAW x2,y2:DRAW x
2,y1-2                                    (2F51)
700 x1=x1+2:x2=x2+2                       (2064)
710 y1=y1-2:y2=y2-2                       (20A2)
720 MOVE x1,y2:DRAW x2,y2:DRAW x2,
y1                                        (2BE5)
730 CLS:SOUND 1,45,3,15:RETURN            (13B2)
810 'UNTERPROGRAMM SYMBOLDEFINITIO
N                                         (25A8)
820 '                                     (0776)
830 POKE &B295,255                        (0C6D)
840 x1=9:x2=20:y1=13;y2=17:GOSUB 6
10:PRINT:PRINT" SYMBOLNR.?":PRINT:
INPUT"    ",symnr                         (4F60)
850 IF symnr<32 OR symnr>128 THEN
260                                       (211A)
860 FOR n=0 TO 7:sy$(n)=BIN$(PEEK(
HIMEM+(symnr-32)*8+n+1),8):NEXT           (403E)
870 x1=10:x2=25:y1=3:y2=5:GOSUB 61
0:PRINT:PRINT CHR$(24)"DEFINE SYMB
OL"CHR$(24)                               (4A8A)
880 x1=2:x2=33:y1=7:y2=16:GOSUB 61
0:GOSUB 1240                              (30DD)
890 x1=3:x2=10:y1=8:y2=15:GOSUB 61
0:WINDOW 3,10,8,16                        (3601)
900 ORIGIN 32,160:x=1:y=1:vx=1:vy=
113:rsx=145:rsy=113                       (3E60)
910 rsy=rsy+16                            (159B)
920 FOR i=1 TO 8                          (0E87)
930 FOR n=1 TO 8                          (0EDE)
940 a$=MID$(sy$(i-1),n,1)                 (217D)
950 IF a$="1"THEN PEN 2:LOCATE n,i
:PRINT CHR$(143);:PLOT rsx+2*n,rsy
-2*i,2:PEN 1                              (47BC)
960 NEXT                                  (0645)
970 NEXT                                  (0659)
980 PEN 2:rsx=147:rsy=111:rsy=rsy+
16                                        (2CEE)
990 farb=TEST(vx,vy)                      (1C58)
1000 LOCATE x,y                           (0F21)
1010 PEN 1:PRINT CHR$(143);:PLOT r
sx,rsy,farb                               (270F)
1020 WHILE INKEY$="":WEND                 (0DBA)
1030 LOCATE x,y:PEN farb:PRINT CHR
$(143);                                   (2182)
1040 IF NOT INKEY(1)OR NOT INKEY(7
5)THEN x=x+1:vx=vx+16:rsx=rsx+2:IF
 x>8 THEN x=8:vx=113:rsx=161              (61A8)
1050 IF NOT INKEY(8)OR NOT INKEY(7
4)THEN x=x-1:vx=vx-16:rsx=rsx-2:IF
 x<1 THEN x=1:vx=1:rsx=147                (6092)
1060 IF NOT INKEY(0)OR NOT INKEY(7
2)THEN y=y-1:vy=vy+16:rsy=rsy+2:IF
 y<1 THEN y=1:vy=113:rsy=127              (6144)
1070 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN y=y+1:vy=vy-16:rsy=rsy-2:IF
 y>8 THEN y=8:vy=1:rsy=113                (604A)
1080 IF NOT INKEY(18)OR NOT INKEY(
76)THEN farb=2:GOTO 1120                  (2451)
1090 IF NOT INKEY(66)THEN 1130            (11F3)
1100 IF NOT INKEY(79)THEN farb=0:G
OTO 1120                                  (1C92)
1110 farb=TEST(vx,vy)                     (1C48)
1120 GOTO 1000                            (0904)
1130 LOCATE x,y:PEN farb:PRINT CHR
$(143):PLOT rsx,rsy,farb                  (37DB)
1140 v=0                                  (0B80)
1150 FOR n=8 TO 1 STEP-1                  (1198)
1160 v=v+1:a$=""                          (182A)
1170 FOR a=8 TO 1 STEP-1                  (11A6)
1180 farb=TEST((16*a-2),(16*n-1))         (2867)
1190 IF farb=2 THEN a$="1"+a$ELSE
a$="0"+a$                                 (2CCD)
1200 NEXT                                 (0626)
1210 POKE HIMEM+(symnr-32)*8+9-n,V
AL("&X"+a$)                               (2D23)
1220 NEXT                                 (064E)
1230 PEN 1:RETURN                         (0937)
1240 LOCATE 19,2:PRINT CHR$(240)          (12C5)
1250 LOCATE 18,3:PRINT CHR$(242)          (12DD)
1260 LOCATE 20,3:PRINT CHR$(243)          (1218)
1270 LOCATE 19,4:PRINT CHR$(241)          (1210)
1280 LOCATE 11,6:PRINT CHR$(24)"EN
TER"CHR$(24)" : SET POINT"                (2D9E)
1290 LOCATE 11,7:PRINT CHR$(24)"ES
C"CHR$(24)"  : EXIT                       (266C)
1300 LOCATE 11,8:PRINT CHR$(24)"DE
L"CHR$(24)"  : DELETE POINT               (2EEE)
1310 RETURN                               (0668)
1390 'UNTERPROGRAMM SAVE BIN              (1D9A)
1400 '                                    (0700)
1410 x1=9:x2=20:y1=13:y2=17:GOSUB
610:PRINT:PRINT" NUMBER:":PRINT:IN
PUT"  ",num                               (49B2)
1420 IF num<1 OR num>27 THEN RETUR
N                                         (1AD5)
1430 IF num<>27 THEN nam$="g3"+CHR
$(64+num)ELSE nam$="g3pat"                (3988)
1440 SAVE nam$+".bin",b,HIMEM+1,&3
00                                        (21C2)
1450 RETURN                               (067F)
1530 'UNTERPROGRAMM LOAD BIN              (1DE2)
1540 '                                    (0719)
1550 x1=9:x2=20:y1=13:y2=17:GOSUB
610:PRINT:PRINT" NUMBER:":PRINT:IN
PUT"  ",num                               (49CA)
1560 IF num<1 OR num>27 THEN RETUR
N                                         (1AEF)
1570 IF num<>27 THEN nam$="g3"+CHR
$(64+num)ELSE nam$="g3pat"                (39A0)
1580 LOAD nam$+".bin",HIMEM+1             (1870)
1590 RETURN                               (0699)
1600 '                                    (0791)
1670 'UNTERPROGRAMM DELETE                (1B53)
1680 '                                    (0730)
1690 POKE &B295,255                       (0C29)
1700 SYMBOL AFTER 256                     (0AD6)
1710 SYMBOL AFTER 32                      (099C)
1720 POKE &B295,0                         (0B43)
1730 RETURN                               (06B0)
```




Haben Sie Probleme? CPC Welt beantwortet Ihre Fragen Postfach 1161 D8044 Lohhof

CHECKER- der Eingabechecksummer für alle CPCs

Von Ihnen lange gefordert, ist er nun endlich da, der Checksummer für alle CPCs. Die nähere Besprechung dieses Eingabe-Prüfprogrammes erfolgt im Rahmen der Artikelreihe über die Programmierung in Maschinensprache. Hier nun nur die Bedienungsanleitung:

Tippen Sie das Programm sorgfältig ab, denn trotz der Prüfsummen am Ende jeder DATA-Zeile können Statementvertauschungen nicht erkannt werden. Nach dem Abtippen gleich abspeichern, damit bei einem eventuellen Eingabefehler nicht alles neu eingegeben werden muß!

Nach einem Programmlauf stehen zwei RSX-Befehle zur Verfügung: EIN und AUS. RSX-Befehle beginnen mit einem senkrechten Strich (Taste neben P mit Shift eingeben), und 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 Hexademizalzahl, die in eckige Klammern eingeschlossen ist. Diese Prüfsumme muß mit der im einzugebenden Listing enthaltenen Summe (am Ende jeder Basiczeile) übereinstimmen. Ist dies nicht der Fall, dann stimmt irgendetwas nicht! Geben Sie die Zeile neu ein (kann auch mit Edit Zeilennummer oder mittels der COPY-Cursor-Methode geschehen!).

Da unser Checker -- im Gegensatz zu anderen Checksummern -- 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 abgetippte Programm starten, denn es könnte ja sein, daß das Programm dort, wo unser Checker seinen Platz hat, selbst aktiv wird oder eigene RSX-Befehle mit gleicher Aufrufsbefehlsfolge hat und es deshalb Fehler gibt!

Da die Firmware der drei CPC-Typen leider unterschiedlich ist, verhält sich unser "Eingabe-Prüfer" beim Befehl AUTO unterschiedlich. Dieser Befehl kann beim CPC 464 nicht benutzt wwerden, da dann keine Prüfsumme ausgegeben wird. Beim CPC 664 und 6128 ist der Einsatz dieses Befehles möglich.

Viel Glück mit dem Prüfprogramm! *LM* □

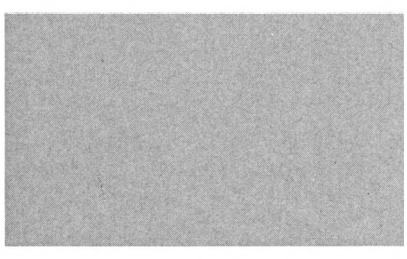
```
100 MEMORY &A2FF:'CHECKI.HEX
110 a=&A300:e=&A3F6:zb=1000:e=e+1
120 PRINT"Moment bitte...":FOR i=a TO e:
READ d$:IF LEFT$(d$,1)="&"THEN flag=1
130 IF(flag AND p<>VAL(d$))THEN PRINT"Fe
hler in Zeile "zb+1:END
140 IF(flag AND i=e)THEN 180
150 IF flag THEN i=i-1:zb=zb+1:p=0:d$=""
:flag=0:GOTO 170
160 POKE i,VAL("&"+d$):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:POKE &A39D,&AA:P
OKE &A39E,&E7:POKE &A3C0,&2
230 POKE &A3C1,&AC:POKE &A3CF,&5B:POKE &
A3E0,&5B:POKE &A3F2,&2:POKE &A3F3,&AC:PO
KE &A3EF,&8A:GOTO 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
```

Ein Wort an die Leser der CPC-WELT

Leser als Autoren

Immer wieder erhält die Programmredaktion der CPC-WELT Programme zugesandt, die von Ihnen, unseren Lesern, geschrieben wurden, und die Sie so gut finden, daß Sie sie für einen Abdruck geeignet halten. Diese Mitarbeit der Leser freut uns und nützt anderen Lesern. Bevor Sie jedoch Programme einsenden, sollten Sie die redaktionellen Kriterien für einen Abdruck beachten. Sie ersparen uns vergebliche Arbeit und sich selbst Frust und unnötige Wartezeiten.

Wenn Sie ein eigenes Programm geschrieben haben und meinen, daß es gut und für andere Leser interessant genug ist, dann schicken Sie es uns. Jedoch sollten Sie zunächst folgende Mindestanforderungen beachten:



1. Welche Programme eignen sich zum Abdruck?

Programme, die so oder ähnlich bereits von uns gedruckt wurden, haben nur eine geringe Chance, angenommen zu werden. Wenn Ihr Programm jedoch etwas Neues kann, dann her damit. Vokabeltrainer und Mindmaster haben wir bereits mehr als genug. Große Chancen, abgedruckt zu werden, haben dagegen Utilities, auch wenn sie nur kurze Routinen darstellen. **Nicht die Länge eines Programms ist entscheidend, sondern Qualität und Originalität.** Für gute Anwenderprogramme haben wir ebenfalls Verwendung.

2. Genaue Programmbeschreibung

Sie erhöhen Ihre Chancen, wenn Sie zum Programm eine gute und **ausführliche Beschreibung** beilegen, entweder als Text oder, noch besser, als **Textfile,** das Sie mit einem Textverarbeitungsprogramm wie Tasword oder WordStar angelegt haben.
Eine Kurzbeschreibung mit den Worten „Das Programm erklärt

sich von selbst" ist auf keinen Fall ausreichend und kann zur Ablehnung in der Redaktion führen.
Neben der Beschreibung der Leistungsfähigkeit ist auch eine kurze **Erklärung von Feinheiten in der Programmierung** wünschenswert. Denn Abtipper von Listings wollen ja in erster Linie etwas lernen von denen, die sich mit der Programmierung schon besser auskennen.
Die **Angabe der** notwendigen **Peripherie** ist unerläßlich, so also, ob das Programm eine Floppy benötigt, ob ein Drucker gebraucht wird, auf welchen CPC-Typen es läuft. Wenn Sie nicht alle CPC-Typen haben, genügt der Hinweis, auf welchem Rechner Sie das Programm geschrieben haben. Wir testen es auf allen Typen durch.

3. Testen Sie Ihr Programm

Ein guter **Programmautor testet** sein Programm **vollständig durch.** Wenn wir bereits beim ersten Probelauf auf SYNTAX ERROR oder SUBSCRIPT OUT OF RANGE stoßen, wissen wir sofort, daß Sie das nicht getan haben. Manchmal haben wir den Eindruck, daß uns Programme eingesandt werden mit dem Hintergedanken, daß unsere Programmredaktion die Fehler schon finden und ausbessern werde. Dies ist vielleicht noch bei der ersten oder zweiten Fehlermeldung der Fall. Kommen aber noch weitere, müssen wir den Test abbrechen und das Programm an Sie zurücksenden.

4. Verwechslungsgefahr vermeiden

Ihre **Einsendungen** müssen deutlich Ihren **vollständigen Absender** tragen. Am besten, Sie verwenden dazu den Einsendebogen, der in jedem Heft abgedruckt ist. Sinnvoll ist auch eine Beschriftung des oder der Datenträger (Diskette oder Kassette).

5. Sendung ausreichend frankieren

Frankieren Sie Ihre Sendung **richtig.** Legen Sie in jedem Fall ausreichend **Rückporto** bei. Nicht ausreichend frankierte Sendungen oder solche ohne Rückporto haben bei der redaktionellen Bearbeitung keine Chance.

6. Keine Steuerzeichen

Seit Heft 1/88 der CPC-WELT drucken wir alle Listings mit **Checksummer.** Deshalb können wir keine Listings abdrucken, die Steuerzeichen enthalten.
Alle Programme werden vor dem Ausdruck von uns so behandelt, daß sie keine überflüssigen Leerzeichen enthalten. Dieses Programm verträgt ebenfalls keine Steuerzeichen. Also bitte durch die entsprechenden chr**$**(..)-Befehle ersetzen.

7. Keine Zeilen schinden

Blasen Sie Ihr Programm nicht künstlich auf, um das erwartete Honorar zu erhöhen. Sie vermindern damit Ihre Chancen, daß es überhaupt abgedruckt wird.
Das höchste Seitenhonorar erhalten **komprimiert geschriebene BASIC-Programme** und solche, die in wesentlichen Teilen in **Maschinensprache** geschrieben wurden.

8. Der Abdruck Ihres Programms/Beitrags

Wenn Sie von uns die Zusage zum Abdruck erhalten haben, ist dies **für beide Seiten bindend.** Sie dürfen Ihr Programm also nicht weiter anbieten. Denken Sie aber bitte daran, daß wir ein gewisses Polster an guten Programmen vorrätig haben. Eine Veröffentlichung gleich im nächsten Heft ist unwahrscheinlich. Ist Ihr Programm jedoch schlichtweg exzellent, können wir es eventuell vorziehen. Der Regelfall ist aber eine mehrmonatige Wartezeit.

9. Garantie Ihrer Urheberrechte/Honorare

Bei der Einsendung Ihres Programmes garantieren Sie mit Ihrer Unterschrift rechtsverbindlich, daß Sie das Programm **selbst entwickelt** haben und der **alleinige Inhaber der Urheberrechte** sind. Bedenken Sie bitte, daß auch die Programmredaktion ständig in- und ausländische Fachmagazine aufmerksam liest und „umgestaltete" Programme schnell erkennt. Mit unserer schriftlichen Zusage, Ihr Programm in der CPC-WELT abzudrucken, geht Ihr Copyright an den Verlag der CPC-WELT über, der es wirtschaftlich verwerten kann. Dafür erhalten Sie ein **Honorar bis zu 1000 Mark.** Am einfachsten ist es, Sie benutzen den erwähnten Vordruck. Sie finden ihn in jeder Ausgabe der CPC-WELT.

10. Auch Textbeiträge sind willkommen

Gute **eigene Texte** sollten nicht in der Schublade liegen bleiben. Wenn Sie an einem Bericht arbeiten, der Ihrer Meinung nach für die Leser interessant sein könnte, dann halten Sie mit uns Rücksprache.
Wir werden klären, ob er thematisch ins Heft paßt. Schreiben Sie ihn am besten auf einem Datenträger im ASCII-Format. Sie sollten dazu ein Textverarbeitungsprogramm wie Tasword, WordStar oder den CP/M-Editor ED.COM benutzen.
Entgültig entscheiden, ob Ihr Beitrag gedruckt wird, können wir erst dann, wenn uns der Text vorliegt und redaktionell geprüft wurde. Wenn Sie passende und **reprofähige Fotos, Zeichnungen** oder **Skizzen** beifügen, können Sie sich gutes Geld hinzuverdienen, indem Sie andere Leser an Ihrem Wissen teilhaben lassen.

Ihre Redaktion
CPC-WELT □

Kleine Hilfe, ganz groß

DISKETTEN STETS GRIFFBEREIT?

Ein nicht ganz ernstzunehmender Testbericht

Ein Hersteller von Computer-Zubehör ließ uns ein praktisches Gerät zukommen: Die Plonker-Box. Sie sorgt endlich auch auf unseren Redaktions-Schreibtischen für Ordnung.

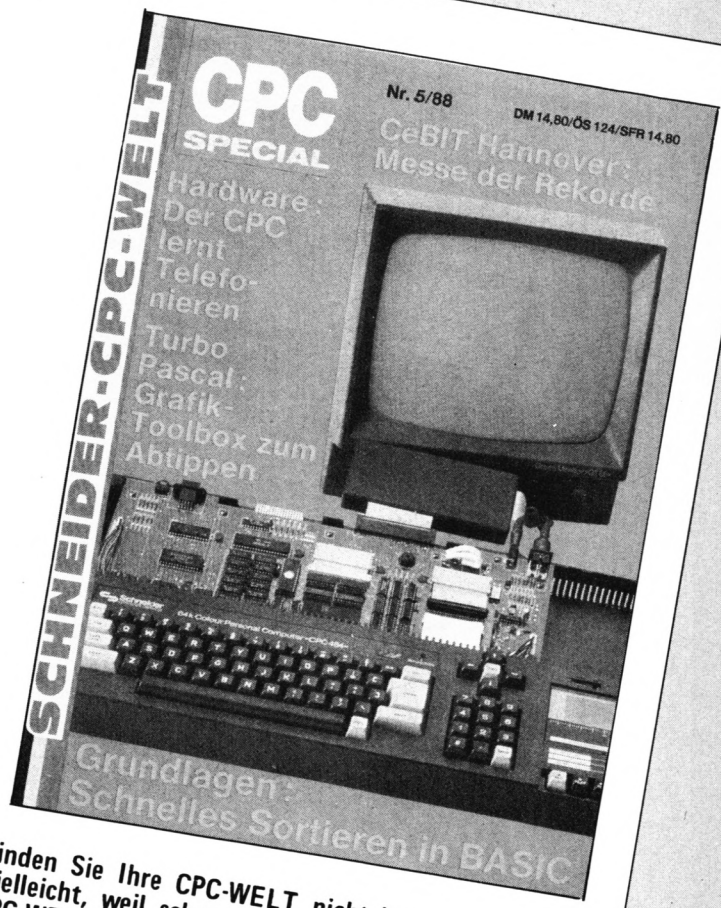
Der oft in den Medien gezeigte aufgeräumte Computer-Arbeitsplatz entspricht wohl selten der Realität. In der Regel häufen sich auf dem Schreibtisch – entsprechende Arbeitswut vorausgesetzt – eine Vielzahl von Papieren gut gemischt mit einer Anzahl von Disketten. Da ist es schnell geschehen: Der Anwender zieht unter einem Stapel eine Diskette hervor und erwischt diese gerade an ihrer griffigsten Stelle, der Lese-Schreiböffnung."
So beginnt der Text der Presse-Mitteilung, in der für die Plonker-Box Reklame gemacht wird. Das Testexemplar macht auf uns einen sehr guten Eindruck. Die hervorragende Verarbeitung aus solidem, modernem Plastikmaterial ist ebenso ein Pluspunkt wie die stabile Befestigungs-Vorrichtung aus einseitigem Klebeband. Alles in allem repräsentiert der Plonker offenbar den neuesten Stand der Produktionstechnik.
Negativ zu bewerten ist die farbliche Gestaltung in dezentem Schwarz. Sie stellt am beige-grauen Gehäuse unseres Textcomputers, einem Commodore PC-10, einen unangenehmen Kontrast dar.
Hier sollte sich der Hersteller das Design dieses Gerätes noch einmal durch den Kopf gehen lassen, wobei auch moderne Aspekte der Farbpsychologie berücksichtigt werden sollten. So wirkt die Farbe Rot unheimlich anspornend bei eintöniger Büro-Arbeit; grün hingegen hat einen beruhigenden Einfluß und sollte daher bevorzugt in der Chef-Etage zum Einsatz kommen.

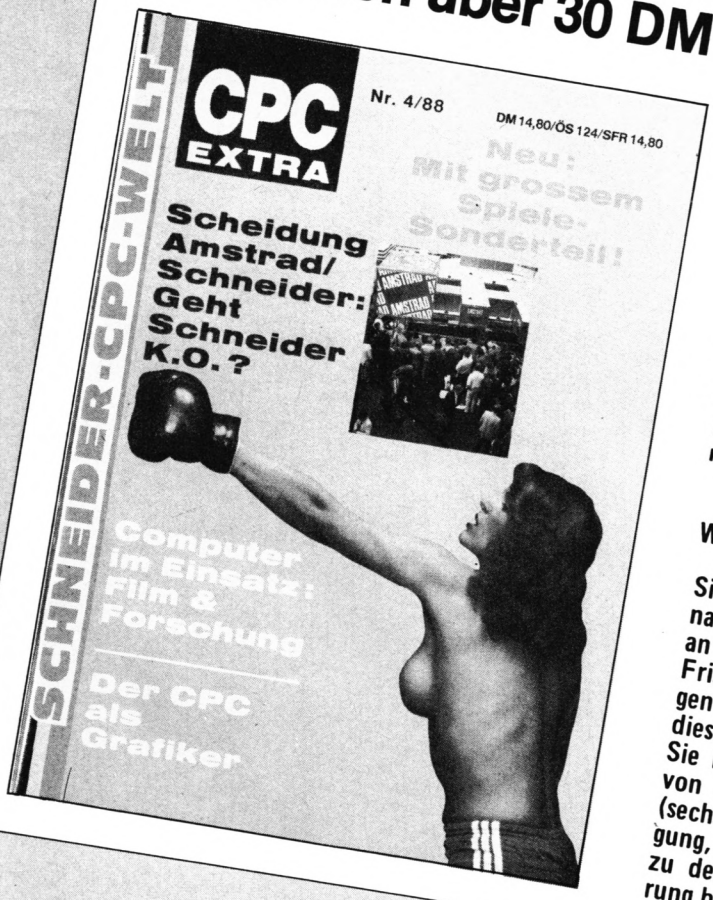
Lesen Sie bitte weiter auf Seite 83

Jetzt schlägt's dreizehn!

CPC-WELT kommt direkt zu Ihnen ins Haus! 7x pro Jahr! Für nur 70 DM!

Sie sparen über 30 DM!





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 liefern Sie im Abonnement mit sechs plus einer Ausgabe für ganze 70 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). Sechsmal. 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! Zur Wahrung der Frist genügt die rechtzeitige Absendung. Bitte, bestätigen Sie durch Ihre zweite Unterschrift, daß Sie von diesem Widerspruchsrecht Kenntnis genommen haben. Sie können dieses Abonnement jeweils mit einer Frist von einem Monat zum Ende des Bezugszeitraumes (sechs Hefte) kündigen. Unterlassen Sie diese Kündigung, wird die Belieferung mit weiteren sechs 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 sechs (+ 1) 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)

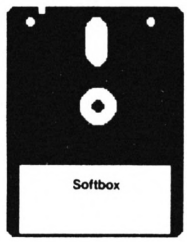
Unterschrift _____
Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____ **SPI/9**

CPC-WELT
ABO-SERVICE SPI/9
POSTFACH 1161
D-8044
UNTERSCHLEISSHEIM

PROGRAMMSERVICE

Hiermit bestelle ich in Kenntnis Ihrer Verkaufsbedingungen die Listings dieses Heftes auf
☐ Diskette mit Grafik-Editor CPC 464 (60 DM)
☐ Diskette mit übrigen Listings des Heftes (40 DM)
☐ Beide zusammen (90 DM)



Ich zahle (Zutreffendes bitte ankreuzen!):
per beigefügtem Scheck ☐ **Schein** ☐
gegen Bankabbuchung am Versandtag ☐

Meine Bank (mit Ortsname) _____
Meine Kontonummer _____
Meine Bankleitzahl _____

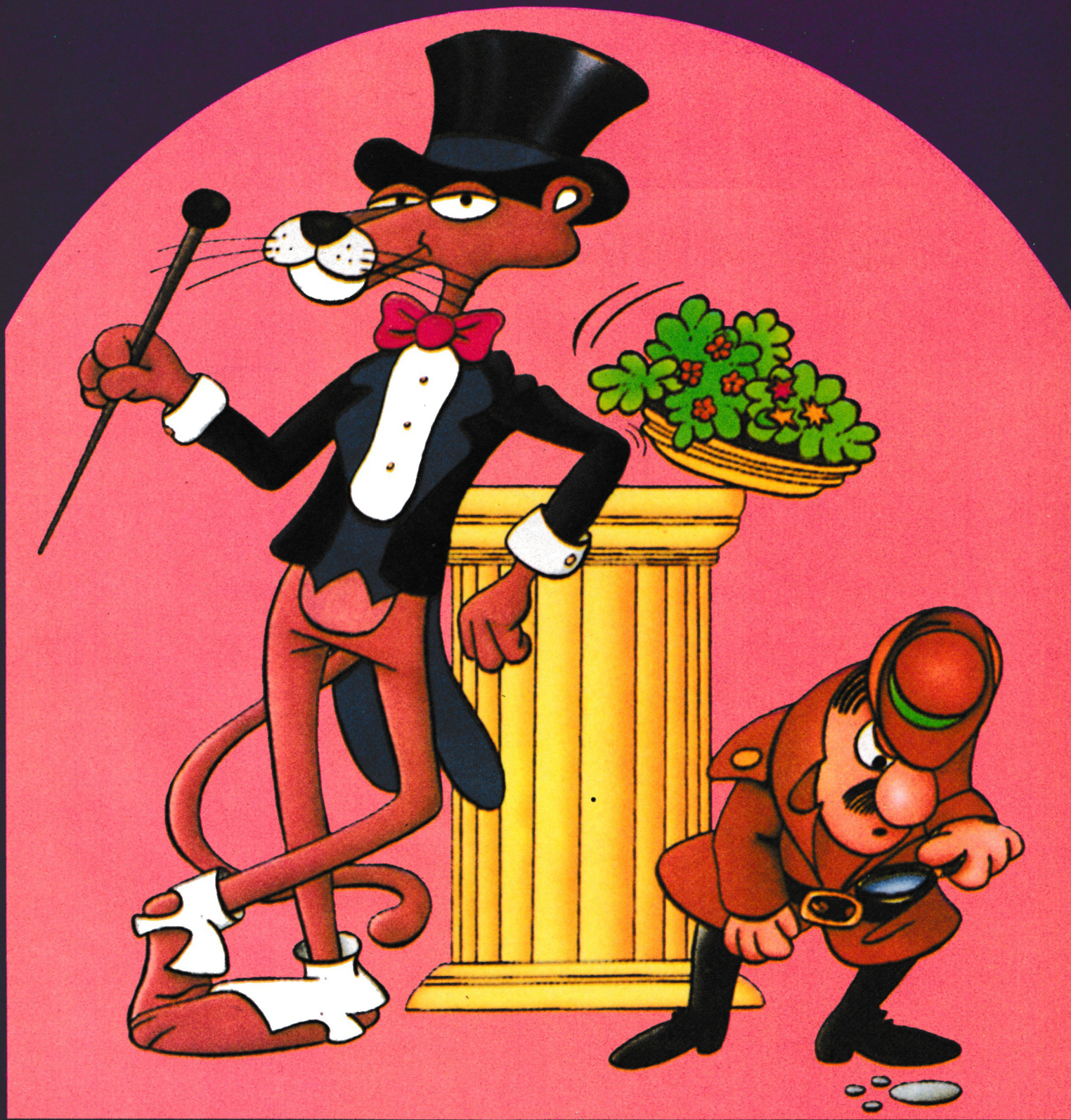
(steht auf jedem Bankauszug)
Nachname _____
PLZ/Ort _____ **SPI/9**
Name _____
Str./Nr. _____

Verkaufsbedingungen: Lieferung nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme. Umtausch bei Nichtfunktionieren.

Unterschrift _____
Bitte ausschneiden und einsenden an

CPC-WELT
SOFTWARE-SERVICE SPI/9
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

LOAD&RUN



Das Spiele-Magazin

Hallo, liebe Leser,

Infogrames scheint umgestellt zu haben. Spiele, wie wir sie bisher von der Firma gewohnt waren, werden kaum noch produziert. Dafür hat man sich jetzt auf die Umsetzung von erfolgreichen Comic-Bänden bekannter Autoren spezialisiert.
Nach der gelungenen Bob-Moran-Trilogie, die wir für LOAD & RUN 5/88 getestet haben, erscheint nun ein neues Programm auf dem deutschen Softwaremarkt. Jeder kennt Bobo, den Ausbrecherkönig der Haftanstalt Riegelfest, der bereits seit 17 Jahren unsere Lachmuskeln strapaziert. Zeichner Paul Deliege war mit seiner Comicfigur damals der Durchbruch gelungen. Anfangs fanden Bobos Abenteuer lediglich in Rolf Kaukas Fix und Foxi Platz. Mittlerweile gibt es viele Sonderbände mit Bobo und seinen Kumpanen vom Zuchthaus Riegelfest.
Infogrames ist mit seinem neuesten Werk Bobo ein Meisterstück gelungen. Wir sind gespannt, wann das Spiel in den internationalen Software-Hitlisten auftauchen wird. Einen ausführlichen Testbericht lesen Sie in dieser Ausgabe.

Die Players Pages sind ja bereits ein fester Bestandteil von LOAD & RUN. Da diese Spielehilfe-Seiten bei unseren Lesern großen Anklang gefunden haben, wie die vielen Zuschriften beweisen, werden wir sie auch in Zukunft beibehalten. Auch in diesem Monat kommen die Schneider/Amstrad-CPC-Besitzer sehr gut weg. Eine lange Poke-Liste für den CPC soll den bösen Sprites wieder einmal das Leben schwer machen.
An dieser Stelle möchte ich Sie nochmals aufrufen, sich an der Gestaltung der Players Pages aktiv zu beteiligen. Auf gut deutsch: Schicken Sie uns Pokes, Karten, Tips, Tricks, Lösungen oder Fragen zu jedem x-beliebigen Spiel für jedes x-beliebige Computersystem. Als kleinen Ansporn verlosen wir jeden Monat ein Originalspiel, deshalb Systemangabe nicht vergessen! Ich hoffe auf große Resonanz von Ihrer Seite.

In diesem Sinne wünsche ich Ihnen im Namen der Redaktion viel Spaß mit dem neuen LOAD & RUN.

Ihr
Thomas Bosch

Inhalt

Schmetterling durch CPC
Volleyball-Simulator für den Sportspielefreak **Seite III**

Pink Panther
Der Rosarote Panther, nach Kino und Fernsehen auch als Geschicklichkeitsspiel **Seite IV**

Bobo
Der König der Ausbrecher, in seinem Element, dem Gefängnis Riegelfest **Seite V**

Vier auf einen Streich
Deep Space — Hacker II — Brataccs — LCP **Seite VI**

Troll
Ein Jump- and Run-Game für geschickte Spieler **Seite VII**

Top Fuel Eliminator
Ein neues Autorennspiel, das wirklich „top" ist **Seite VIII**

Arkanoid II
Die Breakout-Welle im erwarteten Aufschwung **Seite IX**

Rimrunner
Fleißiges Ballern ist bei diesem Game angesagt **Seite X**

10 Great Games II
Software-Sampler für den Commodore 64, für jeden Geschmack etwas **Seite XI**

Games für kleine Geldbeutel
On Cue — The Official FA Cup — Impossible Mission — Bruce Lee — The Eidlon **Seite XII**

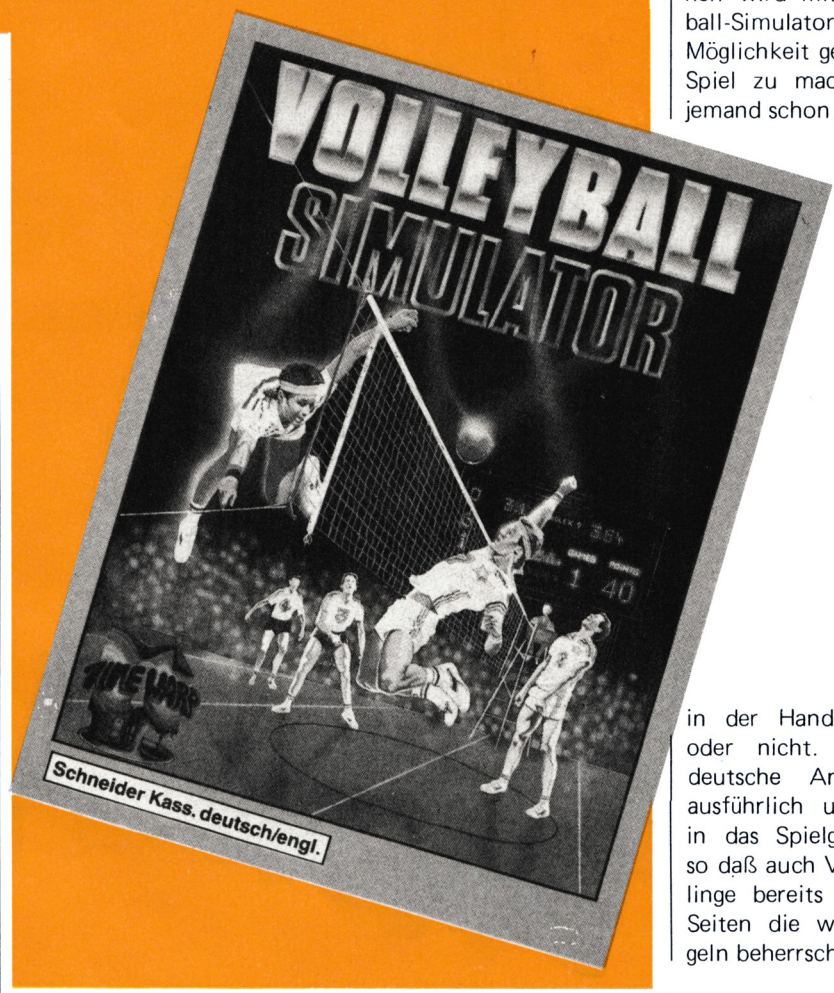
Kurzberichte
Trauma — Bob Moran — Witzball — Iznogud — Patton vs. Rommel — Starfleet **Seite XIV**

Computer Classics
Dynamite Dan — Zynaps — Into The Eagle Nest — Aliens — The Pumpkin Strikes Back **Seite XV**

Players' Pages
Tips & Tricks für aktive Joystickartisten **Seite XVI**

Schmetterball durch CPC

Auf dem Softwaremarkt gibt es die unterschiedlichsten Sportsimulationen. Vom Fußball über Handball bis zum Eishockey kann sich der Sportfreak an seinem Computer zumindest theoretisch austoben. Eine Kategorie fehlte bisher: Volleyball.



Uns stand für den Test die Kassettenversion für den Schneider CPC zur Verfügung. Die etwas ungewöhnlich lange Ladezeit wird durch mehrere Bilder und Grafik-Effekte überbrückt. Danach steht ein Menü mit vier Optionen zur Verfügung, welche mit den Cursor-Tasten angewählt werden. Hier können zunächst einige Voreinstellungen wie Verändern der Bildschirmfarben (wichtig für Grünseher!) oder Auswahl der Steuerung tätigen. Auch ein abgespeicherter Spielstand kann geladen werden.

Bevor Sie die Option Play anwählen, müssen Sie sich entscheiden, welche Rolle der Computer im Spiel übernehmen soll.

Bis zu 14 Spieler, also zwölf Feldspieler und zwei Trainer, können am Volleyball-Turnier teilnehmen. Die größere Zahl der Spieler erhöht den Spaß.

Für Einzelspieler gibt es mehrere Möglichkeiten. Der Computer kann die volle Kontrolle über die Mannschaft übernehmen. In diesem Fall fungiert der Spieler lediglich als Trainer und muß im geeigneten Moment die Taktik wechseln.

Eigene Taktiken können mit dem integrierten Tatik-Editor entworfen werden. Der Spieler kann seine Mannschaft selbst steuern.

Sehr gut gelungen ist dem Autor die Animation der Spielfiguren und des Balles, welche sich zwar etwas langsam, dafür aber völlig flimmerfrei über den Bildschirm bewegen. Untermalt wird das ganze Spektakel von diversen Sound-Effekten wie dem Aufklatschen des Balles oder dem Pfeifen des Schiedsrichters.

Durch zahllose Varianten, Möglichkeiten und Optionen wird mit dem Volleyball-Simulator jedem die Möglichkeit gegeben, „sein" Spiel zu machen, egal, ob jemand schon mal einen Ball in der Hand gehalten hat oder nicht. Die 50seitige deutsche Anleitung führt ausführlich und informativ in das Spielgeschehen ein, so daß auch Volleyball-Neulinge bereits nach wenigen Seiten die wichtigsten Regeln beherrschen. ● *TB*

Titel:	**The Volleyball-Simulator**		
Getestet:	**Schneider CPC**		
Umsetzungen:	**C-64, C-128 (geplant: Amiga, Atari ST)**		
Im Test:	**Preis (DM):**	☒ **Joystick**	
☐ 💾		☒ **Tastatur**	
☒ 📼	**29,95**	☐ **Maus**	

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Pink Panther

Nach ihrem Superhit Vampire's Empire brachte das Edel-Softwarehaus Magic Bytes ein qualitativ kaum unterlegeneres Produkt auf den deutschen Markt. Das Geschicklichkeitsspiel „Pink Panther" bringt die Handlung der bekannten Fernsehserie in Ihren Rechner.





Die Handlung von Pink Panther kann nicht gerade als ehrenwert bezeichnet werden. Unser Held Paulchen ist mal wieder total abgebrannt und braucht dringend Geld, um sich über Wasser zu halten. Deshalb sucht er sich einen Job bei den Reichsten der Stadt als Hausdiener. Allerdings hat er dabei einen ganz gemeinen Hintergedanken: Tagsüber in den noblen Häusern wohnen und nachts dann eben diese von sämtlichen Wertgegenständen befreien.

Reiche Hausherren ausrauben

Fünf Hausherren würden schon ausreichen, um sich einen Urlaub im sonnigen Süden leisten zu können. Leider hat er dabei nicht damit ausgerechnet, daß die Eigentümer Schlafwandler sein könnten. Auch Inspektor Clouseau ist nicht untätig und versucht, Paulchen auf frischer Tat zu ertappen. Deshalb schnüffelt auch er nachts ständig durch die Häuser. Paulchen muß also aufpassen, daß er während seiner nächtlichen Raubzüge weder Clouseau trifft, noch daß ein Hausherr versehentlich gegen den Wandschrank schlafwandelt und dadurch unsanft bei seinem Spaziergang gestört wird. Gelingt es ihm, eine Villa komplett auszuräumen, kann er mit der nächsten weitermachen. Insgesamt gibt es fünf Villen auszuräubern, bis das Spiel gewonnen ist.

Ist das Spiel geladen, so stehen Sie vor der Wahl, ob Sie ins Kaufhaus gehen, um sich passend zu kleiden, oder sich aber gleich um einen Job bewerben, was für den Anfang wohl das Vernünftigere ist. Nach einem kurzen Gespräch mit Ihrem neuen Herrn können Sie dann gleich mit der Arbeit anfangen. Natürlich kommt in diesem Game nur das nächtliche – wesentlich interessantere – Leben Paulchens zur Geltung.. Sie sehen das Haus, dessen Etagen und Einrichtungen wie Möbel und Bilder von der Seite. Bewegen Sie den Joystick, so scrollen der Vorder- und Hintergrund fließend in alle Richtungen mit.

Die Grafiken aller Objekte sind hervorragend gezeichnet und stimmen in allen Details, auch in der Farbe, mit dem Trickfilm-Vorbild überein. Die Programmierer haben wirklich auf alles Rücksicht genommen. Sogar die Knöpfe an den Schränken und an Stiegengeländern sind hervorragend zu erkennen. Die Animation der Figuren läßt eigentlich nichts zu wünschen übrig. Sauber und fließend wandelt der schläfrige Hausherr durch die Flure, während ihm der Panther alle möglichen Gegenstände aus dem Weg räumt. Inspector Clouseau spaziert pixelgenau durch die Gegend.

Der Sound hingegen ist etwas zu kurz gekommen. Nur ein monotones Gedudel des berühmten Soundtracks wird rauf und runter gespielt. Daneben sind natürlich noch die üblichen Hintergrundgeräusche à la „Knacks" und „Rumms" zu hören. Wenigstens die Melodie hätten die Programmierer ansprechender machen können.

Auch mit der Motivation sieht es nicht ganz so gut aus. Viel Action und Hektik sind bei der ganzen Handlung nicht drin. Bereits nach einigen nächtlichen Streifzügen wird die Ausrauberei langweilig, da sich nichts Neues mehr bietet.

Fazit

Pink Panther kommt leider nur in Sachen Grafik an Vampire's Empire heran. In allen anderen Punkten, besonders aber im Spielspaß, schleicht Pink Panther weit hinter ihm her. Technisch ist das Game jedoch durch und durch gelungen. Freunde von ruhigeren Denk- und Geschicklichkeitsspielen sollten am rosaroten Panther nicht vorbeisehen. ● *mn*

Titel:	**Pink Panther**
Getestet:	**Amiga**
Umsetzungen:	**Atari ST, Schneider CPC, C-64**

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): k.A.

☒ Joystick ☐ Tastatur ☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Bobo

Jeder Comic-Fan kennt Bobo, den König der Ausbrecher. Seit nunmehr 17 Jahren versucht er, aus dem Gefängnis Riegelfest auszubrechen, weil er dort ständig schikaniert wird. Infogrames hat ihre „Collection Spirou" um die Abenteuer Bobos erweitert.



Sie werden für den Atari ST mit ROM-TOS auf zwei Disketten geliefert und bestehen aus sechs Einzelspielen, die Sie entweder der Reihe nach oder auch einzeln spielen können. Maximal sechs Spieler können sich an dem Spektakel beteiligen.

Das erste Spiel läuft in der Kantine des Gefängnisses ab. Bobo muß seinen hungrigen Kameraden Essen auftischen, sonst droht eine Meuterei. Schafft er es nicht, alle zufrieden zustellen, bekommt er den Napf aufgesetzt und das Spiel ist beendet.

In einer weiteren Szene muß Bobo einen großen Haufen Erdäpfel schälen. Gar nicht so einfach, da der Haufen ständig größer wird. Die Sache muß gründlich gemacht werden, denn schlecht geschälte Kartoffeln kommen wieder zurück. Dieses Spiel ist beendet, sobald Bobo von ungeschälten Kartoffeln überschwemmt wird.

Den Boden aufzuwischen ist Bobos verhaßteste Arbeit, denn seine Kumpane haben die schlechte Angewohnheit, genau da vorbeizugehen, wo unser Held den Putzlappen schwingt. Da der Boden feucht ist, hinterlassen sie Spuren und Bobo muß hinterherwischen, denn alles muß blitzblank sein, wenn der Herr Direktor zur Inspektion kommt.

Wenn die Gefängiswärter mal wieder streiken, versuchen natürlich die Insassen, auszubrechen. Diesmal springen sie vom Fenster herab auf ein Trampolin, das sie dann über die Mauer befördert. Bobo, intelligent wie er ist, hat die Aufgabe übernommen, das Trampolin von Fenster zu Fenster zu schieben. Verständlich, daß seine Kumpane nicht begeistert sind, wenn sie ins Leere springen.

Im nächsten Spiel ist Bobo endlich die Flucht gelungen. Er balanciert auf Stromkabeln herum und muß dabei von einem auf das andere springen, um den Spannungsblitzen auszuweichen, die seiner Gesundheit schaden können.

Nach einem erfolgreichen Tag legt sich ein professioneller Ausbrecher müde ins Bett und möchte schlafen. Leider schnarchen die Zellenkollegen so laut, daß dies unmöglich ist. Bobo bleibt nichts anderes übrig, als aufzustehen und jeden einzelnen zu beruhigen. Dabei darf er nicht über Nachttöpfe oder ähnliche Gegenstände stolpern.

Alle sechs Spiele sind grafisch sehr sauber und detailliert gestaltet. Die Bewegungen der Figuren erfolgen völlig fließend und ruckfrei. Zu jedem Spiel erklingt eine flotte Hintergrundmusik, die den Spieler anspornt. Jedes Spiel verfügt zusätzlich über einige nette Effekte. Es kann zum Beispiel passieren, daß beim Trampolinspringen ein Gefangener zu flach abspringt und, statt über die Mauer zu segeln, mit einem dumpfen „Boing" dagegen kracht, langsam runterrutscht, den Kopf schüttelt und erneut das Absprunggebäude betritt.



Fazit

Bobo ist ein Spiel, das einfach Spaß machen muß, vor allem dann, wenn mehrere Spieler gegeneinander antreten. Wettbewerbe wie beispielsweise ein Punktekampf können problemlos realisiert werden. Die gute Grafik und der gelungene dreistimmige Sound lassen die Spielmotivation enorm ansteigen. Wer Bobo noch nicht kennt, sollte sich sofort auf den Weg zum Computer-Fachhändler machen. ● *TB*

Titel:	**Bobo**
Getestet:	**Atari ST**
Umsetzungen:	**C-64, C-128**

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): 59,95

☒ Joystick ☒ Tastatur ☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Vier auf einen Streich

Eine recht günstige Anschaffungsgelegenheit für die ansonsten nicht billige Amiga-Software sind die mittlerweile beliebt gewordenen Game-Sampler. Mit ihnen erhält man mehrere Spiele, die zwar oft nicht mehr zu den Neuesten gehören, aber trotzdem gut sind, zum Preis von einem.

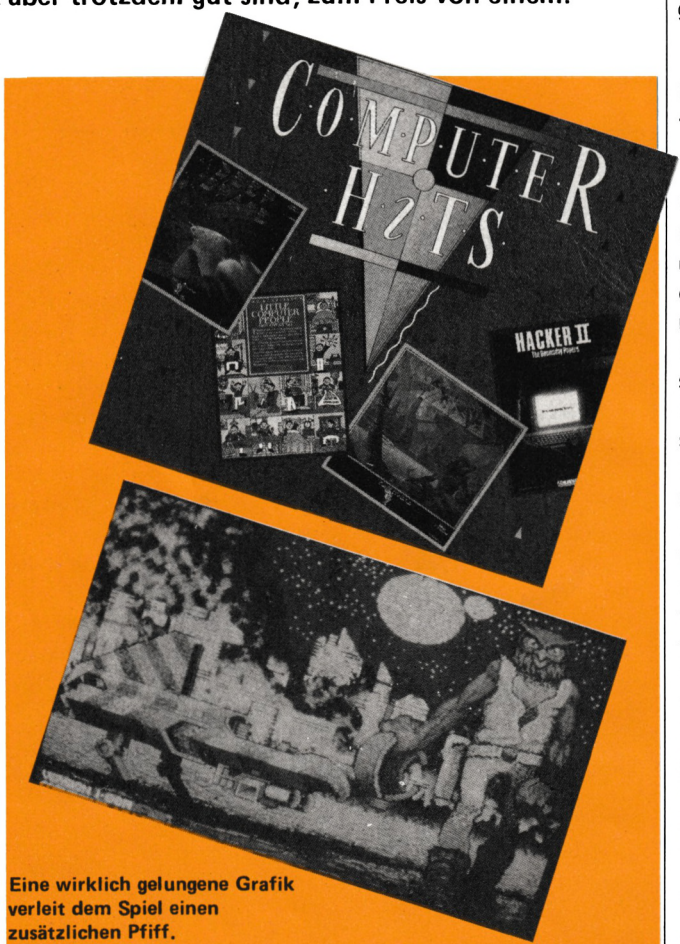
Ariolasoft vertreibt seit einiger Zeit in Deutschland einen neuen Sampler namens „Computer Hits".
Mit den vier Titeln handelt es sich um Deep Space, Hacker II, The Little Computer People und Brataccs.

Deep Space — Krawall im Weltraum

Bei Deep Space handelt es sich um ein Weltraum-Spiel mit dem Blickpunkt aus dem Cockpit eines Kampfraumschiffes. Wie bei vielen anderen Games dieser Sorte kommen alle erdenklichen Gegenstände und Objekte auf Sie zugeflogen. Darunter sind, wie sollte es auch anders sein, auch Asteroiden und feindliche Raumjäger, die Sie unbedingt abknallen sollten, da nämlich Sie sonst schnell vom Himmel gefegt werden. Das Spiel verfügt über einige Extrafunktionen, wie verstellbare Radarschirme und Schutzschilde, die recht hilfreich sein können. Im Großen und Ganzen ein ganz nettes Spiel für zwischendurch, das aber schnell eintönig wird.

Hacker II — The Doomsday Papers

Bei Hacker II sieht es da schon ganz anders aus. Hierbei handelt es sich um ein technisch sehr kompliziertes und forderndes Spiel, das Sie sicher länger vor dem Bildschirm festhalten wird.



Eine wirklich gelungene Grafik verleit dem Spiel einen zusätzlichen Pfiff.

Bei ihm müssen Sie per Datenfernübertragung einen Roboter in einem unterirdischen Bunkerkomplex in Sibirien umhersteuern und mit seiner Hilfe wichtige amerikanische Geheimdokumente finden. Die Wachen und Video-Überwachungskameras, die Sie übrigens manipulieren können, versuchen das natürlich zu vereiteln. Grafik und Sound liegen beim Durchschnitt.
Insgesamt ist Hacker II ein durchweg ansprechendes und interessantes Game.

Little Computer People

Wußten Sie schon, daß in Ihrem Rechner ein kleines Männchen wohnt? Und dieses Männchen wartet nur darauf, endlich den kennenzulernen, dessen Befehle es Tag für Tag millionenmal ausführt.
Sie werden sich wundern, wie bequem es sich unsere Little Computer People in der langweiligen IC-Landschaft gemacht hat. Ein schönes großes Haus mit allem drum und dran, inklusive Hund und Klavier, paßt leicht in ihren Computer.
Mit Hilfe eines komplizierten, aus Forschungsgründen generierten Programm ist es gelungen, dieses Haus auf dem Monitor sichtbar zu machen.
Little Computer People kann eigentlich nur bedingt als Spiel bezeichnet werden, da es hier nichts zu spielen gibt — nur zu beobachten.

Unterhaltung per Tastatur

Über die Tastatur können Sie sich mit dem Männchen unterhalten und es wird dementsprechend darauf reagieren.
Die Grafik ist gelungen und sehr niedlich ausgefallen. Ein „Spiel", mit dem Sie sich länger beschäftigen können, da offenbar noch niemand seinen privaten Mann genau kennt. Überraschenderweise tauchen immer wieder völlig neue Situation auf. Mein Computermännchen hat sich zum Beispiel immer strikt geweigert, ein Bad zu nehmen. Nachdem ich ihn etwas aggressiver dazu aufgefordert und sogar geschimpft habe, ist er murrend aber zügig in die Wanne gestiegen.

Brataccs — Gejagt vom ganzen Universum

Das letzte Game dieses Samplers bereitet bei seiner Beurteilung gemischte Gefühle. Obwohl die Idee gut und das Spiel an sich sauber programmiert ist, kann es nicht recht überzeugen.
Sie sind ein Flüchtling, der von Polizei und Mafia gejagt wird. Ihnen wird das schwere Verbrechen des „genetischen Betrugs" zur Last gelegt. Was immer das auch sei, jedenfalls haben Sie beschlossen, von der Erde zu verschwinden und sich auf einen kleinen Bergwerksasteroiden, Brataccs, tele-

portiert. Hier soll irgendwo der Beweis Ihrer Unschuld sein. Natürlich wissen Sie nicht im geringsten, wie der Beweis aussieht und müssen sich somit mit der Bevölkerung Brataccs' auseinandersetzen. Wie bei allen Wesen gibt es auch hier die unterschiedlichsten Typen, manche freundlich, manche feindlich gesinnt.
Als Spieler führen Sie Ihren Schützling per Tastatur oder Maus über den Bildschirm. Die Steuerung ist derart komplex und empfindlich, daß sie dem Anfänger einige Schwierigkeiten bereitet.
Die Grafik erreicht nicht den heutigen Standard. Vom Sound wollen wir gar nicht reden. Das wäre alles nicht so schlimm. Aber woran es Brataccs wirklich fehlt, ist Spielwitz. Bereits nach einigen Minuten wird das Spiel langweilig. Für Freaks, die sich gerne mit komplizierten Games beschäftigen ist es aber sicher gar nicht so schlecht.

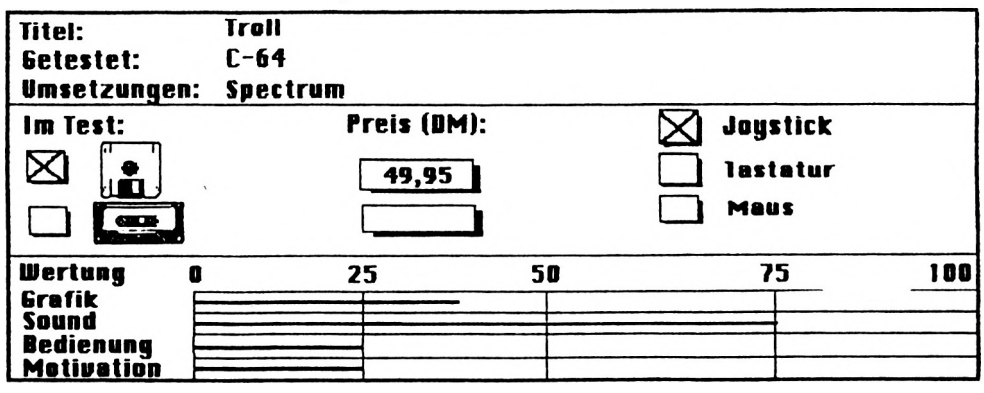
Fazit

Computer Hits enthält teilweise Spiele aus der Anfangszeit des Amiga, als er noch mindestens 4.000 Mark kostete. Deshalb nutzt keines der Games den Rechner vollkommen aus. Bedenkt man aber, daß Sie zum (anfangs etwas hoch erscheinenden) Preis von knapp achtzig Mark gleich vier Games bekommen, so wird sich mancher überlegen, ob er nicht zugreift. Zwei der Spiele sind unbedingt empfehlbar: Hacker II und Little Computer People. Die anderen haben einige Macken, gehören aber trotzdem nicht zur schlechtesten Sorte. ● *mn*

Troll

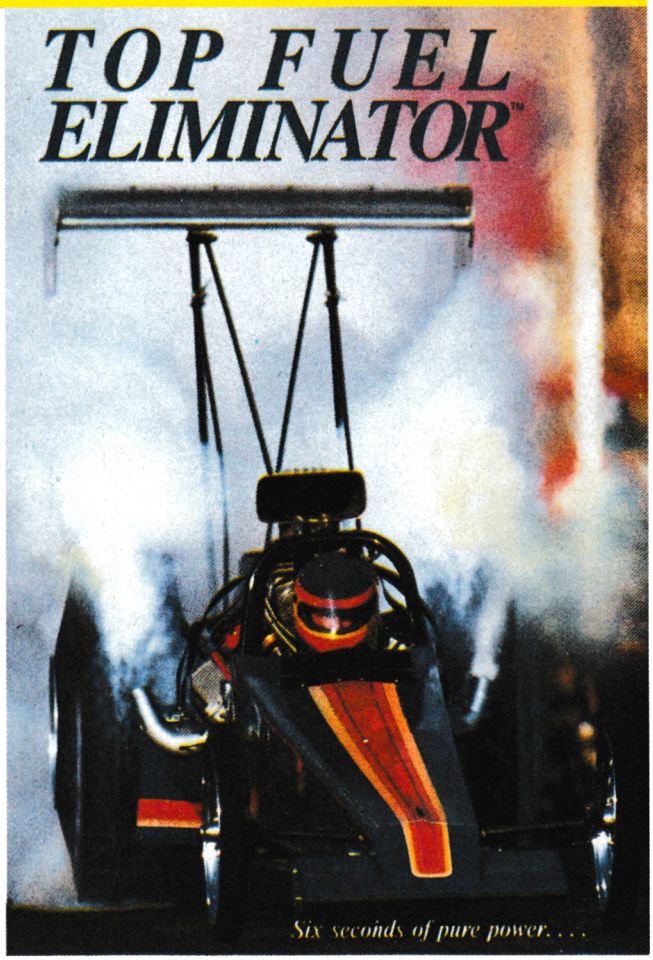
Jump- and Run-Spiele sind in. Um sie erfolgreich auf den Markt zu bringen, muß man sie nur neu verkleiden und sich immer wieder fantasiereiche Geschichten dazu einfallen lassen. Ein neues Produkt nach diesem Schema nennt sich kurz und bündig Troll und verdankt seine Existenz dem Hause Denton Designs.

Die Story: Die Hauptperson ist ein etwas zu groß geratener, aber dennoch recht liebenswerter Troll namens Humgruffin. Dieser ist durch einen Fluch in die Unterwelt von Narc geraten, ein unheimliches Land voller Kristallhöhlen, bevölkert von einem Stamm von Kobolden, die nichts als Böses im Sinne haben. Er weiß, daß seine einzige Chance, lebend aus dieser Welt zu fliehen, darin besteht, den Fluch umzudrehen. Nur durch das Sammeln von bestimmten Kristallen, die in sein Amulett passen, ist dies möglich.
Hiermit ist das Ziel von Troll eigentlich schon genannt: Wandern Sie durch zahlreiche Räume und sammeln Sie die Kristalle auf. Natürlich versuchen die Kobolde und allerlei anderes Ungetier, Sie an diesem Vorhaben zu hindern. Um Ihre Mission erfolgreich zu bestehen, sind Sie mit sehr eigentümlichen Waffen ausgerüstet. Um immer bereit zu sein, tragen Sie ständig eine gewisse Zahl von Löchern(!) in Ihren Taschen mit, um sie bei Bedarf einem Kobold vor die Füße zu werfen, der auf Nimmerwiedersehen darin verschwindet. Ein Kobold benötigt seinerseits ein Loch, um in den nächsten Raum zu kommen. Diese Art von Löchern trägt er aber nicht bei sich, sie müssen erst vom Oberzwerg Fjalar beschworen werden.
Die durch die Story des Spiels erweckten Hoffnungen werden jedoch durch seine Grafik enttäuscht. Auf dem Bildschirm präsentiert sich ein annähernd dreidimensionales Spielfeld, aufgebaut aus vielen einzelnen Plateaus. Die Figuren, also die Kobolde und der Troll, erscheinnen darauf fast unsichtbar klein. Hinzu kommt die zwar recht gelungene Farbgebung, die aber bei einer dermaßen klein geratenen Grafik das ganze noch unübersichtlicher macht. Die Animation erfolgt zwar recht schnell, ist aber trotzdem sehr ruckelig.
Da das Spielfeld sehr klein geraten ist, erweckt es den Anschein, als sei eine ganze Menge darauf los. Trotzdem fällt bei genauerem Betrachten auf, daß sich nicht mehr als acht Sprites auf dem Bildschirm befinden.
Lobenswert ist die Musikuntermalung. Während des Spielablaufs tönt ständig eine fetzige und hervorragend komponierte Melodie aus dem Lautsprecher. Dafür ist jedoch der Sound zu kurz gekommen, was allerdings kaum auffällt.
Die Steuerung macht das Game enorm schwierig. Leider wird sie unkorrekt und viel zu nervös abgefragt und umgesetzt. Dadurch ist es fast unmöglich, den Troll genau zu steuern. „Troll" ist dadurch fast unspielbar.
Das Interesse am Spiel verliert sich dadurch bereits nach kurzer Zeit. ● *mn*



Top Fuel Eliminator

Ein neues Autorennspiel reißt inzwischen niemanden mehr vom Hocker. So war ich auch relativ abgeneigt, ein solches Spiel zu testen. Was sollte da schon "top" sein? Meine Meinung mußte ich aber sehr schnell ändern . . .



Sie befinden sich in einem Fahrzeug mit über 2500 PS unter der Haube. Der Wagen wurde speziell für Sie gebaut, um Sie in kürzestmöglicher Zeit zu einem 400 Meter entfernten Ziel zu bringen.

2500 PS unter der Haube

Ihre Reflexe müssen blitzschnell sein, denn dies ist wie das Steuern einer Rakete, die auf dem Boden entlanggleitet. Top Fuel Eliminator ist die Simulation einer ganzen Dragster-Rennsaison. Die Saison besteht aus neun Rennereignissen und wird auf der Basis eines einheitlichen Punktesystems bewertet.

Sie müssen gegen acht der besten Rennfahrer der Welt antreten. Um ein Rennen zu gewinnen oder sogar nur, um ohne größeren Schaden ans Ziel zu kommen, müssen Sie auch auf die Wetterbedingungen und den Fahrbahnzustand achten und Ihren Dragster entsprechend darauf einstellen. Fast wie im richtigen Leben.

Top Fuel Eliminator bietet Ihnen eine Vielzahl von Optionen an, mit denen Sie Ihre Gewinnchancen beeinflussen und erhöhen können. Unter anderem können Sie Ihren Flitzer mit verschiedenen Reifen oder sonstigen Zusatzgeräten ausrüsten. Tanken nicht vergessen, sonst geht sowieso nichts. Sind alle Vorbereitungen getroffen, kann's losgehen. Das Steuern des Rennwagens ist wesentlich schwieriger, als ich zunächst angenommen hatte. Gerade als ich meinem Flitzer endlich unter Kontrolle gebracht habe, drängt mich ein Seitenwind an die Leitplanke. Also das Ganze nochmal von vorne. Diesmal klappts, mein Wagen passiert elegant

Übermut tut selten gut

die Ziellinie und kracht gegen die Zuschauertribüne. Aber macht nichts , passiert ist passiert. Auf zum Halb-Finale.

Im Laufe des Tages gelangte ich noch bis zur zweiten Rennstrecke, dann war mein Auto so kaputt, daß mich die Reparatur zu teuer gekommen wäre. Na, wenigstens durfte ich mich in der Highscore-Liste verewigen.

Fazit

Anfängern ist Top Fuel Eleminator nicht zu empfehlen, aber auch reine Action-Rennfreaks werden keine Freude an diesem Spiel haben. Es bietet so viele Optionen und Möglichkeiten, daß es ziemlich lange dauern kann, bis Sie sich damit zurechtfinden. Wenn Sie sich aber erstmal auskennen, werden Sie garantiert auf lange Zeit viel Spaß am Spiel haben. Und selbst wenn der Top Fuel Eliminator einmal in der Schublade landen sollte: An einem langen Winterabend wird er bestimmt wieder hervorgeholt werden, da er sich wohltuend von der Masse der Autorennspiele abhebt. Den Kauf bereut mit Sicherheit niemand. ● *TB*

Titel:	**Top Fuel Eliminator**
Getestet:	**C-64**
Umsetzungen:	**Apple II**

Im Test: ☒ ☐

Preis (DM): k.A.

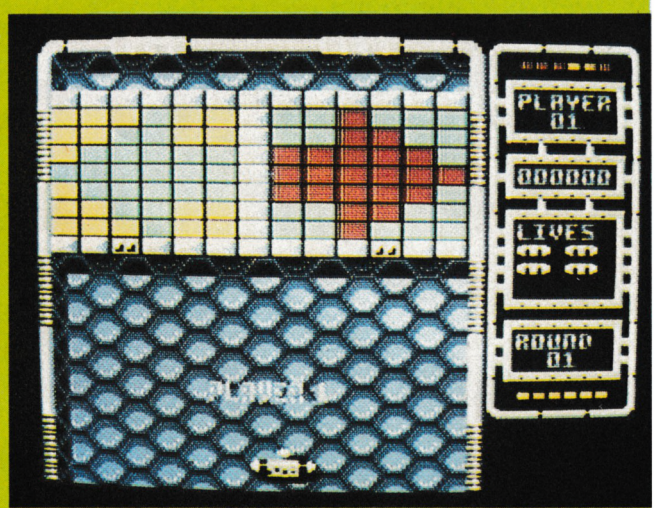
☒ Joystick
☒ Tastatur
☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Arkanoid II

Vor einem knappen Jahr erlebte die Breakout-Welle einen neuen Aufschwung. Der Grund hieß Arkanoid. Nun ist die Fortsetzung da: Arkanoid II.

Die Breakout-Welle reißt nicht ab. Es dürfte kein Computersystem mehr geben, für das nicht auch ein Breakout zu haben ist. Verständlich, denn schließlich macht diese Spielegattung sehr viel Spaß. Erstaunlich ist nur, daß sich Breakout-Versionen immer noch auf dem Softwaremarkt, der ja schon seit Jahren von außergewöhnlichen Spielideen verwöhnt wird, behaupten können. Besonders Imagine Software hat sich durch Arkanoid vor einem knappen Jahr einen guten Namen gemacht. Angenehm überrascht waren wir daher, als vor einigen Tagen der Nachfolger von Arkanoid in der Redaktion eintraf: Arkanoid II – Revenge of Doh. Das Spiel steht seinem Vorgänger in nichts nach. Lassen Sie sich deshalb warnen: Es macht süchtig.



Spannung und Faszination machen dieses Spiel zu einem heißbegehrten Game. Die Grafik unterstützt diese Aussage voll und ganz.

Breakout – ein Klassiker

Für Computerneulinge sei hier kurz das Breakout-Prinzip erläutert: Es gilt, mit einem Ball eine Mauer aus Einzelsteinen zu zerstören. Trifft der Ball auf einen Stein, so verschwindet dieser und der Spieler erhält einen Punkt gutgeschrieben. Der Ball trifft abwechselnd auf die Mauer und einen Schläger, der sich am unteren Bildschirmrand nach links und rechts bewegen lassen kann. Der Spieler übernimmt die Steuerung des Schlägers. Verfehlt er den Ball, wird ein Leben abgezogen.
Dasselbe Prinzip wird auch bei Arkanoid II angewandt.

Einmalige Features

Wie schon der Vorgänger weist diese Breakout-Variante einige Features auf, die das Spiel äußerst positiv von der Masse der Breakout-Versionen abheben. Einige der Mauerstücke lassen nämlich eine Art Pille zu Boden sinken, wenn sie vom Ball getroffen wurden. Nimmt der Spieler sie mit seinem Schläger auf, kann dies, je nach Farbe der Pille, unterschiedliche Wirkungen haben. Der Schläger kann verbreitert oder verschmälert werden, was die Treffsicherheit steigert beziehungsweise vermindert. Manchmal läßt sich der Schläger in eine Kanone umwandeln, mit welcher Sie die Steine einfach abschießen können. Das geht natürlich viel schneller. Ein besonderer Leckerbissen ist die Pille mit der Aufschrift „B". Durch sie können Sie nämlich problemlos ins nächste Level wechseln und gleichzeitig auch noch eine Menge Bonuspunkte einstreichen.

Viel Spaß durch viele Levels

Arkanoid II besitzt so viele Level, daß es ziemlich lange dauern kann, bis Sie alle gesehen haben. Damit sei wenigstens eine lang anhaltende Spielmotivation garantiert, werden Sie jetzt sicher denken. Viel besser – dieses Spiel macht süchtig. So unwahrscheinlich es auch klingen mag, aber wer einmal Arkanoid II gespielt hat, kommt so schnell nicht mehr vom Computer los. Wir in der Redaktion saßen stundenlang gebannt vor den Monitoren. Wenn uns vor einem Monat jemand gesagt hätte, daß uns ein Breakout so faszinieren würde, hätten wir ihn wohl mit einem Lächeln abgespeist. Nun, Arkanoid II – Revenge of Doh hat uns eines Besseren belehrt.

Fazit

Das Game kann jedem, ob Ballerfreak oder Arcade-Champ, uneingeschränkt empfohlen werden. Am besten, Sie reservieren sich gleich noch heute ein Exemplar bei Ihrem Computer-Fachhändler, sonst kann es leicht passieren, daß Ihnen ein anderer Arkanoid-Süchtiger zuvorkommt. ● *TB*

Titel: Arkanoid II
Getestet: C-64
Umsetzungen: Schneider CPC, Amiga

Im Test: ☒ Diskette ☐ Kassette

Preis (DM): 39.-

☒ Joystick ☒ Tastatur ☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

Rimrunner

Immer neue Katastrophen bahnen sich in den hintersten Ecken des Universums an. Immer neue machthungrige Herrscher greifen in das friedliche Alltagsleben idyllischer Planeten und Monde ein. So auch bei Rimrunner von Palace Software.



Folge 4.357 unserer Science-Fiction-Serie handelt von den armen Ameisigen, die ständig von den bösen Rickrakkern angegriffen werden. Um ihre Kolonien zu schützen, verwenden die Ameisigen Kraftschilder, mit denen sie große metallene Gegenstände abwehren können, so auch die Rickracker-Sturmtruppen.

Die Kraftgeneratoren, die die Schutzschilde für lange Zeit aufrechterhalten, brauchen ungeheure Energiemengen, die ihnen ständig zugeführt werden müssen. Dafür sind die Randpatrouillen verantwortlich, die am äußersten Rand einer Kolonie eingesetzt werden und mit ihren schnellen Reit-Reptilien die Generatoren aufladen und allerlei Kleingetier, das die Schilde durchbrochen hat, vernichten.

Fleißig Ballern ist angesagt

Dieser Patrouille gehören auch Sie an. Sie haben vier Minisaurier zur Verfügung, die Sie nach Bedarf verwenden können. Natürlich darf auch eine Schußwaffe nicht fehlen, mit der Sie unter den Aliens aufräumen. Wie sich herausgestellt hat, gibt es eine ganze Menge Eindringlinge.

Nach einem kurzen Ladevorgang erscheint ein mittelprächtiges Titelbild zum Spiel, das einen Ameisigen auf seinem Reitsaurier zeigt. Wollten wir das Game nach seinem Bild beurteilen, so fiele die Wertung nicht sehr gut aus. Glücklicherweise ist das nicht nötig und wir wollen weiterladen. Leider ergeben sich bei uns ab hier ständige Ladefehler, die zum Absturz des Systems führen. Diese Fehler müssen aber nicht bei jeder Version auftreten. Trotzdem sollten Sie sich vor dem Kauf vergewissern, daß der Händler ein schadhaftes Spiel zurücknimmt.

Ist das Game komplett in den Speicher geladen, so erscheint eine gut aufgemachte Highscore-Liste. Erst nach Tastendruck lädt der hervorragend animierte Ameisige sein Gewehr durch und das Spiel kann beginnen.

Der Spielbildschirm bei Rimrunner besteht aus einem bewegten Vordergrund, auf dem die Hauptfigur mit seinem Reittier nach links oder rechts reiten kann. Im Hintergrund scrollt die Silhouette einer Weltraumkolonie fleißig den Bewegungen des Sauriers nach.

Während Sie Ihre Figur über den Planeten lenken, fällt allerlei Zeug vom Himmel, das Sie abschießen sollten. Ab und zu schwebt Ihnen ein Generator entgegen. Laden Sie ihn auf, denn das ist das Ziel des Spiels. Da das nicht vom Reittier aus geht, haben Sie die Möglichkeit, abzusitzen und zu Fuß zu gehen.

Sowohl die Grafik des Sauriers als auch die des Reiters ist gut gelungen, sauber animiert und sehr farbenfroh. Die Aliens sind zwar sauber konstruiert, allerdings nur in den seltensten Fällen nennenswert animiert.

Der Vorder- und Hintergrund scrollt fließend in horizontale Richtungen. Da sich der Hintergrund langsamer bewegt, entsteht ein gewisser Tiefeneffekt. Auch die Gebäude sind sauber gezeichnet und glänzen durch ihre Farbenfreude.

Außer den gut gelungenen Grafikeffekten hat das Spiel noch mit einigen hervorragenden Musikstücken aufzuwarten. Besonders der

Soundtrack der Highscoreliste läßt keine Wünsche mehr offen. Aber auch die Hintergrundgeräusche wurden gut programmiert. Sowohl die Explosionen und Schußgeräusche als auch der Pfiff des Ameisigen, der sein Reittier zu sich ruft, tönt ansprechend aus dem Lautsprecher.

Altmodisches Spielprinzip

Trotz dieser vielen Pluspunkte fehlt es Rimrunner an etwas sehr Wichtigem: der Originalität. Das Spielprinzip ist absolut nichts Neues, ganz im Gegenteil: Die Programmierer scheinen in die Anfangszeit des Commodore 64 zurückgekehrt zu sein. Allerdings nur in ihren Ideen, nicht in den Programmiertechniken.
Durch die altmodische Spielidee flaut die anfängliche Begeisterung über Grafik und Sound schnell wieder ab und verwandelt sich in Langeweile.

Fazit

Hätte Rimrunner nicht diese antiquierte Spielidee, wäre ein erstklassiges Actiongame daraus geworden. In Sachen Grafik und Sound hat das neue Palace-Produkt nämlich eine ganze Menge zu bieten. So aber wird es schnell langweilig, da sich die gesamte Handlung auf hirnloses Ballern beschränkt. Fans solcher scrollenden Ballergames kommen jedoch auf ihre Kosten. • *mn*

Titel: Rimrunner
Getestet: C-64
Umsetzungen: Schneider CPC, Atari ST

Im Test: 

Preis (DM): 49,90

☒ Joystick
☐ Tastatur
☐ Maus

Wertung	0	25	50	75	100
Grafik					
Sound					
Bedienung					
Motivation					

10 Great Games II

Vor einem guten halben Jahr überraschte der Engländer Gremlin mit einem Software-Sampler für den C64. „10 Great Games" eroberte relativ schnell die Hitparaden. Jetzt ist der Nachfolger da.



Software-Sampler, auch Compilations genannt, sind eine tolle Sache. Der Käufer erwirbt in der Regel fünf bis zehn Spiele zum Preis von einem. Die Spiele sind deshalb aber keineswegs schlecht oder seit langem aus der Mode, sondern lediglich nicht mehr die neuesten auf dem aktuellen Software-Markt. Oft lohnt sich der Kauf eines solchen Samplers wesentlich mehr als der eines schlechten aktuellen Spieles. So auch bei „10 Great Games II" von Gremlin.
Auf drei Kassetten beziehungsweise Disketten befinden sich, wie der Name schon sagt, zehn Spiele für den C64. Ob es auch Umsetzungen für andere Computersysteme geben wird, ist noch nicht bekannt.
Auf Kassette beziehungsweise Diskette eins finden Sie die Spiele Water Polo, Rebounder, Convoy Raider und Samurai Triology. Besonders hervorzuheben ist Water Polo. Sie sind eingeladen, an einer Partie Wasserball teilzunehmen. Verhelfen Sie Ihrer Mannschaft zu Sieg und Ruhm. Gute Grafik und flotter Sound unterstützen Sie dabei.
Auf Datenträger zwei gibt's die Fortsetzung des Klassikers Jack The Nipper. Diesmal muß Jack in komplizierten Höhlensystemen seine Bosheit unter Beweis stellen.
Wer den gleichnamigen Kinofilm gesehen hat, darf sich Basil, The Great Mouse Detective nicht entgehen lassen. Lösen Sie einen schwierigen Fall in Londons schäbigen Hafenstraßen und toten Wasserkanälen.
Wenn Sie sich statt mit ungelösten Rätseln lieber mit brutalen Gegnern herumschlagen, dann ist Mask genau richtig für Sie. Bekämpfen Sie mit ihren Truppen die bösen und trügerischen Venoms.
Auf Datenträger Nummer drei finden sich Bulldog, Thing Bounces Back und Auf Wiedersehen Monty. Da die drei Programme noch relativ aktuell sind, verzichte ich auf eine nähere Beschreibung.

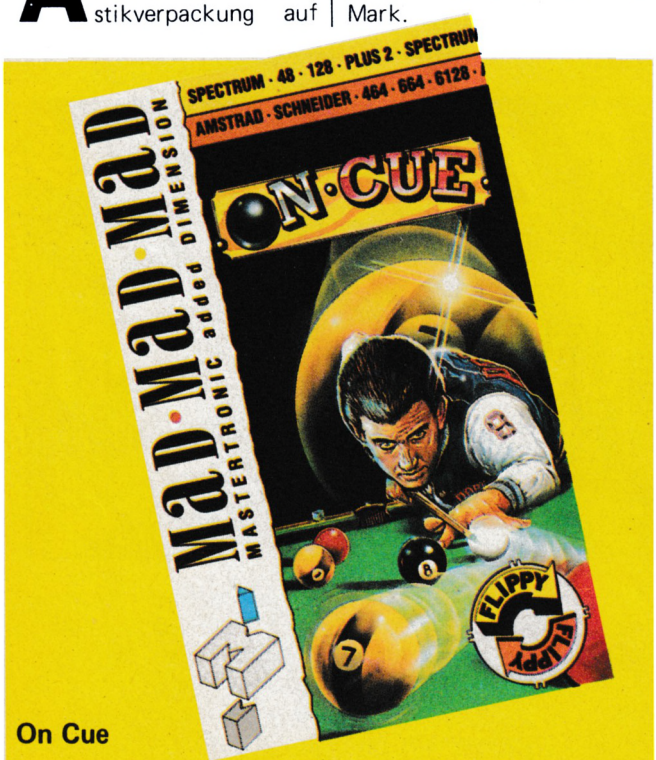
Fazit

Wer die einzelnen Spiele noch nicht besitzt oder kennt, sollte so schnell wie möglich zugreifen. Und wer ein paar Mark übrig hat, kann sich den Vorgänger „10 Great Games" auch noch anschauen. • *TB*

Games für kleine Geldbeutel

Daß Spiele nicht unbedingt teuer sein müssen, bewies uns wieder einmal die Firma Mastertronic. Lesen Sie selbst, womit der bekannte Software-Hersteller diesen Monat aufwartet.

Alle Spiele werden in der gewohnten Plastikverpackung auf Kassette geliefert. Sie kosten zwischen 10 und 15 Mark.



On Cue

Wer eine preiswerte Billiard-Simulation sucht, sollte sich bei seinem Fachhändler nach On Cue erkundigen. Gute Grafiken und eine flotte Hintergrundmusik lassen einer flotten Billiard-Partie nichts mehr im Wege stehen. Es können zwei Spieler gegeneinander antreten, nicht gerade eine Selbstverständlichkeit bei Simulationen dieser Art. Natürlich können Sie auch allein gegen den Computer spielen. Allerdings ist dieser ein so ausgezeichneter Spielpartner, daß Sie spätestens nach zwei verlorenen Partien die Lust verlieren werden.
Gesteuert wird On Cue mit dem Joystick oder über die Tastatur. Die Schneider-CPC-Besitzer sollten einen Farbmonitor ihr eigen nennen. Zu haben ist das Game für den Schneider CPC und den Spectrum.

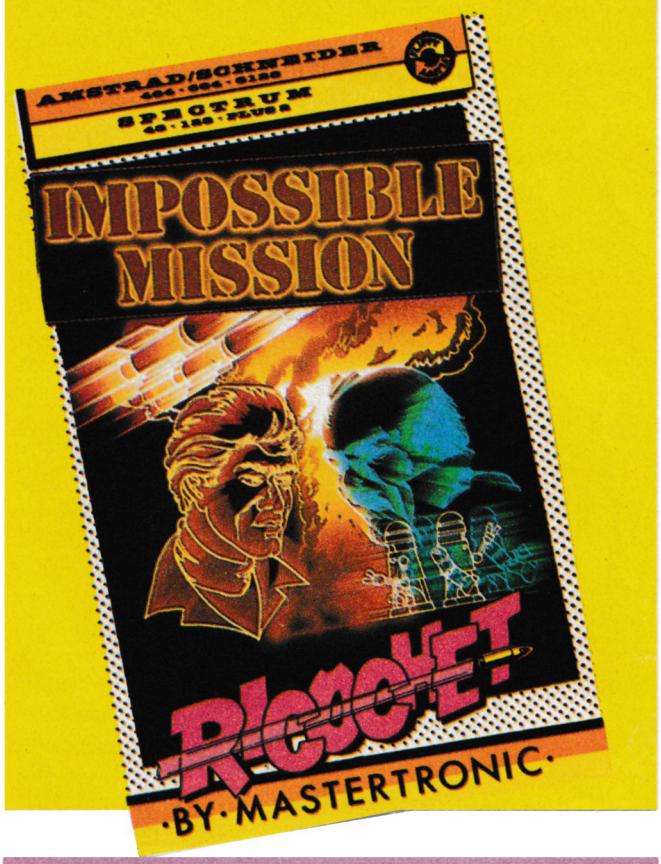
The Official FA Cup Football

Ebenfalls für CPC und Spectrum wird das offizielle Spiel zum Fußball-World-Cup angeboten. Sie müssen eine oder mehrere Mannschaften managen, das heißt Begegnungen arrangieren, Spieler ein- und verkaufen und vieles mehr. Vom Aufbau her erinnert dieses Game an den legendären Football Manager. Auch FA Cup besitzt keine richtige Grafik, an Sound ist lediglich ein Piepston zu hören.
Bis zu acht Spieler können teilnehmen. Allerdings wirkt sich eine hohe Spielerzahl negativ auf die Motivation aus, da man länger warten muß, bis man wieder an die Reihe kommt. Im großen und ganzen kann ich Ihnen von diesem Game nur abraten, denn der Spielablauf ist recht eintönig. Wer jedoch an Football Manager Gefallen gefunden hat, kann sich das Game ja mal ansehen, zumal es mit 15 Mark auch nicht allzu teuer ist.



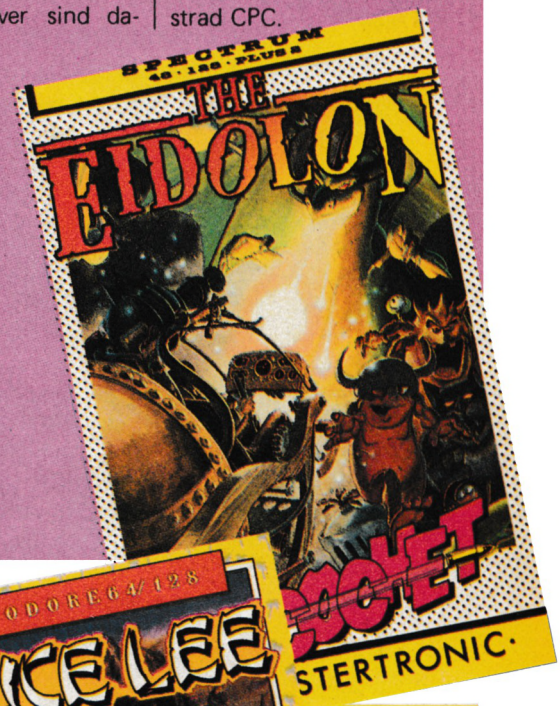
Impossible Mission

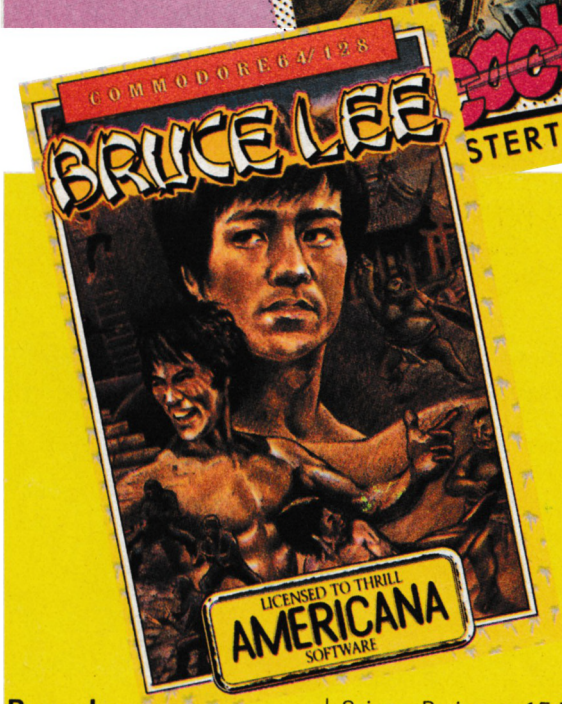
Mit diesem Game hat Mastertronic, wie in letzter Zeit schon oft, die Vertriebsrechte für einen Oldie erworben, den man allerdings besser als Goldie bezeichnen sollte. Erinnern Sie sich zwei Jahre zurück. Ein verrückter Professor ist aus dem Irrenhaus ausgebrochen und will mal wieder die Welt erobern. Zu diesem Zweck baut er sich ein unterirdisches Laboratorium, das er von Robotern bewachen läßt. Im Inneren seiner Höhle schmiedet er einen grausamen Plan zur Vernichtung der Menschheit. Nur ein Mann kann ihn stoppen: Elvin Atombender, dessen Rolle Sie übernehmen dürfen. Dringen Sie in das unterirdische Reich des Wahnsinnigen ein, zerstören Sie seine Apparaturen und nehmen Sie ihn gnadenlos fest. Genauso spannend wie die Story klingt, ist auch das Spiel. Von der Grafik her gehört Impossible Mission trotz seines Alters immer noch zur Spitzenklasse. Alle Sprites sind sehr schön animiert und lassen kaum Wünsche offen. Zur aktuellen Spielsituation erklingen diverse Hintergrund-Geräusche, auf manchen Systemen sogar digitalisiert. Impossible Mission können Sie künftig mit dem Joystick auf dem C64 und dem Schneider CPC spielen.



The Eidolon

Vor gut 18 Monaten war The Eidolon noch in den Software-Top-Ten zu finden. Auch für dieses Spiel hat Mastertronic nun die Rechte aufgekauft und vertreibt es künftig als Billigspiel. Ihre Aufgabe ist es, mit einem Raumgleiter durch ein unterirdisches Höhlensystem zu fliegen und dabei magische Juwelen aufzusammeln. Diese benötigen Sie, um von einem Level ins nächste zu gelangen, denn am Ende jedes Levels wartet ein feuerspeiender Drache, der magische Juwelen überhaupt nicht verträgt. Auf Ihrem Höhlenflug werden Ihnen viele Gesellen begegnen, die Ihr Raumschiff zerstören wollen. Gewagte Ausweichmanöver sind daher an der Tagesordnung. Gutgemachte dreidimensionale Grafiken und flotte Soundeffekte lassen The Eidolon zu einem Vergnügen werden. Das Spiel gibt's für C64 und Schneider/Amstrad CPC.





Bruce Lee

Über dieses Spiel brauche ich wohl keine großen Worte zu verlieren. Ihre Aufgabe ist es, als Bruce Lee in den magischen Tempel einzudringen und diverse Gegenstände aufzusammeln, die Punkte einbringen. Bruce Lee ist der Urahn der Jump-and-Run-Spiele, macht aber immer noch viel Spaß. Seinen Preis von 15 Mark ist es allemal wert. Bruce Lee ist für C64, Schneider/Amstrad und Spectrum erhältlich.

Fazit

Auf dem Billigsoftwaremarkt tut sich einiges, wie man sieht. Die LOAD & RUN-Redaktion bleibt am Ball. ● ***TB***

Kurzberichte

In diesem Monat git es wieder für jeden Rechner Neuigkeiten auf dem Spielesektor. Da im letzten Spiele-Magazin ja nur die Games für den Commodore 64 betrachtet wurden (weil keine anderen verfügbar waren), sind in diesem LOAD & RUN auch wieder Spiele für Atari ST und Amiga vertreten.

Auf dem Atari ST gibt es diesen Monat – außer den im Spieleteil ausführlich getesteten Games – nur zwei nennenswerte Neuheiten: Trauma und Bob Moran IV, beide von der französischen Softwarefirma Infogrames.

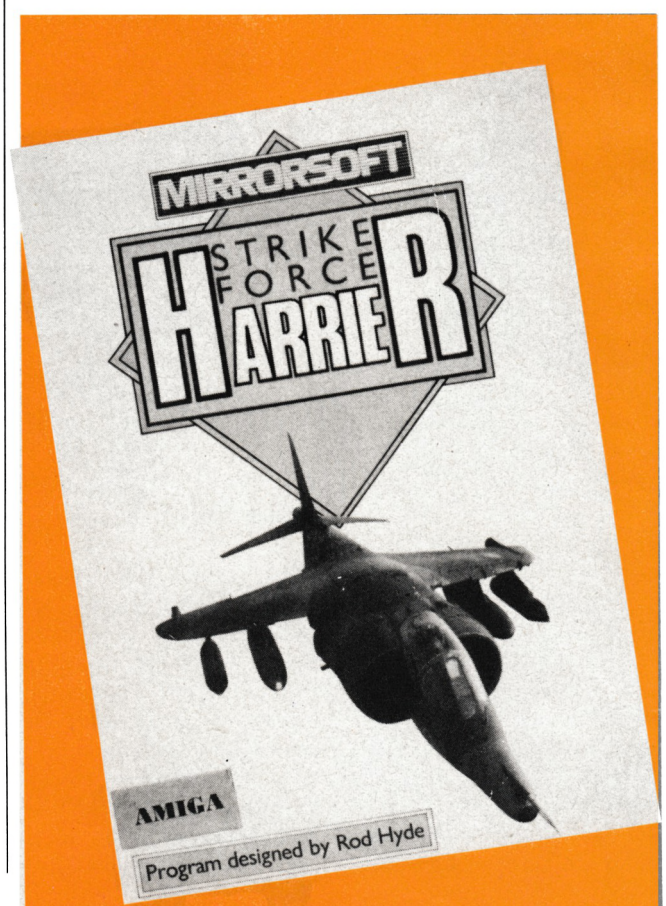
Trauma: Im Weltraum nichts Neues

Bei Trauma handelt es sich wieder einmal um ein wüstes Weltraumballerspiel, wie es sie schon zu hunderten gibt. Nach Angaben des Herstellers soll jedoch etwas ganz Besonderes an ihm dran sein.

Bob Moran: Abenteuer unter Wasser

Bob Moran IV ist bereits der vierte Teil der Helden-Serie. Nach den drei Vorgängern mit den Titeln Rittertum, Im Jungel und Science Fiction spielt der neueste Teil tief unter der Meeresoberfläche. Auch hier hat der Held wieder gegen übermächtige Feinde zu kämpfen und ist wie immer mit supermodernen Geräten und Waffen ausgerüstet.
Die Atari-Version ist ganz akzeptabel, bei der Commodore 64-Adaption bleibt abzuwarten, ob sie genauso ist, wie bei Teil I.

Für den Amiga gibt es zwei Neuheiten, die eigentlich nicht so ganz neu sind. Strike Force Harrier wurde vom Schneider- und Commodore-Rechner auch auf den Amiga adaptiert. Heraus kam eine mittelmäßige Mischung aus Flugsimulator und Aktiongame, die einem richtigen Simulator wie Jet nicht einmal annähernd das Wasser reichen kann. In Sachen Geschwindigkeit bleibt zwar kein Wunsch offen, technisch gesehen unterscheidet den Simulator von einem primitiven Ballerspiel mit einigen Extras nicht mehr viel. Alle nötigen Funktionen sind vorhanden, aber auch nicht mehr. Außerdem fehlt bei Simulator die Vorstellung des Fluggefühls. Die Landschaft fiel sehr monoton aus. Außer einigen Feldern und wenigen Bergen (Pyramiden) hat das Programm nichts zu bieten. Auf dem Commodore 64 wäre Strike Force Harrier eine Wucht, aber auf dem Amiga müssen schon andere Maßstäbe gesetzt werden.



Witzball: Wie auf dem Commodore64

Eine andere Adaption nennt sich Witzball. Auf dem 64er schon als Superspiel bekannt, wurde es nun auf den Sechzehn-Bitter übernommen – und enttäuscht ebenfalls ein wenig. Es sind kaum Veränderungen gegenüber der Commodore-Version feststellbar. Der Sound ist kläglich. Außer einer Titelmelodie tönt nur ein dumpfes Aufprall- und Ballergeräusch aus dem Lautsprecher. Die Grafik wurde nur wenig verbessert, ist aber trotzdem akzeptabel. Vom Spielwitz ging nichts verloren, da auch das C64-Game eine Menge davon besaß. Trotz der wenigen Verbesserungen ist Witzball also immer noch sehenswert – auch auf dem Amiga.

Iznogud: Eine Katastrophe für sich

Ebenfalls drei neue Games gibt es für den Commodore64.
Iznogud ist ein weiteres Spiel von Infogrames, das wir wegen Qualitätsmangels nicht ausführlich testen wollten. Es lehnt sich an die bekannte Comicserie vom Kalifen an und ist wieder einmal ein reines Geschicklichkeitsspiel. Qualitativ ist es kaum eine Erwähnung wert. Ein derart unerkennbares Durcheinander von verschiedenen Pixeln, die Sprites darstellen sollen, bekommt man selten geboten. Selbst das vor einigen Ausgaben angesprochene Ninja Hamster ist nichts dagegen. Auch der Sound ist eigentlich keine Erwähnung wert. Iznogud ist ein völlig daneben-geratenes und unspielbares Game. Qualitativ eine Katastrophe.

Ein Strategiespeil namens Patton versus Rommel veröffentlichte Electronic Arts.

Patton Versus Rommel: Wüste(r)n Krieg

Das Game ist spieltechnisch sehr komplex aufgebaut. Darunter hat die Programmierung allerdings nicht wenig gelitten. Die Grafik ist teilweise nur sehr schwer zu erkennen und dazu langsam bewegt, obwohl nur Icons über eine Landkarte bewegt werden. Überhaupt läßt die Programmgeschwindigkeit etwas zu wünschen übrig. Von Electornic Arts ist der Bildschirm-Spider eigentlich Besseres gewöhnt. Da das Strategiespiel zudem den Afrikakrieg im Zweiten Weltkrieg simuliert, ist Patton versus Rommel eher ein Fall für die Bundesprüfstelle als für einen Testbericht.

Starfleet: VC-20-Zeiten im Weltraum

Aber Electronic Arts schlug nocheinmal mit einem Simulations- und Strategiespiel zu. Und wieder ging der Schlag daneben. Starfleet ist für den Commodore64 und Amiga erhältlich. Wir besprechen jedoch nur den CBM-Version, da sie der Amiga-Variante fast gleicht – oder umgekehrt. Jedenfalls erweckt Starfleet den Eindruck, als wäre für den Amiga zu diesem Game gleich auch ein C64-Emulator mitgeliefert worden – der Einfachheit halber. So ähnlich sind die beiden Versionen.
Das Game macht den Eindruck, als sei es in purem BASIC geschrieben, so primitiv und langweilig ist es. Es läßt sich am ehesten mit einem etwas komplizierten „Schiffeversenken im Weltraum" beschreiben. Der Bildschirm ist nur in ein Textfenster, ein Rasterfeld und ein Statuswindow unterteilt. Das Feld ist sehr klein geraten und die Symbole darauf sind derart winzig, daß sie schon gar nicht mehr ins Auge fallen. Die Schrift ist in normalem Zeichensatz geschrieben und wird einer Art und Weise gedruckt, wie wir sie aus den BASIC-Zeiten des Commodore kennen. Der Spielwitz hält sich das ganze Spiel über auf einem Niveau um Null. Das Game nützt den Rechner nie und nimmer aus und ist auf dem Amiga und dem Commodore gleich schlecht. Bleibt zu hoffen, daß die beiden Strategiespiele ein einmaliger Ausrutscher bei der sonst so guten Produktreihe von Electronic Arts bleiben. Ein schlechter Monat für das Edel-Softwarehaus. ● *mn*

Computer Classics

Neben dem ebenfalls in dieser Ausgabe vorgestellten Spiele-Sampler 10 Great Games traf von Ariolasoft eine weitere Compilation ein: Computer Classics für den Commodore C64 (128 im 64er Modus).

Auf der Diskette finden Sie fünf Spiele, die zwar nicht mehr ganz neu, aber trotzdem noch ihr Geld wert sind. Vor allem für Ein- und Umsteiger bieten solche Software-Sampler ja bekanntlich viel Spiel für wenig Geld.

Dynamite Dan

Das seinerzeit äußerst beliebte Jump-and-Run-Spiel Dynamite Dan verführt Sie in die Welt des bösen Doktor Blitzen, der einen Plan zur Eroberung der Erde entworfen hat. Sie als der Held Dynamite Dan müssen diesen Plan stehlen. Da das Spiel über 30 verschiedene Screens hat, kann die Suche unter Umständen sehr viel Zeit beanspruchen. Dadurch wird die Sache nicht so schnell langweilig.

Into The Eagles Nest

Im Action-Adventure Into The Eagles Nest müssen Sie Gefangene aus dem Lager des Feindes befreien. Aber Vorsicht, denn der Gegner schläft nicht und hat seine gesamte Armee mobilisiert, um Ihren Helden zu vernichten. Es darf geballert werden. Grafik und Sound sind nicht gerade überragend, aber dennoch ausreichend.

Zynaps

Ein weiteres Ballerspiel ist Zynaps. Hier dürfen Sie sich so richtig austoben. Mit den modernsten Waffen ausgerüstet, müssen Sie das Lager der feindlichen Aliens zerstören. Gute Grafiken und eine flotte Hintergrundmusik lassen so schnell keine Langeweile aufkommen.

The Pumpkin Strikes Back

Die böse Hexe hat all Ihre Kürbiskumpane vernichtet. Sie, der letzte überlebende Kürbis, haben Rache geschworen. Dringen Sie ins verwünschte Schloß der Hexe ein, finden Sie die Übeltäterin und schneiden Sie ihr eine Haarsträhne ab, aus der anschließend ein Zaubertrank gebraut werden muß. Die Grafik von Cauldron II ist sehr liebevoll gestaltet und auch der Sound läßt kaum Wünsche offen. Meiner Meinung nach ist Cauldron II das gelungenste Spiel auf der Diskette.

Aliens

Die Aliens kommen. Bei diesem Spiel handelt es sich um das offizielle Computerspiel zum Kinofilm Aliens, der bei uns vor einigen Jahren die Kassen füllte. Sie als Officer Ripley müssen die Außerirdischen aufhalten – nicht gerade ein Zuckerlekken. Spaß macht Aliens auf jeden Fall, auch wenn die Grafik bei der C64-Version nicht sonderlich gut gelungen ist. *TB*

◀◀◀ SSS ▶▶▶ Siggis Software Shop ◀◀◀ SSS ▶▶▶
★ Knüllerpreise ★ Ein Preisvergleich lohnt sich immer ★ Knüllerpreise ★

Spiele	Amiga/ST	Spiele	Amiga/ST	Spiele	Amiga/ST
Arctic Fox	54,50/54,50	Jump Jet	42.50/42,50	Superstar Icehoc	64,50/64,50
Arkanoid	64,50/42,50	Karting Grand Prix	26,50/26,50	Terramex	54,50/54,50
Backlash	45,50/44,50	King of Chicago	58,50/58,50	Test Drive	74,50/74,50
Bad Cat	49,50/54,50	Las Vegas	26,50/26,50	Tetris	54,50/54,50
BMX Simulator	42,50/44,50	Leaderboard Golf	64,50/64,50	Time & Magic	54,50/54,50
California Games	64,50/64,50	Lurkins Horror	74,50/74,50	Two & Two Basketb.	64,50/64,50
Chessmaster 2000	72,50/74,50	Moebius	64,50/64,50	Willy the Kid	26,50/26,50
Clever & Smart	54,50/56,50	Phantasie 3	54,50/58,50	Winterolympiade 88	56,50/56,50
Defender of Crown	66,50/64,50	Ports of Call	88,50/—,—	Wizball	54,50/54,50
Dungeon Master	64,50/64,50	Planetfall	—,—/74,50	Xenon	54,50/54,50
Eagles Nest	54,50/56,50	Rings of Zilfin	64,50/64,50		
Emerald Mine	26,50/26,50	Roadwar 2000	54,50/54,50	Anwenderprogramme	Amiga/ST
Fam. Feuerstein	54,50/54,50	Roadwar Europa	68,50/68,50	C-64 Emulator	156,50/—,—
Flight-Simulator 2	99,50/98,50	Roadwars	54,50/54,50	Cambridge Lisp	—,—/358,50
Frightnigth	64,50/64,50	Rolling Thunder	64,50/54,50	Lattice C	528,50/—,—
Ferrari Formula 1	72,50/—,—	Sindbad	68,50/58,50	Lisp	478,50/—,—
Giana Sisters	52,50/54,50	Slaygon	54,50/54,50	Macro Assembler	178,50/—,—
Gridstart	26,50/26,50	Soccer King	26,50/26,50	MCC Pascal	—,—/248,50
Hotball	64,50/64,50	Strip Poker 2 Plus	42,50/44,50	Pascal	248,50/—,—
Iridon	42,50/—,—	Sub Battle Sim.	64,50/64,50	Pro Sprite Designer	—,—/ 99,50

S. Gebauer
Parkstr. 7a
5880 Lüdenscheid
Tel. (0 23 51) 2 45 02

◀◀ SSS ▶▶
◀◀ SSS ▶▶
◀◀ SSS ▶▶
◀◀ SSS ▶▶

Versandkosten: Vorkasse + DM 4,50 / Nachnahme + DM 7,50. Zur Auslieferung gelangt ausschließlich nur Originalware. Angebote freibleibend. Liefermöglichkeiten vorbehalten. Bei großer Nachfrage nicht jeder Artikel sofort lieferbar.

Players' Pages

Hallo, Spielefreaks und Joystickartisten! Willkommen bei den Players Pages des LOAD & RUN-Spielemagazins. Wer Tips, Tricks, Lösungen oder Karten zu jedem x-beliebigen Spiel für jedes x-beliebige System hat oder sucht, der ist hier genau richtig.

Zuerst mal ein dickes Lob an die MSX-User, die sich diesmal für die Players' Pages besonders ins Zeug gelegt haben.

MSX-Spielepokes

Von unserem Leser Dag Frommhold aus Neuffen stammen die folgenden kleinen Poke-Programme für den MSX-Computer. Einfach abtippen, starten und staunen.

```
 5 'Hunchback the Adventure
10 BLOAD"CAS:"
20 POKE -28370,0
30 DEFUSR=&H9000:A=USR(0)

10 ' Turmoil
20 LOAD "CAS:"
30 POKE &HEC1,&H0
40 DEFUSR=&HE646:A=USR(0)

10 ' Mutant Monty
20 CLEAR 100,&H87FF
30 BLOAD "CAS:"
40 POKE -27872,0
50 DEFUSR=&H9000:A=USR(0)
```

Dag hat auch beim Spiel Valkyr einen Cheat-Mode entdeckt, in den Sie durch gleichzeitiges Drücken von TAB, CTRL, SHIFT, ESC, CAPS, CURSOR UP, DOWN und LEFT, allerdings erst im Hauptmenü, gelangen. Als kleine Zugabe noch die Keywörter für The Goonies: GOONIES, MR SLOTH, GOON DOCKS.

Masters Of The Universe

Unser Leser Hanno Schehly hat uns für alle Masters of the Universe einen Stadtplan zugeschickt, auf dem die magischen Chords markiert sind. Jetzt dürfte es wirklich kein Problem mehr sein, das Spiel zu beenden.

Alles für den CPC

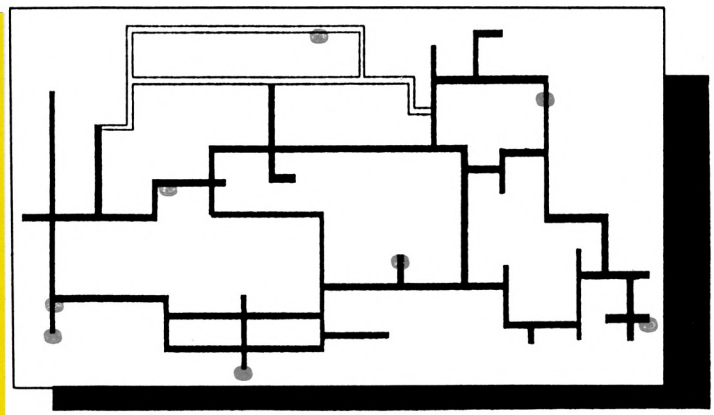
Markus Sledzinski hat sich für die CPC-User in der Spielewelt umgesehen – mit Erfolg, wie Sie an der nachfolgenden Poke-Liste sehen können.

SIGMA SEVEN	POKE 34432, anzahl
COPOUT	POKE &13A,&FF
NEMESIS	POKE &33C,&00
PROJECT FUTURE	POKE &9B9C,&08
ANDROID ONE	POKE &7391,&00
SPINDIZZY	POKE &A85E,&C9
SHOCKWAY RIDER	POKE &7D8E,&00
KILLAPEDE	POKE &4A14,&00
IMP. MISSION	POKE &12BF,&C9
WIZBALL	POKE &7CC2,&00
BLAGGER	POKE &7CC2,&00 POKE &7CC4,&00
DRUID	POKE 11265,00 POKE 8449,195
MIKIE	POKE 25116,195
BOMBSCARE	POKE &A1F0, &FF
ARMY MOVES (1)	POKE &0B3E,&00
ARMY MOVES (2)	POKE &0865,&00
CLASSIC AXIENS	POKE &478C,&7F

Wer weiß mehr?

Auch diesmal sind wieder viele Fragen zu beantworten. Wer kann weiterhelfen? Eine ganze Menge Probleme hat Hanno Schehly. Bei Jack The Nipper findet er die Diskette nicht. Wie kann man den Leim verwenden? Beim Spiel Nightmare findet Hanno die Exit-Steine nicht und für Molecule Man benötigt er einen Plan. Auch für das Spiel Die Erbschaft wird dringend Hilfe benötigt. Wer nimmt im ersten Teil welchen Gegenstand und was muß man im Flughafen (Teil 2) machen? Wer Hanno helfen kann, möge uns bitte schreiben.

Euer
Thomas Bosch



Von ihrem Ausgangspunkt betrachtet, müssen Sie versuchen möglichst ohne Umwege, ihr Ziel zu erreichen. Hier ist der Lageplan des Labyrintes aus dem Spiel Heman

Der clevere Kontakt:

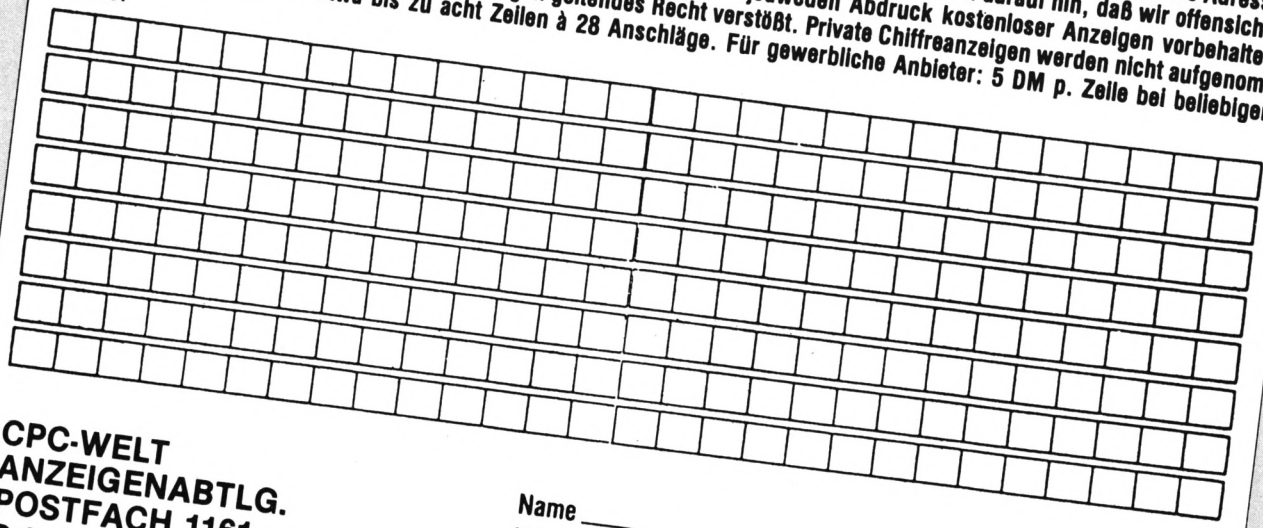
Immer die neueste Software zu absolut coolen Preisen! Testen Sie uns noch heute, wir sind jederzeit für Sie da!



SOFTWARE VERSAND
Andreas Bachler
Postfach 429
D-4290 Bocholt
Tel. (0 28 71) 18 30 88

ANZEIGENSERVICE

Die große Börse für jeden Zweck in CPC-WELT. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: etwa bis zu acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. Zeile bei beliebiger Größe.



CPC-WELT
ANZEIGENABTLG.
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____

DAS SONDERANGEBOT: PRIVATE KLEINANZEIGEN SIND KOSTENLOS

Das bietet Ihnen CPC-WELT. KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!
Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!
Wir veröffentlichen nur Kleinanzeigen privater Inserenten kostenlos, gewerbliche Anzeigen kosten pro Zeile zu 29 Buchstaben DM 5,00 plus Mehrwertsteuer!
Wir versenden für Privat-Inserenten keine Belegexemplare!

DIE INSERTION IST NICHT VOM HEFTKAUF ABHÄNGIG!
Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!
Anzeigenabdruck in der Reihenfolge ihres Eingangs, kein Rechtsanspruch auf den Abdruck in der nächsten Ausgabe!
Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes — Computer — gehören, nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeigenteils zuläßt.

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 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 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.

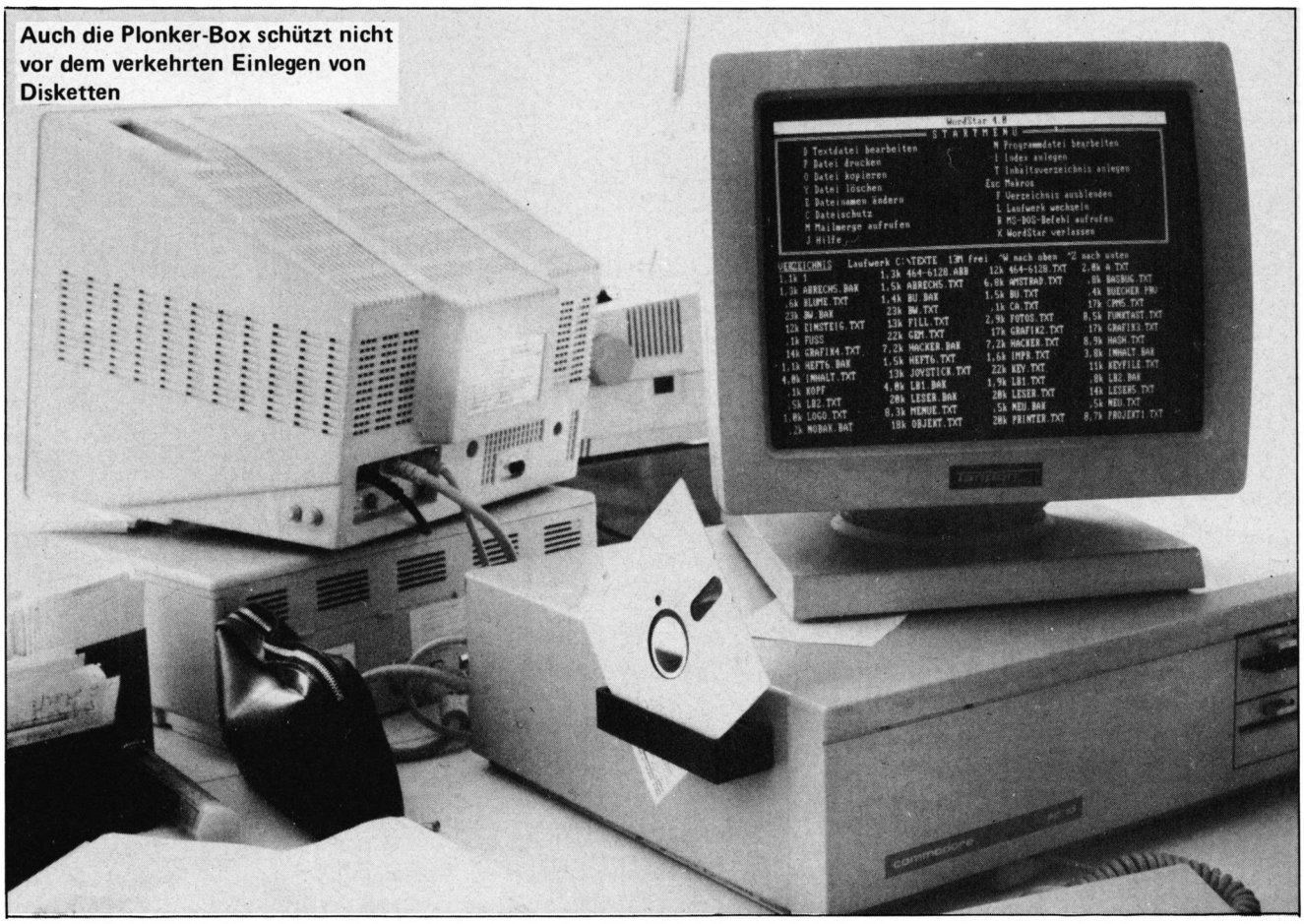
_____

Rechtsverbindliche Unterschrift

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

Fortsetzung von Seite 62

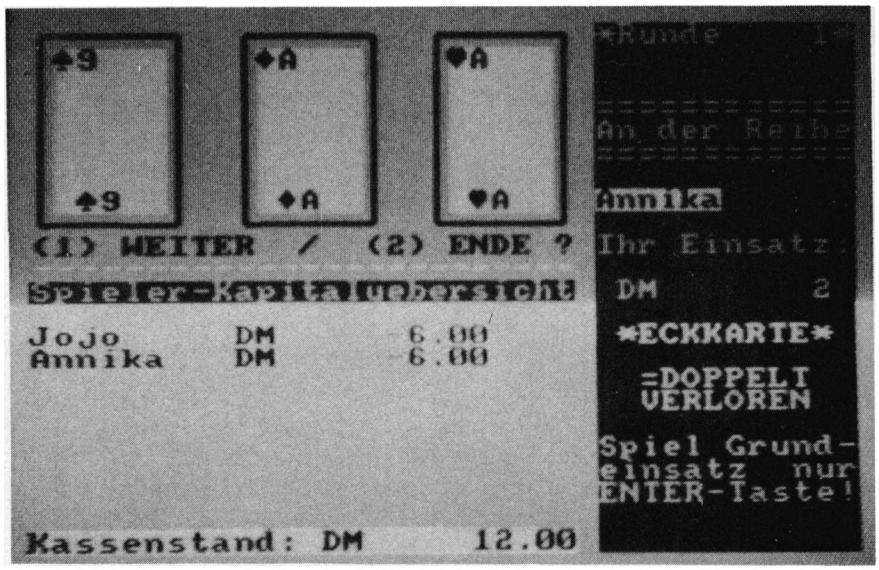
Auch die Plonker-Box bewahrte uns nicht vor dem Verlust einiger Disketten-Hüllen. Daher wanderten einige Disketten einfach ohne Hüllen in die Box. Wieder zeigten sich einige Schwachstellen im Design.
Gerade das, was die Plonker-Box verhindern soll, nämlich Berührungen der Schreib-Lese-Öffnung, trat häufiger auf als zuvor. Der Plonker bietet nämlich keinen Schutz davor, eine Diskette verkehrt einzulegen. Ein Hinweis zur richtigen Benutzung ist hier durchaus angebracht. Für Anwender, die des Lesens nicht mächtig sind, ist eine Bebilderung unentbehrlich.
Um solche Fehlbedienungen zu verhindern, sollte die technische Ausstattung der Plonker-Box nochmals überarbeitet werden. Eine Verbreiterung des Plonkers in Verbindung mit einem fotoelektrischen Sensor und einem einfachen Piezo-Piepser bietet die Möglichkeit, das lagerichtige Einführen einer Disk zu überprüfen und den Anwender gegebenenfalls zu warnen.
Fairerweise müssen wir aber erwähnen, daß die Berührung der Schreib-Lese-Öffnung unsere Disketten in keiner Weise beschädigte. Der Grund lag wohl in der Verwendung von 3,5-Zoll-Disketten.
Im Gegensatz zu den Behauptungen des Herstellers ist der Plonker nicht nur für 5,25- und 3,5-Zoll-Disks geeignet. In wochenlangem Einsatz an einem Schneider CPC traten auch mit dem Drei-Zoll-Format dieses Rechners nicht die geringsten Probleme auf. Die oben angeführte Behauptung ist also ein unnötiges Understatement, mit dem die Firma sich dem großen Kundenkreis der Schneider-Anwender verschließt.
Auch der Einsatzbereich der Plonker-Box wurde vom Hersteller nicht richtig eingeschätzt. Wer sich in jahrelanger Arbeit an einem unaufgeräumten Schreibtisch an die Stapelverarbeitung bei der Diskettensuche gewöhnt hat, sucht die am häufigsten benötigten Disks überall, nur nicht da, wo sie sind: in der Box.
Um dem Computer-Anwender den Umstieg auf den Plonker zu erleichtern, sollten also zunächst die selten oder nie benötigten Disketten in den Plonker wandern.
Wer Schreibtisch-Chaos oder Chaos-Schreibtische kennt, weiß es schon lange: Wo Ordnung herrscht, wird zuletzt gesucht.
Mit den oben vorgeschlagenen Verbesserungen und vor allem einem ausführlichen Benutzer-Handbuch könnte der Plonker allerdings zu einem durchschlagenden Erfolg im Bereich der Computer-Peripherie werden. Wir hoffen, daß der Hersteller auf diese Anregungen reagieren wird, und werden Sie ständig über den neuesten Stand der Entwicklung auf dem laufenden halten. *AE* □



Auch die Plonker-Box schützt nicht vor dem verkehrten Einlegen von Disketten

ECK-KARTE

Das Programm unseres Autors Bernd Ehring ist kurzweilig und trotzdem recht interessant. Zwei bis acht Spieler können sich daran beteiligen.
Es geht um Geld: Spielgeld natürlich. Pro Durchgang erhält jeder Spieler zwei Karten vorgeblendet. Je weiter diese Karten im Wert auseinander liegen, desto größer ist die Chance, daß die dann folgende dritte Karte dazwischenpaßt. Vorher ist noch der Einsatz zu bestimmen.
Paßt die Karte, so erhält der jeweilige Spieler den Einsatz auf seinem Konto gutgeschrieben. Liegt sie jedoch im Wert unter der niedrigen oder über der hohen Karte, so hat der Spieler Pech gehabt und muß den Einsatz an die Kasse abgeben.
Stimmt die dritte Karte gar mit der kleineren oder der größeren überein, ist doppelter Einsatz zu zahlen.



Immer wieder beliebt sind Kartenspiele auf dem Computer. Hier können Sie Ihr Glück versuchen, ohne dafür tief in die Tasche greifen zu müssen.

Wenn die Kasse leer ist, zahlt jeder Spieler einen gleich hohen Betrag ein. Als höchster Einsatz kann der Betrag gewählt werden, der sich im Augenblick noch in der Kasse befindet.
Ein schnelles Spiel, eine gute Grafik. Also ran ans Tippen. Wem diese Arbeit zuviel ist, der kann aber auch unsere Softbox bestellen und spart Zeit. Für's Spiel nämlich. *JE*□

```
10 '******************************* (24C3)
20 '*          ECKKARTE           * (249B)
30 '*            VON              * (2407)
40 '*        BERND EHRING         * (2493)
50 '*            FUER             * (247E)
60 '*     SCHNEIDER CPC-WELT      * (245A)
70 '*      CPC 464/664/6128    JE* (24D1)
80 '******************************* (244E)
90 '                               (07BF)
100 MODE 1                         (0770)
110 DIM k$(32):DIM aus(32):DIM wer
t(32)                              (2787)
120 DATA 7,8,9,10,B,D,K,A          (1759)
130 FOR i=226 TO 229               (1055)
140 FOR l=1 TO 8                   (0E2F)
150 READ x$                        (0A5C)
160 z=z+1:w=w+1                    (1C3A)
170 k$(z)=CHR$(i)+x$               (1D44)
180 wert(z)=w                      (173F)
190 NEXT l                         (0AF8)
200 w=0:RESTORE 120                (10FA)
210 NEXT i                         (0A15)
220 PAPER 1:PEN 0:BORDER 10:CLS    (1038)
230 WINDOW 2,39,2,24:CLS           (110C)
240 DRAW 0,0:DRAW 0,399,0:DRAW 639
,399:DRAW 639,0:DRAW 0,0           (2770)
250 '***Spielerklaerung***        (1C83)
260 LOCATE 9,3:PRINT CHR$(222)STRI
NG$(19,207)CHR$(223)               (209A)
270 LOCATE 9,4:PRINT CHR$(207)"
                "CHR$(207)         (2CCC)
280 LOCATE 9,5:PRINT CHR$(207)"  E
 C K K A R T E  "CHR$(207)         (2C70)
290 LOCATE 9,6:PRINT CHR$(207)"
                "CHR$(207)         (2C32)
300 LOCATE 9,7:PRINT CHR$(221)STRI
NG$(19,207)CHR$(220)               (20D3)
310 PAPER 2:LOCATE 7,10:PRINT CHR$
(164)" 7/1987 by  Bernd Ehring":PA
PER 1                              (325C)
320 LOCATE 1,12:PRINT"Dies ist ein
 Kartenspiel  fuer 2 bis 8"        (3484)
330 LOCATE 1,13:PRINT"Personen. Es
 wird um Geld gespielt. In"        (34C6)
340 LOCATE 1,14:PRINT"jedem Durchg
ang  erhaelt jeder Spieler"        (347F)
350 LOCATE 1,15:PRINT"zwei Karten
(eine links, eine rechts)."        (3440)
```

```
360 LOCATE 1,16:PRINT"Je weiter di
e Karten im Wert auseinan-"      <34D4>
370 LOCATE 1,17:PRINT"derliegen,
desto groesser die Chance,"      <345C>
380 LOCATE 1,18:PRINT"das die fol
gende Dritte Karte im Wert"      <344E>
390 LOCATE 1,19:PRINT"dazwischen p
asst. Vorher ist der Ein-"       <3409>
400 LOCATE 1,20:PRINT"satz zu best
immen. Passt die Karte da-"      <3475>
410 LOCATE 1,21:PRINT"zwischen,wir
d der doppelte Einsatz vom"      <3447>
420 GOSUB 610:PAPER 2:PEN 0:LOCATE
 7,23:PRINT"Bitte eine Taste druec
ken!":CALL &BB06:PAPER 1:PEN 0:CLS
                                 <40DE>
430 LOCATE 1,1:PRINT"Spieler entno
mmen.Der Einsatz darf nie"       <339E>
440 LOCATE 1,2:PRINT"groesser sein
 als der Kassenstand! Man"       <33BB>
450 LOCATE 1,3:PRINT"kann auch ste
ts um alles spielen. Ist"        <336C>
460 LOCATE 1,4:PRINT"die dritte K
arte aber groesser oder"         <3314>
470 LOCATE 1,5:PRINT"kleiner im We
rt, muss der Spieler den"        <33BE>
480 LOCATE 1,6:PRINT"getaetigten E
insatz in die Kasse zah-"        <33F0>
490 LOCATE 1,7:PRINT"zahlen. Ist j
edoch die dritte Karte im"       <33D7>
500 LOCATE 1,8:PRINT"Wert gar ein
e gleiche wie seine linke"       <338F>
510 LOCATE 1,9:PRINT"oder rechte
Karte, so muss er seinen"        <33BE>
520 LOCATE 1,10:PRINT"doppelten Ei
nsatz in die Kasse zahlen."      <34DD>
530 LOCATE 1,12:PRINT"Ist die Kass
e leer (wie auch am Anfang"      <3432>
540 LOCATE 1,13:PRINT"des Spieles)
,so zahlt jeder den Grund-"      <342C>
550 LOCATE 1,14:PRINT"einsatz ein,
  der frei vereinbart wer-"      <34CC>
560 LOCATE 1,15:PRINT"den kann.
Wenn alles klar ist, bitte"      <3492>
570 LOCATE 1,16:PRINT"eine Taste
druecken und die Namen der"      <34D0>
580 LOCATE 1,17:PRINT"Spieler eing
eben. Viel Spass.....!":GOSUB
610                              <3921>
590 CALL &BB06                   <09EB>
600 GOTO 630                     <095A>
610 FOR so=15 TO 8 STEP-1:SOUND 1,
120,7,so:NEXT so:RETURN          <2A7D>
620 '***Eingaben der Namen und des
 Einsatzes***                    <3118>
630 CLS                          <0619>
640 LOCATE 1,1:PRINT"Bitte geben S
ie die Anzahl der Spieler"       <3397>
650 LOCATE 1,2:PRINT"ein. Mindeste
ns 2, maximal 8 Spieler !"       <33C4>
660 x$=INKEY$:IF x$=""THEN 660   <19D0>
670 s=VAL(x$)                    <1228>
680 IF s<2 OR s>8 THEN GOSUB 610:G
OTO 660                          <1DA9>
690 LOCATE 7,4:PRINT"Es nehmen"s"S
pieler teil ."                   <2AE3>
700 LOCATE 1,6:PRINT"Bitte geben S
ie die Namen der Spieler"        <3332>
710 LOCATE 1,7:PRINT"ein.(Maximal
8 Buchstaben) und beenden"       <33D0>
720 LOCATE 1,8:PRINT"Sie jede Eing
abe mit der [ENTER]-Taste"       <33F2>
730 FOR i=1 TO s                 <110B>
740 LOCATE 1,9+i                 <0E88>
750 PRINT""i". Spieler/Name: -> "
;:LINE INPUT n$(i)               <30E8>
760 IF LEN(n$(i))<1 OR LEN(n$(i))>
8 THEN GOSUB 610:LOCATE 1,9+i:PRIN
T"
    ":GOTO 740                   <64B4>
770 NEXT i                       <0A78>
780 CLS                          <0645>
790 LOCATE 1,1:PRINT"Die Namen si
nd eingegeben. Bestimmen"        <3382>
800 LOCATE 1,2:PRINT"Sie bitte den
 Grundeinsatz in DM. Fuer"       <33EF>
810 LOCATE 1,3:PRINT"50 Pfennig ge
ben Sie z.B. ein 0.50 --"        <335C>
820 LOCATE 1,4:PRINT"stets durch P
unkt '.' trennen.....!!"         <331E>
830 LOCATE 1,6:INPUT"Betrag eingeb
en und [ENTER]: ";grund          <338C>
840 IF grund>99 THEN LOCATE 1,7:PR
INT"Bitte nicht mehr als DM 99,00
Einsatz!":FOR t=1 TO 2300:NEXT t:L
OCATE 1,7:PRINT"
                 ":GOTO 830      <8744>
850 IF grund<=0 THEN GOSUB 610:GOT
O 830                            <1A19>
860 LOCATE 1,8:PRINT"Grundeinsatz
pro Spieler ist DM";USING"###.##";
grund                            <3FB7>
870 FOR i=1 TO s:geld(i)=geld(i)-g
rund:pot=pot+grund:NEXT i        <5311>
880 LOCATE 1,13:GOSUB 610:PRINT"Da
s Spiel kann beginnen. Viel Glueck
 !"                              <39C8>
890 '***Spielfeldaufbau***       <1C81>
900 WINDOW#1,3,7,3,9:PAPER#1,2:PEN
#1,0                             <1C61>
910 WINDOW#2,12,16,3,9:PAPER#2,2:P
EN#2,0                           <1E72>
920 WINDOW#3,21,25,3,9:PAPER#3,2:P
EN#3,0                           <1EC0>
930 WINDOW#4,28,39,2,24:PAPER#4,3:
PEN#4,1                          <1F3E>
940 FOR t=1 TO 2000:NEXT t:CLS   <1814>
950 CLS:LOCATE 1,1:PRINT CHR$(150)
STRING$(5,154)CHR$(156)          <21F7>
960 LOCATE 10,1:PRINT CHR$(150)STR
```

```
ING$(5,154)CHR$(156)                    <2084>
970 LOCATE 19,1:PRINT CHR$(150)STR
ING$(5,154)CHR$(156)                    <20F3>
980 FOR i=1 TO 7                        <0EFB>
990 LOCATE 1,1+i:PRINT CHR$(149)"
    "CHR$(149)                          <239D>
1000 LOCATE 10,1+i:PRINT CHR$(149)
"     "CHR$(149)                        <24DA>
1010 LOCATE 19,1+i:PRINT CHR$(149)
"     "CHR$(149)                        <240E>
1020 NEXT i                             <0A6C>
1030 LOCATE 1,9:PRINT CHR$(147)STR
ING$(5,154)CHR$(153)                    <1FFB>
1040 LOCATE 10,9:PRINT CHR$(147)ST
RING$(5,154)CHR$(153)                   <2008>
1050 LOCATE 19,9:PRINT CHR$(147)ST
RING$(5,154)CHR$(153)                   <207F>
1060 CLS#1:CLS#2:CLS#3:LOCATE 1,11
:PEN 2:PRINT"----------------------
----":PEN 0                             <3969>
1070 GOTO 1140                          <0901>
1080 LOCATE 1,12:PAPER 3:PEN 1:PRI
NT"Spieler-Kapitaluebersicht":PAPE
R 1:PEN 0                               <3308>
1090 FOR m=1 TO s                       <11E5>
1100 LOCATE 1,13+m:PRINT n$(m)          <1B4B>
1110 LOCATE 10,13+m:PRINT USING"DM
 +#####.##";geld(m)                     <2FB6>
1120 NEXT m                             <0A44>
1130 RETURN                             <06FF>
1140 GOSUB 1080                         <0992>
1150 LOCATE 1,23:PAPER 2:PRINT"Kas
senstand: ";USING"DM ######.##";po
t:PAPER 1                               <3888>
1160 '***Spielbeginn***                 <187B>
1170 r=r+1                              <104C>
1180 FOR i=1 TO s                       <1190>
1190 FOR j=1 TO 4:CLS#j:NEXT j          <1BAD>
1200 LOCATE#4,1,1:PRINT#4,"*Runde
";USING"####*";r                        <2834>
1210 LOCATE#4,1,4:PRINT#4,"=======
======"                                 <1F45>
1220 LOCATE#4,1,5:PRINT#4,"An der
Reihe"                                  <1FA3>
1230 LOCATE#4,1,6:PRINT#4,"=======
======"                                 <1F8D>
1240 PAPER#4,2:PEN#4,0:LOCATE#4,1,
8:PRINT#4,n$(i):PAPER#4,3:PEN#4,1       <33AB>
1250 k1=INT(RND(TIME)*32+1)             <1AC4>
1260 IF aus(k1)=1 THEN 1250 ELSE a
us(k1)=1:ka=ka+1                        <386D>
1270 k2=INT(RND(TIME)*32+1)             <1A09>
1280 IF aus(k2)=1 THEN 1270 ELSE a
us(k2)=1:ka=ka+1                        <387F>
1290 dif=ABS(wert(k1)-wert(k2)):IF
 dif=0 OR dif=1 THEN aus(k1)=0:aus
(k2)=0:CLS#1:CLS#3:GOTO 1250            <6D09>
1300 IF wert(k1)<wert(k2)THEN we1=
wert(k1):we2=wert(k2)                   <4F04>
1310 IF wert(k1)>wert(k2)THEN we1=
wert(k2):we2=wert(k1)                   <4F3C>
1320 LOCATE#1,1,1:PRINT#1,k$(k1):L
OCATE#1,2,7:PRINT#1,k$(k1)              <34D2>
1330 LOCATE#3,1,1:PRINT#3,k$(k2):L
OCATE#3,2,7:PRINT#3,k$(k2)              <34C1>
1340 dif=ABS(we1-we2):IF dif-1>1 T
HEN LOCATE 1,10:PRINT"++++"dif-1"K
arten passen ++++"ELSE LOCATE 1,10
:PRINT"+++++"dif-1"Karte passt +++
++"                                     <7EB5>
1350 GOSUB 610:LOCATE#4,1,10:PRINT
#4,"Ihr Einsatz:"                       <25ED>
1360 LOCATE#4,1,19:PRINT#4,"Spiel
Grund-":LOCATE#4,1,20:PRINT#4,"ein
satz  nur":LOCATE#4,1,21:PRINT#4,"
ENTER-Taste!"                           <5804>
1370 LOCATE#4,1,12:INPUT#4,"DM";sa
tz                                      <1E05>
1380 IF satz>pot THEN GOSUB 610:GO
TO 1370                                 <1EC5>
1390 IF satz<grund OR satz=0 THEN
satz=grund                              <3174>
1400 LOCATE#4,1,12:PRINT#4,USING"
DM #######";satz                        <28B0>
1410 k3=INT(RND(TIME)*32+1)             <1A0D>
1420 IF aus(k3)=1 THEN 1410 ELSE a
us(k3)=1:k=k+1                          <3679>
1430 we3=wert(k3)                       <1A6D>
1440 SOUND 1,350,17,13:SOUND 3,900
,13,14:SOUND 5,1900,9,15:FOR t=1 T
O 1000:NEXT t                           <3CF8>
1450 LOCATE#2,1,1:PRINT#2,k$(k3):L
OCATE#2,2,7:PRINT#2,k$(k3)              <3414>
1460 IF we3<we1 THEN LOCATE#4,3,14
:PRINT#4,"Kleiner-":geld(i)=geld(i
)-satz:pot=pot+satz:LOCATE#4,3,16:
PRINT#4,"VERLOREN":GOTO 1510            <821D>
1470 IF we3>we2 THEN LOCATE#4,3,14
:PRINT#4,"Groesser":geld(i)=geld(i
)-satz:pot=pot+satz:LOCATE#4,3,16:
PRINT#4,"VERLOREN":GOTO 1510            <8266>
1480 IF we3=we1 OR we3=we2 THEN FO
R mu=2222 TO 111 STEP-25:SOUND 1,m
u,2,14:NEXT mu                          <4898>
1490 IF we3=we1 OR we3=we2 THEN LO
CATE#4,2,14:PRINT#4,"*ECKKARTE*":G
OSUB 2000:geld(i)=geld(i)-satz*2:p
ot=pot+satz*2:LOCATE#4,3,16:PRINT#
4,"=DOPPELT":LOCATE#4,3,17:PRINT#4
,"VERLOREN":GOTO 1510                   <B3D7>
1500 LOCATE#4,1,17:PRINT#4,"==GEWO
NNEN==":FOR n=3000 TO 100 STEP-30:
SOUND 1,n,1,13:NEXT n:GOSUB 1870:G
OSUB 610:geld(i)=geld(i)+satz:pot=
pot-satz                                <886B>
1510 GOSUB 1090:GOSUB 610               <0EDA>
1520 LOCATE 1,23:PAPER 2:PRINT"Kas
senstand: ";USING"DM ######.##";po
t:PAPER 1                               <386E>
1530 IF ka>18 THEN LOCATE 1,10:PRI
```

```
NT"=Karten  werden gemischt=":FOR
f=1 TO 32:aus(f)=0:SOUND 1,133,4,1
3,4,3,12:FOR t=1 TO 100:NEXT t:NEX
T f:GOSUB 610:ka=0                    <81CE>
1540 IF pot<grund THEN LOCATE 1,10
:PRINT"== Alle  neu einzahlen ==":
FOR g=1 TO s:GOSUB 1990:geld(g)=ge
ld(g)-grund:pot=pot+grund:NEXT g:G
OSUB 1090:FOR t=1 TO 1000:NEXT t      <A395>
1550 LOCATE 1,23:PAPER 2:PRINT"Kas
senstand: ";USING"DM ######.##";po
t:PAPER 1                             <38AB>
1560 IF i=s THEN GOTO 1590            <146C>
1570 LOCATE 1,10:PRINT"*** BITTE E
INE TASTE! ***":GOSUB 610             <2C08>
1580 CALL &BB06:FOR z=1 TO 4:CLS#z
:NEXT z                               <2033>
1590 NEXT i                           <0AE3>
1600 LOCATE 1,10:PRINT"(1) WEITER
 /  (2) ENDE ?"                       <27AC>
1610 d$=INKEY$:IF d$=""THEN 1610      <1943>
1620 IF d$="1"THEN GOTO 1140          <13E5>
1630 IF d$="2"THEN GOTO 1660          <133B>
1640 GOTO 1610                        <092F>
1650 '***Endabrechnung***             <1AA6>
1660 CLS:LOCATE 1,1:PRINT"Das Spie
l ist beendet. Endabrechnung :"       <3540>
1670 teile=pot/s:FOR g=1 TO s:geld
(g)=geld(g)+teile:NEXT g              <518A>
1680 FOR i=1 TO s                     <117A>
1690 LOCATE 2,2+i                     <0E76>
1700 PRINT n$(i)                      <1092>
1710 LOCATE 14,2+i:PRINT USING"DM
######.##";geld(i)                    <2DE8>
1720 LOCATE 26,2+i                    <0FD0>
1730 IF geld(i)=0 THEN PRINT"....+
 - NULL"                              <25FC>
1740 IF geld(i)>0 THEN PAPER 3:PEN
 1:PRINT"....GEWONNEN":PAPER 1:PEN
 0                                    <31DE>
1750 IF geld(i)<0 THEN PRINT"....V
ERLOREN"                              <25E6>
1760 NEXT i                           <0A37>
1770 PRINT:PRINT                      <08DC>
1780 PRINT"Der Kasseninhalt betrug
 ....";USING"DM ####.##";pot          <39FC>
1790 PRINT"und wurde durch die Spi
eler geteilt..!"                      <2EAB>
1800 PRINT:PRINT                      <081A>
1810 PAPER 2:PRINT"(1) Noch einmal
 spielen oder  (2) Ende":PAPER 1:G
OSUB 610                              <39A6>
1820 c$=INKEY$:IF c$=""THEN 1820      <198C>
1830 IF c$="1"THEN RUN                <1031>
1840 IF c$="2"THEN 1860               <12DA>
1850 GOTO 1820                        <096E>
1860 LOCATE 1,23:PRINT" Spiel been
det. Auf Wiedersehen ....!":FOR t=
1 TO 3333:NEXT t:GOTO 1890            <4AC2>
1870 FOR mus=3333 TO 111 STEP-50:S
OUND 2,mus,2,13:NEXT mus:RETURN       <301C>
1880 '***Abschluss***                 <1673>
1890 MODE 0                           <076F>
1900 PAPER 0:PEN 1:CLS                <0C78>
1910 LOCATE 6,12:PRINT"GAME  OVER"
                                      <18B8>
1920 FOR i=25 TO 1 STEP-1             <1295>
1930 INK 1,i                          <0C96>
1940 FOR t=1 TO 66:NEXT t             <159C>
1950 NEXT i                           <0AB3>
1960 FOR t=1 TO 1000:NEXT t           <1625>
1970 MODE 1:INK 1,25:PAPER 0:PEN 1
:BORDER 1                             <16DD>
1980 LOCATE 2,2:PRINT"BASIC 1.0":P
RINT:END                              <1A30>
1990 SOUND 1,500,3,13,15:RETURN       <1501>
2000 FOR k=1 TO 70:far=INT(RND(TIM
E)*24+1):BORDER far:NEXT k:BORDER
10:RETURN                             <3A46>
```

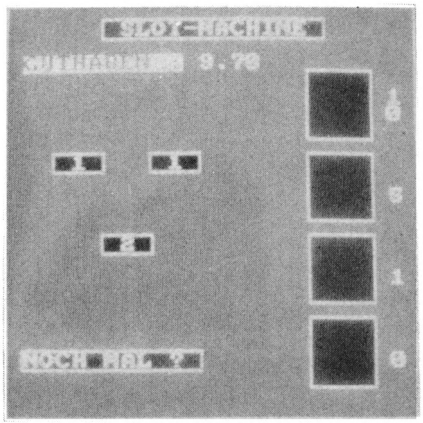
Alle Programme auf Disc erhältlich!

Bestell-Coupon Seite 81

CPC-Welt Hotline! Jeden Montag 15 - 19°° Tel.: 089/18 40 24

SLOT-Maschine

Hier die Simulation eines Spielautomaten, die Autor Thorsten Stumpf „Slotmaschine“ genannt hat. Diese einarmigen Banditen sind ja allgemein bekannt.
Es drehen sich drei Walzen mit Zahlen. Sind zwei Zahlen gleich, gibt es dreißig Pfennig als Gewinn, bei drei gleichen Zahlen besteht die Chance, in einer Sonderausspielung zwischen zehn Mark und gar nix zu gewinnen. Die Sonderausspielung startet automatisch, angehalten wird sie mit der Space-Taste. Der entsprechende Betrag wird dann addiert.
Der Einsatz pro Spiel beträgt dreißig Pfennig. Nach dem Start des Programms muß der Ersteinsatz festgelegt werden.
Das Programm ist in reinem BASIC geschrieben und leicht nachzuvollziehen. Es läuft auf dem 664 und 6128. *JE* □



```
10 '*****************************  (2397)
20 '*       SLOT-MASCHINE       *  (2300)
30 '*            VON            *  (2375)
40 '*      THORSTEN STUMPF       *  (233E)
50 '*            FUER           *  (2345)
60 '*    SCHNEIDER CPC-WELT      *  (233C)
70 '*       CPC 664/6128         *  (236A)
80 '*****************************  (2322)
90 '                               (07BF)
100 INK 0,0:BORDER 0:INK 1,26:PEN
1:MODE 0:LOCATE 4,3:PRINT"STUMPF-S
OFT":LOCATE 2,5:PRINT"proudly pres
ents":LOCATE 4,12:PRINT"SLOT-MACHI
NE":CALL &BB06                     (602E)
110 INK 1,24:INK 0,1:PEN 1:BORDER
1                                  (159C)
120 DIM zu(50)                     (0F76)
130 MODE 1:INPUT"Slotmachine....Wi
eviel DM moechten Sie  einsetzen "
;geld:IF geld>20 THEN PRINT"Bitte
nicht mehr als 20 DM":CALL &BB06:G
OTO 130 ELSE IF geld<0.3 THEN PRIN
T"Jedes Spiel kostet 0.30 Pf":CALL
 &BB06:GOTO 130 ELSE GOTO 140      (B8DC)
140 CLS:PRINT"Spielregeln :":PRINT
:PRINT"1 Spiel => -30 Pf":PRINT"2
gleiche Zahlen => +30 Pf":PRINT"3
gleiche Zahlen => Ausspielung":CAL
L &BB06                            (74B6)
150 GOSUB 300                      (0966)
160 CLS#3:IF geld<0.3 THEN GOTO 27
0 ELSE FOR x=1 TO 50:LOCATE 11,10:
zu(x)=INT(RND*5):PRINT zu(x):SOUND
 1,x*5,1,15:NEXT:z1=zu(50)         (733A)
170 FOR x=1 TO 50:LOCATE 17,10:zu(
x)=INT(RND*5):PRINT zu(x):SOUND 1,
x*5,1,15:NEXT:z2=zu(50)            (595A)
180 FOR x=1 TO 50:LOCATE 14,15:zu(
x)=INT(RND*5):PRINT zu(x):SOUND 1,
x*5,1,15:NEXT:z3=zu(50)            (5953)
190 IF z1=z2 AND z2=z3 THEN GOTO 4
60                                 (222F)
200 IF z1=z2 OR z2=z3 OR z1=z3 THE
N geld=geld+0.3:GOTO 230           (45FD)
210 geld=geld-0.3                  (1B50)
220 IF geld<0.3 THEN GOTO 270 ELSE
 GOTO 230                          (1FF3)
230 PEN#1,2:LOCATE#1,12,4:PRINT#1,
USING"##.##";geld:PEN#3,2:PRINT#3,
"NOCH MAL ?"                       (3FD5)
240 IF geld>99.99 THEN 510         (187C)
250 ON INSTR("?Jj¡Nn",INKEY$)+1 GOT
O 250,250,160,160,260,260          (2ECE)
260 MODE 1:PEN 1:PRINT"Ihr letztes
 Guthaben :";USING"##.##";geld:CAL
L &BB06:/BASIC                     (4221)
270 MODE 1:PEN 1:PRINT"Ihr Guthabe
n sank unter 30 Pf => E N D E":PRI
NT:PRINT"Moechten Sie ein neues Sp
iel beginnen ?"                    (63EA)
280 ON INSTR("?Jj¡Nn",INKEY$)+1 GOT
O 280,280,290,290,260,260          (2E1A)
290 RUN                            (0670)
300 MODE 1                         (0701)
310 REM   SPIELFELD                (1237)
320 INK 0,0:BORDER 0:INK 1,2:INK 2
,26:INK 3,6                        (1C79)
330 WINDOW#1,8,33,1,25:PAPER#1,1:C
LS#1:WINDOW#2,14,27,2,2:PAPER#2,3:
CLS#2:WINDOW#3,9,19,22,22:PAPER#3,
3:CLS#3:PEN 2                      (500B)
340 PEN#2,2:LOCATE#2,2,1:PRINT#2,"
SLOT-MACHINE"                      (252A)
350 PEN 2:LOCATE 1,6:PRINT CHR$(16
```

```
4):PRINT:PRINT"S":PRINT"T":PRINT"U
":PRINT"M":PRINT"P":PRINT"F":PRINT
" ":PRINT"S":PRINT"O":PRINT"F":PRI
NT"T"                                    <4D0D>
360 PEN#1,2:LOCATE#1,2,4:PRINT#1,C
HR$(24);"GUTHABEN :";CHR$(24)            <31B4>
370 GRAPHICS PEN 2:ORIGIN 111,0:DR
AWR 0,399:DRAWR 416,0:DRAWR 0,-399
:DRAWR-416,0                             <2CA4>
380 ORIGIN 127,47:DRAWR 0,18:DRAWR
 178,0:DRAWR-0,-18:DRAWR-178,0           <2605>
390 ORIGIN 207,367:DRAWR 0,18:DRAW
R 227,0:DRAWR 0,-18:DRAWR-227,0          <26F2>
400 ORIGIN 159,239:DRAWR 0,18:DRAW
R 50,0:DRAWR 0,-18:DRAWR-50,0:ORIG
IN 255,239:DRAWR 0,18:DRAWR 50,0:D
RAWR 0,-18:DRAWR-50,0:ORIGIN 207,1
59:DRAWR 0,18:DRAWR 50,0:DRAWR 0,-
18:DRAWR-50,0                            <678C>
410 WINDOW#4,27,30,20,23:PAPER#4,3
:CLS#4:WINDOW#5,27,30,15,18:PAPER#
5,3:CLS#5:WINDOW#6,27,30,10,13:PAP
ER#6,3:CLS#6:WINDOW#7,27,30,5,8:PA
PER#7,3:CLS#7                            <6A27>
420 ORIGIN 415,31:DRAWR 0,66:DRAWR
 66,0:DRAWR 0,-66:DRAWR-66,0:ORIGI
N 415,110:DRAWR 0,66:DRAWR 66,0:DR
AWR 0,-66:DRAWR-66,0:ORIGIN 415,19
0:DRAWR 0,66:DRAWR 66,0:DRAWR 0,-6
6:DRAWR-66,0:ORIGIN 415,270:DRAWR
0,66:DRAWR 66,0:DRAWR 0,-66:DRAWR-
66,0                                     <8D0F>
430 LOCATE#1,25,22:PRINT#1,"0":LOC
ATE#1,25,17:PRINT#1,"1":LOCATE#1,2
5,12:PRINT#1,"5":LOCATE#1,25,7:PRI
NT#1,"0":LOCATE#1,25,6:PRINT#1,"1"
                                         <5C37>
440 RETURN                               <0697>
450 REM  AUSSPIELUNG                     <1322>
460 CLS#3:PRINT#3,"<SPACE>":FOR x=
1 TO 30:FOR y=4 TO 7:PAPER#y,2:CLS
#y:SOUND 1,200,7,15:PAPER#y,3:CLS#
y                                        <56A8>
470 IF INKEY$=" "THEN 490 ELSE NEX
T y:NEXT x                               <1D53>
480 IF y=8 THEN y=7:PAPER#y,3:CLS#
y:y=4:PAPER#y,2:CLS#y                    <3A3C>
490 INK 3,6:PAPER#y,2:IF y=4 THEN
geld=geld ELSE IF y=5 THEN geld=ge
ld+1 ELSE IF y=6 THEN geld=geld+5
ELSE IF y=7 THEN geld=geld+10            <7C43>
500 PAPER#y,2:CLS#y:FOR x=1 TO 100
0:NEXT:PAPER#y,3:CLS#y:CLS#3:GOTO
230                                      <3B1D>
510 MODE 1:PRINT"Sie haben die Gre
nze von 99.99 DM er-   reicht !":P
RINT:PRINT"Moechten Sie ein neues
Spiel beginnen ?"                        <6866>
520 ON INSTR("?JjNn",INKEY$)+1 GOT
O 520,520,290,290,260,260                <2E7F>
```

Mau-Mau

Wer kennt dieses Spiel nicht aus langweiligen Schulstunden? Wer hat es noch nie unter der Schulbank gespielt? Gemeint ist das altbekannte Mau-Mau-Spiel. Hier ist eine Umsetzung für alle CPC-Typen von unserem Leser **Wolfgang Hesseler**:

Die Spielregeln sind ganz einfach. Jeder Spieler erhält am Anfang sechs Karten. Eine Karte liegt gesondert mit der Bildseite nach oben. Nun müssen die Spieler abwechselnd Karten ablegen, die entweder mit dem Wert oder der Farbe der Einzelkarte übereinstimmen. Buben (Bauern) dürfen immer abgelegt werden, der Spieler kann danach die Farbe bestimmen, die der andere als nächstes ablegen muß.

Kann keine Karte abgelegt werden, muß eine vom Haufen genommen werden. Hat jemand ein As gelegt, kann er gleich nochmals eine Karte loswerden. Legt ein Spieler eine Sieben, muß der nächste Spieler zwei Karten aufnehmen. Wer als letzte Karte einen Buben gelegt hat, erhält in der Spielauswertung einen Extrapunkt.

Das Programm akzeptiert keine falschen Eingaben, denn schummeln kann der CPC nicht.

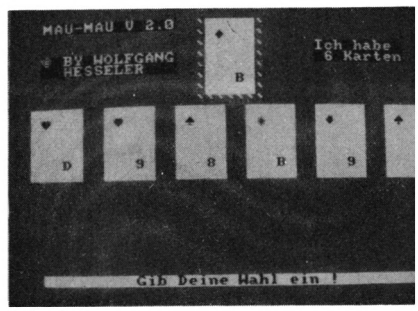
Bei der abzulegenden Karte wird zuerst die Farbe eingegeben:

K = Kreuz
P = Pik
H = Herz
C = Karo

Danach folgt der Kartenwert:

A = As
K = König
D = Dame
B = Bube
1 = 10
9 = 9
8 = 8
7 = 7

Das Programm läuft auf allen CPCs. *JE*□



Mau-Mau

```
10 '*****************************  (2397)
11 '*          MAU-MAU            *  (2370)
12 '*            VON              *  (2351)
13 '*     WOLFGANG HESSELER       *  (23AA)
14 '*           FUER               *  (236E)
15 '*     SCHNEIDER CPC-WELT       *  (23E3)
16 '*      CPC 464/664/6128   JE*  (2370)
17 '*****************************  (23A5)
20 MODE 1                          (07D1)
30 INK 0,0:INK 1,2:INK 2,26:INK 3,
6                                  (1910)
40 SYMBOL AFTER 229:SYMBOL 230,0,8
,42,28,127,28,42,8                 (2217)
50 SPEED KEY 255,255               (0CB1)
60 DATA KARO,HERZ,PIK,KREUZ        (1A45)
70 FOR i=1 TO 4:READ na$(i):NEXT   (1D05)
80 DIM u$(32),e(32),n(2),p$(12)    (28E7)
90 DEF FNf(t)=ASC(u$(t))+162       (2317)
100 DEF FNv$(t)=MID$(u$(t),2,2)    (232F)
110 DATA A7,B7,C7,D7,A8,B8,C8,D8,A
9,B9,C9,D9,AB,BB,CB,DB,AD,BD,CD,DD
,AK,BK,CK,DK,A10,B10,C10,D10,AA,BA
,CA,DA                             (6ACF)
120 RANDOMIZE TIME                 (082C)
130 RESTORE 110                    (0953)
140 FOR i=1 TO 32                  (0F5A)
150 j=INT(32*RND)+1                (15B2)
160 IF u$(j)<>""THEN 150           (17B1)
170 READ u$(j)                     (10ED)
180 NEXT                           (062A)
190 IF a$="J"THEN 300              (121A)
200 l=INT(0.5-RND)                 (17A7)
210 PAPER 2:PEN 0:BORDER 11:CLS    (103C)
220 LOCATE 13,1:PRINT"***";:PEN 3:
PRINT"MAU-MAU";:PEN 0:PRINT"***"   (2BE5)
230 LOCATE 1,3:PRINT"Die Kartenwah
l geschieht wie folgt:":PRINT"1.Di
e Abkuerzung der Farbe eingeben:":
PEN 3:PRINT"   "CHR$(227);:PEN 0:P
RINT"=C      ";:PEN 3:PRINT CHR$(2
28);:PEN 0:PRINT"=H      "CHR$(229
)"=P      "CHR$(230)"=K"           (AE20)
240 PRINT:PRINT"2.Dann gibt man di
e 2.Spezifikation ein   (d.h. 7,8,
9,1 fuer 10,B,D,K,A)"              (521D)
250 PRINT:PRINT"Die Eingaben muess
en mit ENTER abge-    schlossen we
rden. Falsche Eingaben      koenne
n mit CLR geloescht werden."       (7B9A)
260 PRINT"Mit SPACE zieht man Kart
en.":PRINT:PRINT"Achtung:":PRINT"B
ei der 7 sind zwei Karten zu ziehe
n . Beim AS ist man nocheinmal an
 der Reihe Beim Bauer darf man ein
e Farbe waehlen":PRINT:PRINT"Weite
re Instruktionen erhaelt man"      (D238)
270 PRINT"waehrend des Spiels vom
Computer."                         (29F1)
280 LOCATE 13,25:PRINT"Weiter mit
SPACE"                             (1F22)
290 IF INKEY$<>" "THEN 290         (1057)
300 PAPER 1:CLS:BORDER 2           (0C8D)
310 p=-(FNv$(13)="B"):w=13         (2288)
320 PAPER 3:PEN 2:LOCATE 2,1:PRINT
"MAU-MAU V 2.0"                    (2057)
330 LOCATE 2,4:PRINT CHR$(164);" B
Y WOLFGANG":LOCATE 2,5:PRINT"  HES
SELER   ":PAPER 2                  (39EA)
340 FOR i=1 TO 36 STEP 7:FOR j=9 T
O 15:LOCATE i,j:PRINT"     ":NEXT
j,i                                (3B4C)
350 FOR i=1 TO 7:LOCATE 18,i:PRINT
"     ":NEXT                       (2209)
360 FOR i=2 TO 37 STEP 7:IF FNf(I\
7+1)<229 THEN PEN 3 ELSE PEN 0     (2C5D)
370 LOCATE i,10:PRINT CHR$(FNf(i\7
+1))                               (2249)
380 PEN 0:LOCATE i+2,14:PRINT FNv$
(i\7+1):NEXT                       (2532)
390 IF FNf(13)<229 THEN PEN 3      (1535)
400 LOCATE 19,2:PRINT CHR$(FNf(13)
)                                  (19E1)
410 PEN 0:LOCATE 21,6:PRINT FNv$(1
3)                                 (18A5)
420 PAPER 1:PEN 2:FOR i=1 TO 8:LOC
ATE 17,i:PRINT"`":LOCATE 23,i:PRIN
T"`":NEXT:LOCATE 17,8:PRINT"``````
":PAPER 2:PEN 0                    (4881)
430 n(1)=6:n(2)=6                  (18FB)
440 l=NOT l                        (0F0D)
450 LOCATE 13,25:IF l XOR FNv$(13)
="A"THEN PRINT"Du faengst an !":GO
TO 460 ELSE PRINT"Ich fange an !"  (4A46)
460 LOCATE 29,3:PAPER 3:PEN 2:PRIN
T"Ich habe ":LOCATE 29,4:PRINT" 6
Karten":PAPER 2:PEN 0              (36CC)
470 FOR i=1 TO 6:p$(i)=u$(i):p$(i+
6)="":e(i)=1:e(i+6)=2:NEXT         (5264)
480 FOR i=14 TO 32:e(i)=3:NEXT     (1FD9)
490 e(13)=4                        (0F70)
500 FOR i=1 TO 2500:NEXT           (120F)
510 LOCATE 2,25:PRINT SPACE$(38)   (1267)
520 IF l XOR FNv$(13)="A"THEN 560
ELSE 1400                          (21CD)
530 REM SPIELER LEGT AB            (16A6)
540 LOCATE 2,25:PRINT SPACE$(38)   (12A4)
550 IF FNv$(W)="A"THEN SOUND 1,202
5,20:LOCATE 5,25:PRINT"Ich bin noc
heinmal an der Reihe":FOR i=1 TO 1
400:NEXT:GOTO 1410                 (5B2B)
560 IF FNv$(w)="7"THEN SOUND 1,202
5,50:LOCATE 6,25:PRINT"Du musst zw
ei Karten ziehen !":FOR i=1 TO 600
:NEXT:GOSUB 1210:GOSUB 1210:FOR i=
1 TO 500:NEXT                      (6C33)
570 LOCATE 2,25:PRINT SPACE$(38)   (12E0)
580 SOUND 2,142,12,12:LOCATE 11,25
:PRINT"Gib Deine Wahl ein !"       (2F75)
590 a$=INKEY$:IF a$=""THEN 590     (19AD)
600 a$=UPPER$(a$)                  (12D8)
610 IF a$=" "THEN 1080             (12BA)
620 IF INSTR("CHPK",a$)=0 THEN 590 (1BC2)
```

```
630 g=INSTR("CHPK",a$)+64                    <1C5B>
640 SOUND 1,142,5                            <0CFB>
650 LOCATE 32,25:PRINT na$(g-64);            <1CF8>
660 a$=INKEY$:IF a$=""THEN 660               <1962>
670 a$=UPPER$(a$)                            <1265>
680 IF ASC(a$)=16 THEN 570                   <15AF>
690 IF INSTR("7891BDKA",a$)=0 THEN
 660                                         <1F76>
700 IF a$="1"THEN b$="10"ELSE b$=a
$                                            <23C0>
710 SOUND 1,142,5                            <0C86>
720 PRINT"-"b$                               <0D0A>
730 a$=INKEY$:IF a$=""THEN 730               <1921>
740 a$=UPPER$(a$)                            <12F1>
750 IF ASC(a$)=16 THEN 570                   <153A>
760 IF ASC(a$)<>13 THEN 730                  <154D>
770 b$=CHR$(g)+b$                            <17E4>
780 FOR g=1 TO 32:IF u$(g)<>b$THEN
 NEXT:GOTO 1330                              <2722>
790 IF e(g)<>1 THEN 1330                     <1677>
800 IF p=1 THEN 860                          <101D>
810 IF FNf(g)=p THEN 860                     <1A4D>
820 IF FNv$(g)="B"THEN 860                   <19F6>
830 IF p THEN 1330                           <0E00>
840 IF FNf(g)=FNf(w)THEN 870                 <21EB>
850 IF FNv$(g)<>FNv$(w)THEN 1330             <211C>
860 p=0                                      <0BF0>
870 LOCATE 11,25:PRINT SPACE$(29)            <130E>
880 e(w)=5:e(g)=4:w=g                        <287C>
890 FOR h=1 TO 12:IF p$(h)=u$(g)TH
EN 900 ELSE NEXT                             <2DBC>
900 j=h*7-6+(h)6)*42                         <1EEE>
910 FOR i=0 TO 6:PEN 1:LOCATE j,9-
(h)6)*8+i:PRINT CHR$(24);"      ";C
HR$(24):NEXT:PEN 0                           <4571>
920 p$(h)=""                                 <12D6>
930 LOCATE 19,2:IF FNf(g)<229 THEN
 PEN 3                                       <1D42>
940 PRINT CHR$(FNf(g)):PEN 0                 <1820>
950 LOCATE 21,6:PRINT"  ";CHR$(8);
CHR$(8);FNv$(g)                              <28A4>
960 n(1)=n(1)-1                              <1631>
970 IF n(1)=0 THEN 2060                      <13CA>
980 IF FNv$(w)<>"B"THEN 1370                 <19A8>
990 LOCATE 2,25:PRINT SPACE$(38):S
OUND 2,142,12,12:LOCATE 8,25:PRINT
"Welche Farbe waehlst du?"                   <404F>
1000 a$=INKEY$:IF a$=""THEN 1000             <19B2>
1010 a$=UPPER$(a$)                           <120D>
1020 IF INSTR("CHPK",a$)=0 THEN 10
00                                           <1BBD>
1030 SOUND 1,142,5                           <0C09>
1040 p=INSTR("CHPK",a$)+226                  <1C7B>
1050 LOCATE 33,25:PRINT na$(p-226)
                                             <1B23>
1060 a$=INKEY$:IF a$=""THEN 1060 E
LSE IF ASC(a$)<>13 AND ASC(a$)<>16
 THEN 1060                                   <37D2>
1070 IF ASC(a$)=16 THEN 990 ELSE 1
370                                          <1AEB>
1080 REM SPIELER KANN NICHT ABLEGE
N                                            <21E2>
1090 LOCATE 2,25:PRINT SPACE$(38)            <12F2>
1100 o=0                                     <0BC1>
1110 FOR i=1 TO 32:IF e(i)=1 THEN
1140                                         <219C>
1120 NEXT                                    <0687>
1130 IF o THEN 1410 ELSE GOSUB 121
0:o=1:GOTO 1110                              <2099>
1140 IF p=1 THEN 1200                        <1067>
1150 IF FNf(i)=p THEN 1200                   <1AB3>
1160 IF FNv$(i)="B"THEN 1200                 <194F>
1170 IF p THEN 1120                          <0E16>
1180 IF FNf(i)=FNf(w)THEN 1200               <21F5>
1190 IF FNv$(i)<>FNv$(w)THEN 1120            <2141>
1200 IF o THEN 570 ELSE 1330                 <13AC>
1210 REM SPIELER KARTE DAZUBEKOMME
N                                            <216D>
1220 LOCATE 2,25:PRINT SPACE$(38):
LOCATE 15,25:PRINT"KARTE ZIEHEN":F
OR i=1 TO 200:NEXT                           <3643>
1230 FOR g=1 TO 32:IF e(g)=3 THEN
1260                                         <21B3>
1240 NEXT                                    <0676>
1250 GOSUB 1870:GOTO 1230                    <0EF9>
1260 FOR h=1 TO 12:IF p$(h)=""THEN
 1270 ELSE NEXT:GOTO 1990                    <2AA0>
1270 j=h*7-6+(h)6)*42                        <1ED2>
1280 FOR i=0 TO 6:LOCATE j,9-(h)6)
*8+i:PRINT"      ":NEXT                      <31BE>
1290 LOCATE j+1,10-(h)6)*8:IF FNf(
g)<229 THEN PEN 3                            <2D2F>
1300 PRINT CHR$(FNf(g)):PEN 0                <18F2>
1310 LOCATE j+3,14-(h)6)*8:PRINT F
Nv$(g)                                       <2781>
1320 e(g)=1:p$(h)=u$(g):n(1)=n(1)+
1:RETURN                                     <3B6D>
1330 REM FEHLER                              <0DC2>
1340 LOCATE 2,25:PRINT SPACE$(38)            <12E7>
1350 SOUND 1,451,50:LOCATE 9,25:PR
INT"F E H L E I N G A B E !"                 <2FE5>
1360 FOR i=1 TO 1600:NEXT:GOTO 570
                                             <173E>
1370 REM COMPUTER AN DER REIHE               <1C23>
1380 LOCATE 2,25:PRINT SPACE$(38)            <1238>
1390 IF FNv$(w)="A"THEN SOUND 1,11
9,30:LOCATE 5,25:PRINT"Du bist noc
heinmal an der Reihe":FOR i=1 TO 6
50:NEXT:GOTO 570                             <5AA4>
1400 IF FNv$(w)="7"THEN SOUND 1,11
9,30:LOCATE 9,25:PRINT"Ich muss 2
Karten ziehen":GOSUB 1800:GOSUB 18
00:FOR i=1 TO 800:NEXT                       <58B4>
1410 o=0:LOCATE 2,25:PRINT SPACE$(
38)                                          <1961>
1420 FOR g=1 TO 3                            <0EDE>
1430 FOR h=1 TO 32                           <0FF4>
1440 IF e(h)<>2 THEN 1480                    <167B>
1450 ON g GOTO 1460,1470,1510                <1620>
1460 IF FNv$(h)="7"OR FNv$(h)="A"T
HEN 1520 ELSE 1480                           <2E10>
```

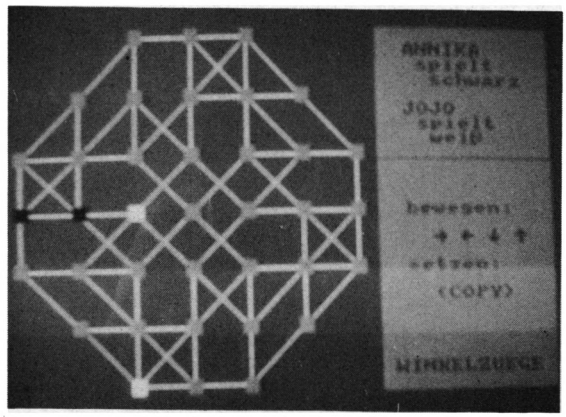
Mau-Mau

```
1470 IF FNv$(h)<>"B"THEN 1520          <19C0>
1480 NEXT h                            <0A01>
1490 NEXT g                            <0A12>
1500 IF o THEN LOCATE 6,25:PRINT"I
ch kann wieder nicht ablegen":FOR
i=1 TO 2200:NEXT:GOTO 570 ELSE LOC
ATE 10,25:PRINT"Ich kann nicht abl
egen":GOSUB 1800:o=1:FOR i=1 TO 22
00:NEXT:LOCATE 10,25:PRINT SPACE$(
30):GOTO 1420                          <948A>
1510 IF FNv$(h)="B"THEN 1610           <19DF>
1520 IF p=1 THEN 1570                  <10FC>
1530 IF FNf(h)=p THEN 1570             <1A37>
1540 IF p THEN 1480                    <0E3C>
1550 IF FNf(h)=FNf(w)THEN 1570         <216E>
1560 IF FNv$(h)<>FNv$(w)THEN 1480      <2167>
1570 IF FNv$(h)<>"A"OR g>1 THEN 16
10                                     <20AB>
1580 FOR i=1 TO 32:IF e(i)=2 THEN
IF FNf(h)=FNf(i)OR FNv$(i)="A"THEN
 1600                                  <4A4F>
1590 NEXT:GOTO 1480                    <0B33>
1600 IF i=h THEN 1590                  <13BA>
1610 REM COMPUTER LEGT KARTE AB        <1DE5>
1620 p=0                               <0BE3>
1630 LOCATE 19,2:IF FNf(h)<229 THE
N PEN 3                                <1DB6>
1640 PRINT CHR$(FNf(h)):PEN 0          <1825>
1650 LOCATE 21,6:PRINT"  ";CHR$(8)
;CHR$(8);FNv$(h)                       <2827>
1660 e(h)=4:e(w)=5:w=h                 <2892>
1670 n(2)=n(2)-1                       <1601>
1680 PAPER 3:PEN 2:LOCATE 29,4:PRI
NT USING"##";n(2):LOCATE 37,4:IF n
(2)=1 THEN PRINT" "ELSE PRINT"n"       <3BAC>
1690 PAPER 2:PEN 0                     <0A10>
1700 IF n(2)=0 THEN 1990               <133B>
1710 IF FNv$(h)<>"B"THEN 530           <19A4>
1720 I=1                               <0B3C>
1730 IF e(i)<>2 OR FNv$(i)="B"THEN
 1750                                  <266F>
1740 a(FNf(i)-226)=a(FNf(i)-226)+1
                                       <30FC>
1750 I=I+1:IF I<=32 THEN 1730          <1D7D>
1760 g=MAX(a(1),a(2),a(3),a(4))        <2D4B>
1770 FOR i=1 TO 4:IF a(i)=g THEN 1
780 ELSE NEXT                          <2633>
1780 P=I+226                           <1162>
1790 LOCATE 9,25:PRINT"Ich waehle
"na$(i)"  <TASTE>":CALL &BB06:FOR
i=1 TO 4:a(i)=0:NEXT:GOTO 530          <52F4>
1800 REM COMPUTER KARTE DAZUBEKOMM
EN                                     <227C>
1810 FOR i=1 TO 32:IF e(i)=3 THEN
1820 ELSE NEXT:GOSUB 1870:GOTO 181
0                                      <2E86>
1820 e(i)=2                            <1127>
1830 n(2)=n(2)+1                       <163A>
1840 IF n(2)=13 THEN 2060              <14A8>
1850 PAPER 3:PEN 2:LOCATE 29,4:PRI
NT USING"##";n(2):LOCATE 37,4:PRIN
T"n":PAPER 2:PEN 0                     <30FE>
1860 RETURN                            <06B6>
1870 REM NEU MISCHEN                   <1232>
1880 LOCATE 2,25:PRINT SPACE$(38):
LOCATE 16,25:PRINT"NEU MISCHEN"        <287D>
1890 h=31-n(1)-n(2)                    <1CBA>
1900 FOR i=1 TO 32                     <0F21>
1910 IF e(i)<>5 THEN 1970              <1643>
1920 o=INT(32*RND)+1                   <15D0>
1930 IF e(o)<>5 THEN 1920              <16ED>
1940 b$=u$(i):u$(i)=u$(o):u$(o)=b$
                                       <3AF1>
1950 e(i)=3:e(o)=3                     <1EF0>
1960 h=h-2                             <10D4>
1970 NEXT                              <062D>
1980 RETURN                            <06A5>
1990 REM VERLOREN                      <0FC1>
2000 LOCATE 8,25:IF FNv$(w)="B"THE
N PRINT"MAU-MAU";:ve=ve+2 ELSE PRI
NT"MAU";:ve=ve+1                       <4C50>
2010 PRINT" - Du hast verloren !"      <1D9B>
2020 ENT 1,120,9,7                     <0E76>
2030 SOUND 1,380,440,12,,1             <15F1>
2040 FOR i=1 TO 5000:NEXT              <123A>
2050 GOTO 2120                         <095C>
2060 REM GEWONNEN                      <0FE3>
2070 LOCATE 8,25:IF FNv$(w)="B"THE
N PRINT"MAU-MAU";:si=si+2 ELSE PRI
NT"MAU";:si=si+1                       <4CD9>
2080 PRINT" - Du hast gewonnen !"      <1D11>
2090 ENT 1,42,-5,5                     <0FF3>
2100 SOUND 1,239,200,12,,1             <1301>
2110 FOR i=1 TO 3000:NEXT              <1249>
2120 INK 3,2:PAPER 3:PEN 2:CLS:BOR
DER 25                                 <157F>
2130 sp=sp+1                           <127F>
2140 LOCATE 1,3:IF si>ve THEN PRIN
T" Gratuliere,";                       <278D>
2150 PRINT" Du hast nach";:IF sp=1
 THEN PRINT" einem Spiel"ELSE PRIN
T sp;"Spielen"                         <4126>
2160 PRINT:IF si=1 THEN PRINT" ein
en Gewinnpunkt";ELSE PRINT si;"Gew
innpunkte";                            <3EEF>
2170 PRINT" und";:IF ve=1 THEN PRI
NT" einen Verlustpunkt"ELSE PRINT
ve;"Verlustpunkte"                     <45E5>
2180 LOCATE 6,23:PRINT"... noch ei
n Spielchen? (J/N)"                    <2B03>
2190 a$=UPPER$(INKEY$):IF a$=""THE
N 2190                                 <1DF8>
2200 IF a$="J"THEN FOR i=1 TO 32:u
$(i)="":NEXT:INK 3,6:PAPER 1:BORDE
R 2:CLS:PEN 0:GOTO 120                 <3E2A>
2210 IF a$="N"THEN PAPER 0:CLS:CAL
L &BC02:PEN 1:CALL &BB00 ELSE 2190
                                       <252D>
```

Winkelzüge

Dieses kleine Spiel unserer Autorin **Friederike Böhler** dient der Entspannung. Genau das Richtige, um sich zwischendurch einmal am Bildschirm zu erholen. Trotzdem ist es recht interessant und fesselt die Spieler immer wieder. Sein Sinn besteht darin, drei laufende Punkte in einem komplexen Gitternetz mit der eigenen Farbe zu kennzeichnen. Da abwechselnd gezogen wird, ist dies nicht ganz einfach. Wer zuerst drei „Mühlen" gesetzt hat, ist Sieger.
Das Spiel kann nur zu zweit gespielt werden und läuft auf den CPC-Typen 664 und 6128.

JE □




```
10 '*****************************  (2397)
20 '*       WINKELZUEGE         *  (236D)
30 '*           VON             *  (23B1)
40 '*    FRIEDERIKE BOEHLER     *  (2351)
50 '*           FUER            *  (2377)
60 '*    SCHNEIDER CPC-WELT     *  (23D8)
65 '*      CPC  664/6128      JE*  (23C1)
66 '*****************************  (2306)
67 '                                (0791)
70 '*****************************  (230E)
80 '***    INITIALISIERUNG     ***  (23B1)
90 '*****************************  (2336)
100 CLEAR:WINDOW#1,1,27,1,25        (142E)
110 WINDOW#0,28,40,1,25             (13E9)
120 INK 0,0:INK 1,26:INK 2,16:INK
3,9                                 (1A34)
130 PAPER#1,3:PEN#1,2,0:BORDER 9:P
APER 2:PEN 3                        (1BB5)
140 CLS:CLS#1                       (0AD9)
150 ENV 1,2,4,5                     (0DBF)
160 DIM x(7),y(7),fld(27,25),sp(2)
                                    (2CEE)
170 zuf=RND:IF zuf<0.5 THEN sp=1 E
LSE sp=2                            (2EF9)
180 RESTORE 1420:FOR i=1 TO 7:READ
 x(i),y(i):NEXT                     (2C15)
190 FOR i=1 TO 7:FOR j=1 TO 7:fld(
x(i),y(j))=1:NEXT:NEXT              (3C0F)
200 FOR i=1 TO 7:FOR j=1 TO 7:IF(i
=1 OR i=7)AND(j=1 OR j=2 OR j=6 OR
 j=7)THEN fld(x(i),y(j))=0          (679F)
210 IF(i=2 OR i=6)AND(j=1 OR j=7)T
HEN fld(x(i),y(j))=0                (459B)
220 NEXT:NEXT                       (0889)
230 p1=0:p2=0                       (1499)
240 '*****************************  (2363)
250 '***   SPIELFELDAUFBAU     ***  (23C4)
260 '*****************************  (238C)
270 MOVE 151,8:DRAW 279,8:DRAW 407
,135:DRAW 407,263:DRAW 279,391:DRA
W 151,391:DRAW 23,263:DRAW 23,135:
DRAW 151,8:DRAW 279,135:DRAW 151,2
63:DRAW 279,391:DRAW 279,263:DRAW
407,263                             (73DE)
280 MOVE 151,391:DRAW 151,263:DRAW
 23,263:DRAW 151,135:DRAW 279,263:
DRAW 407,135:DRAW 279,135:DRAW 279
,8                                  (4324)
290 MOVE 279,391:DRAW 151,263:DRAW
 279,135:DRAW 151,8:DRAW 151,135:D
RAW 23,135                          (31AF)
300 MOVE 23,135:DRAW 151,135:DRAW
151,8                               (1888)
310 MOVE 215,8:DRAW 87,135:DRAW 21
5,263                               (1943)
320 MOVE 23,199:DRAW 151,327:DRAW
279,199                             (1BDE)
330 MOVE 215,263:DRAW 215,391:DRAW
 343,263:DRAW 215,135:DRAW 215,8    (2A73)
340 MOVE 279,199:DRAW 407,199:DRAW
 279,71:DRAW 151,199:DRAW 23,199    (2AC8)
350 MOVE 87,199:DRAW 87,327:MOVE 2
15,327:DRAW 343,327:MOVE 343,71:DR
AW 343,199:MOVE 87,71:DRAW 215,71  (4216)
360 FOR i=1 TO 7:FOR j=1 TO 7       (18B0)
370 IF fld(x(i),y(j))=1 THEN LOCAT
E#1,x(i),y(j):PRINT#1,CHR$(143);EL
SE 380                              (5039)
380 NEXT:NEXT                       (08C9)
390 MOVE 455,215:DRAW 615,215:DRAW
 455,375:DRAW 615,375:MOVE 455,375
:DRAW 455,215:DRAW 615,375:DRAW 61
5,215                               (4840)
400 LOCATE 2,7:PRINT"WINKELZUEGE";
                                    (1907)
410 LOCATE 2,2:PRINT CHR$(143);:LO
```

Winkelzüge

```
CATE 2,12:PRINT CHR$(143);:LOCATE
12,2:PRINT CHR$(143);:LOCATE 12,12
:PRINT CHR$(143);                    <406D>
420 RESTORE 430:FOR i=1 TO 11:READ
 a$:LOCATE 7,i+1:PRINT a$;:NEXT:GO
TO 440                               <3283>
430 DATA W,I,N,K,E,L,Z,U,E,G,E       <1C86>
440 LOCATE 6,15:PRINT"FUER";:LOCAT
E 7,16:PRINT"2";:LOCATE 4,17:PRINT
"SPIELER";                           <31A6>
450 LOCATE 3,20:PRINT CHR$(164);"
1987 by";                            <1EE0>
460 LOCATE 3,22:PRINT"F-R-I-B-O";    <1870>
470 GOSUB 1440                       <0993>
480 '*************************       <222E>
490 '**   SPIELVORBEREITUNG   **     <225D>
500 '*************************       <2256>
510 CLS:LOCATE 3,2:PRINT"ANLEITUNG
";                                   <19BD>
520 LOCATE 1,3:PRINT STRING$(13,"_
");                                  <1606>
530 LOCATE 5,5:PRINT"Jeder";:LOCAT
E 4,6:PRINT"Spieler";:LOCATE 3,7:P
RINT"versucht,";:LOCATE 4,10:PRINT
"DREIMAL";:LOCATE 2,12:PRINT"DREI
FELDER";:LOCATE 4,14:PRINT"ENTLANG
";                                   <716A>
540 LOCATE 2,16:PRINT"EINER LINIE"
;:LOCATE 2,19:PRINT"mit  seiner";:
LOCATE 3,20:PRINT"Farbe  zu";:LOCA
TE 3,21:PRINT"besetzen!";            <5832>
550 GOSUB 1440                       <0935>
560 CLS:LOCATE 3,2:PRINT"NAMEN DER
";:LOCATE 4,4:PRINT"SPIELER";:LOCA
TE 3,6:PRINT"EINGEBEN:";:LOCATE 1,
7:PRINT STRING$(13,"_");             <4F22>
570 CLEAR INPUT:LOCATE 2,10:INPUT
na$:sp2$=UPPER$(na$):LOCATE 2,13:I
NPUT na$:sp1$=UPPER$(na$)            <43FD>
580 GOSUB 1440                       <0971>
590 IF sp=1 THEN farb1$="wei"+CHR$
(177)ELSE farb1$="schwarz"           <37E9>
600 IF sp=2 THEN farb2$="wei"+CHR$
(177)ELSE farb2$="schwarz"           <37E5>
610 GOSUB 1440                       <09AD>
620 CLS:LOCATE 3,2:PRINT sp1$;:LOC
ATE 4,3:PRINT"spielt";:LOCATE 5,4:
PRINT farb1$;                        <34C4>
630 LOCATE 3,6:PRINT sp2$;:LOCATE
4,7:PRINT"spielt";:LOCATE 5,8:PRIN
T farb2$;                            <3222>
640 LOCATE 1,9:PRINT STRING$(13,"_
")                                   <1508>
650 LOCATE 3,13:PRINT"bewegen:";:L
OCATE 5,15:PRINT CHR$(243);CHR$(32
);CHR$(242);CHR$(32);CHR$(241);CHR
$(32);CHR$(240);                     <5052>
660 LOCATE 3,17:PRINT"setzen:";:LO
CATE 5,19:PRINT"<COPY>";             <27DF>
670 GOSUB 1440                       <0924>
680 LOCATE 2,24:PRINT"WINKELZUEGE"   <19F5>
690 GOTO 1150                        <0957>
700 '*************************       <21DD>
710 '***      BEWERTUNG      ***     <21CF>
720 '*************************       <2104>
730 pkt1=0:pkt2=0                    <18DE>
740 RESTORE 1460                     <0956>
750 FOR punkt=1 TO 38:READ i,j,k,l
,m,n                                 <32F7>
760 IF(farb1$="wei"+CHR$(177)AND f
arb2$="schwarz")THEN 770 ELSE 800    <3929>
770 IF(fld(x(i),y(j))=7 AND fld(x(
k),y(l))=7 AND fld(x(m),y(n))=7)TH
EN pkt1=pkt1+1:IF p1<pkt1 THEN FOR
 i=1 TO pkt1:SOUND 1,60:SOUND 1,80
:SOUND 1,95:SOUND 1,119,60:NEXT:p1
=pkt1                                <C3F3>
780 IF(fld(x(i),y(j))=2 AND fld(x(
k),y(l))=2 AND fld(x(m),y(n))=2)TH
EN pkt2=pkt2+1:IF p2<pkt2 THEN FOR
 i=1 TO pkt2:SOUND 1,119:SOUND 1,9
5:SOUND 1,80:SOUND 1,60,60:NEXT:p2
=pkt2                                <C37A>
790 GOTO 820                         <09D2>
800 IF(fld(x(i),y(j))=2 AND fld(x(
k),y(l))=2 AND fld(x(m),y(n))=2)TH
EN pkt1=pkt1+1:IF p1<pkt1 THEN FOR
 i=1 TO pkt1:SOUND 1,60:SOUND 1,80
:SOUND 1,95:SOUND 1,119,60:NEXT:p1
=pkt1                                <C36E>
810 IF(fld(x(i),y(j))=7 AND fld(x(
k),y(l))=7 AND fld(x(m),y(n))=7)TH
EN pkt2=pkt2+1:IF p2<pkt2 THEN FOR
 i=1 TO pkt2:SOUND 1,119:SOUND 1,9
5:SOUND 1,80:SOUND 1,60,60:NEXT:p2
=pkt2                                <C33F>
820 NEXT                             <062E>
830 IF(pkt1=3 OR pkt2=3)THEN 840 E
LSE RETURN                           <22AA>
840 CLS:LOCATE 3,3:PRINT sp1$;:LOC
ATE 6,4:PRINT"hat";:LOCATE 3,5:PRI
NT pkt1;" Punkte";                   <3A7A>
850 LOCATE 3,7:PRINT sp2$;:LOCATE
6,8:PRINT"hat";:LOCATE 3,9:PRINT p
kt2;" Punkte";                       <3883>
860 LOCATE 1,10:PRINT STRING$(13,"
_")                                  <1672>
870 IF pkt1<pkt2 THEN 900            <198B>
880 IF pkt1>pkt2 THEN 910            <1952>
890 IF pkt1=pkt2 THEN 940            <1936>
900 sieg$=sp2$:sieg=pkt2:FOR i=1 T
O pkt2:SOUND 1,119:SOUND 1,95:SOUN
D 1,80:SOUND 1,60,60:SOUND 2,60:SO
UND 2,80:SOUND 2,95:SOUND 2,119,60
:NEXT:GOTO 920                       <708D>
910 sieg$=sp1$:sieg=pkt1:FOR i=1 T
O pkt1:SOUND 1,60:SOUND 1,80:SOUND
 1,95:SOUND 1,119,60:SOUND 2,119:S
```

```
OUND 2,95:SOUND 2,80:SOUND 2,60,60
:NEXT:GOTO 920                          (705D)
920 INK 1,26,0:PEN 1:LOCATE 2,12:P
RINT"BRAVO, ";:LOCATE 4,14:PRINT s
ieg$;"!";                               (3564)
930 LOCATE 4,16:PRINT"Du hast";:LO
CATE 3,18:PRINT"gewonnen!":PEN 3:G
OSUB 1440:GOTO 1010                     (36E7)
940 CLS:LOCATE 3,3:PRINT sp1$;:LOC
ATE 6,4:PRINT"hat";:LOCATE 3,5:PRI
NT pkt1;" Punkte";:LOCATE 3,7:PRIN
T sp2$;:LOCATE 6,8:PRINT"hat";:LOC
ATE 3,9:PRINT pkt2;" Punkte";:LOCA
TE 1,10:PRINT STRING$(13,"_")           (8040)
950 FOR i=1 TO 5:SOUND 1,159:SOUND
 1,239:NEXT:SOUND 1,159,60:SOUND 2
,239,60                                 (2E51)
960 PEN 0:LOCATE 3,13:PRINT"Leider
 hat";:PEN 1:LOCATE 2,15:PRINT"kei
n Spieler";:PEN 0:LOCATE 4,17:PRIN
T"3 Reihen";:PEN 1:LOCATE 4,19:PRI
NT"erobert!":PEN 3                      (64AD)
970 GOSUB 1440                          (097D)
980 '*************************          (210D)
990 '***    NEUES SPIEL ?   ***         (212E)
1000 '*************************         (20AF)
1010 CLS:LOCATE 2,2:FOR i=1 TO 11:
PRINT"?";:NEXT                          (1EEE)
1020 LOCATE 2,10:FOR i=1 TO 11:PRI
NT"?";:NEXT                             (1DFC)
1030 FOR i=1 TO 7:LOCATE 2,i+2:PRI
NT"?";:NEXT                             (2024)
1040 FOR i=1 TO 7:LOCATE 12,i+2:PR
INT"?";:NEXT                            (2104)
1050 LOCATE 6,4:PRINT"Ein";:LOCATE
 5,6:PRINT"neues";:LOCATE 5,8:PRIN
T"Spiel";                               (2F98)
1060 LOCATE 6,15:PRINT"j/n";            (12DF)
1070 a$=INKEY$                          (0C93)
1080 IF a$=""THEN 1070                  (11E3)
1090 IF a$="j"OR a$="J"THEN 100         (1B0C)
1100 IF a$="n"OR a$="N"THEN INK 1,
26:END                                  (1F56)
1110 GOTO 1060                          (09D7)
1120 '*************************         (20A1)
1130 '***    SPIELZUEGE     ***         (20C4)
1140 '*************************         (20C9)
1150 spiel=0                            (0F08)
1160 IF spiel>4 AND spiel<37 THEN
GOSUB 730                               (21FC)
1170 IF spiel=37 THEN GOSUB 730:GO
TO 940                                  (1B62)
1180 xc=x(1):yc=y(7):xca=0:yca=0:I
F sp=2 THEN SOUND 1,239 ELSE SOUND
 1,159                                  (4836)
1190 LOCATE#1,xc,yc:PRINT#1,CHR$(1
44);                                    (209B)
1200 e$=INKEY$                          (0C17)
1210 IF e$=CHR$(243)THEN xca=xc:yc
a=yc:xc=xc+4:IF xc>26 THEN xc=26         (4C87)
1220 IF e$=CHR$(242)THEN xca=xc:yc
a=yc:xc=xc-4:IF xc<2 THEN xc=2           (4AEB)
1230 IF e$=CHR$(241)THEN yca=yc:xc
a=xc:yc=yc+4:IF yc>25 THEN yc=25         (4CB7)
1240 IF e$=CHR$(240)THEN yca=yc:xc
a=xc:yc=yc-4:IF yc<1 THEN yc=1           (4A04)
1250 IF(fld(xca,yca)=0 AND xca>0 A
ND xca<27 AND yca>0 AND yca<26)THE
N LOCATE#1,xca,yca:PRINT#1," ";          (60D4)
1260 IF fld(xca,yca)=2 THEN LOCATE
#1,xca,yca:PEN#1,0:PRINT#1,CHR$(14
3);                                      (41ED)
1270 IF fld(xca,yca)=7 THEN LOCATE
#1,xca,yca:PEN#1,1:PRINT#1,CHR$(14
3);                                      (4157)
1280 IF fld(xca,yca)=1 THEN LOCATE
#1,xca,yca:PEN#1,2:PRINT#1,CHR$(14
3);                                      (4129)
1290 IF sp=1 THEN PEN#1,1 ELSE PEN
#1,0                                     (1AFC)
1300 IF INKEY(9)=0 THEN 1310 ELSE
1190                                     (16DC)
1310 ON sp GOSUB 1320,1370               (1334)
1320 IF fld(xc,yc)=0 THEN LOCATE#1
,xc,yc:PRINT#1," ";:GOTO 1160            (394C)
1330 IF fld(xc,yc)=2 THEN LOCATE#1
,xc,yc:PEN#1,0:PRINT#1,CHR$(143);:
PEN#1,1:GOTO 1160                        (482F)
1340 IF fld(xc,yc)=7 THEN LOCATE#1
,xc,yc:PEN#1,1:PRINT#1,CHR$(143);:
GOTO 1160                                (426C)
1350 PEN#1,1:fld(xc,yc)=7:LOCATE#1
,xc,yc:PRINT#1,CHR$(143);                (3C80)
1360 spiel=spiel+1:sp=2:RETURN           (22FB)
1370 IF fld(xc,yc)=0 THEN LOCATE#1
,xc,yc:PRINT#1," ";:GOTO 1160            (39B0)
1380 IF fld(xc,yc)=2 THEN LOCATE#1
,xc,yc:PEN#1,0:PRINT#1,CHR$(143);:
GOTO 1160                                (429C)
1390 IF fld(xc,yc)=7 THEN LOCATE#1
,xc,yc:PEN#1,1:PRINT#1,CHR$(143);:
PEN#1,0:GOTO 1160                        (484F)
1400 PEN#1,0:fld(xc,yc)=2:LOCATE#1
,xc,yc:PRINT#1,CHR$(143);                (3C2A)
1410 spiel=spiel+1:sp=1:RETURN           (2250)
1420 DATA 2,1,6,5,10,9,14,13,18,17
,22,21,26,25                             (2B64)
1430 '** GOSUB: LEERTASTE **             (1F68)
1440 LOCATE 2,24:PRINT"<Leertaste>
";                                       (1A94)
1450 IF INKEY(47)<>0 THEN 1450 ELS
E RETURN                                 (1599)
1460 DATA 3,1,4,1,5,1,4,2,5,2,6,2,
1,3,2,3,3,3,5,3,6,3,7,3,1,4,2,4,3,
4,5,4,6,4,7,4,1,5,2,5,3,5,5,5,6,5,
7,5,2,6,3,6,4,6,3,7,4,7,5,7              (7E28)
1470 DATA 1,3,1,4,1,5,2,2,2,3,2,4,
3,1,3,2,3,3,3,5,3,6,3,7,4,1,4,2,4,
3,4,5,4,6,4,7,5,1,5,2,5,3,5,5,5,6,
5,7,6,4,6,5,6,6,7,3,7,4,7,5              (7E42)
```

```
1480 DATA 1,4,2,3,3,2,1,3,2,2,3,1,
3,3,4,2,5,1,2,5,3,4,4,3,3,5,4,4,5,
3,4,5,5,4,6,3,3,7,4,6,5,5,5,6,6,5,
7,4,5,7,6,6,7,5                    <72EB>
```

```
1490 DATA 1,5,2,6,3,7,2,5,3,6,4,7,
1,3,2,4,3,5,3,4,4,5,5,6,3,3,4,4,5,
5,3,2,4,3,5,4,4,1,5,2,6,3,5,1,6,2,
7,3,5,3,6,4,7,5                    <726C>
```

Moonlander

Unser Autor **Markus Schöngarth** hat ein Programm für alle Leser der SCHNEIDER CPC WELT geschrieben, die Freude am Spielen haben und sich einmal als Raumschiffkommandant versuchen wollen.
Das Programm ist vollständig in BASIC geschrieben. Es zeichnet sich durch gefällige Grafik, schnellen Bildschirmaufbau und gute Steuerung durch den Joystick aus.

Stellen Sie sich vor, Ihr Raumschiff sei durch einen Meteoritentreffer beschädigt worden und Sie müßten die Landung auf einem erdfernen Planeten ausführen.
Doch leider ist die Oberfläche des Planeten wild zerklüftet und Sie müssen weich auf einer kleinen Plattform landen.
Dabei können Sie durch Gasgeben mit dem Feuerknopf Ihre Geschwindigkeit so regulieren, daß die Anziehungskraft des Planeten Ihr Raumschiff nicht zerschmettert.
Das ist nicht ganz einfach, denn von Level zu Level wird die Anziehungskraft größer. Außerdem besteht bei zuviel Schub die Gefahr, daß Sie auf Nimmerwiedersehen in den Weltraum hinauskatapultiert werden.
Durch Vergrößern oder Verkleinern der Plattform können Sie die Schwierigkeitsstufe verändern. Fangen Sie am besten mit Stufe 1 an, bis Sie genügend Übung haben.
Achten Sie bei Ihren Landungsversuchen unbedingt auf den Treibstoffvorrat, denn der ist nicht unbegrenzt. Je weniger Sprit Sie verbrauchen, desto größer ist die Punktezahl, die Sie bei einer erfolgreichen Landung gutgeschrieben bekommen. *JE* □

```
1 '******************************  <24B1>
2 '*        MOONLANDER          *  <244D>
3 '*           VON              *  <2449>
4 '*     MARKUS SCHOENGARTH     *  <2411>
5 '*           FUER             *  <2407>
6 '*     SCHNEIDER CPC-WELT     *  <24EF>
7 '*        CPC 664/6128     JE*  <24E1>
8 '******************************  <24BF>
10 punkte=0:hpunkte=0              <1D7B>
20 sch1=0:sch2=0:sch3=0:level=1:tr
=32:SYMBOL AFTER 32                <3B82>
25 IF punkte>hpunkte THEN hpunkte=
punkte                             <2F53>
30 SYMBOL 129,0,24,52,44,52,60,66,
231                                <1F60>
31 SYMBOL 130,0,20,42,86,42,52,20,
8                                  <1ED7>
32 SYMBOL 47,255,255,255,255,255,2
55,255,255                         <20A2>
33 SYMBOL 92,255,255,255,255,255,2
55,255,255                         <2055>
50 MODE 1:BORDER 0:INK 0,0:INK 1,1
3,26:INK 2,24:INK 3,6              <23B9>
55 SPEED INK 20,20                 <0CB1>
60 PEN 1:FOR i=1 TO 40:LOCATE INT(
RND*40)+1,INT(RND*25)+1:PRINT CHR$
(144):NEXT                         <3529>
65 FOR i=1 TO 200:PLOT INT(RND*640
)+1,INT(RND*400)+1,INT(RND*2)+1:NE
XT                                 <37C0>
70 PEN 2:RESTORE 10000:FOR i=1 TO
5:READ a$:LOCATE 9,3+i:PRINT a$:NE
XT                                 <2E05>
80 PEN 3:RESTORE 10060:FOR i=1 TO
5:READ a$:LOCATE 6,10+i:PRINT a$:N
EXT                                <2F30>
100 PEN 2:LOCATE 8,18:PRINT"Press
[1] for Instruction"               <2A24>
110 PEN 1:LOCATE 8,20:PRINT"
      or"                          <1F9F>
120 PEN 3:LOCATE 8,22:PRINT"
[2] for start Game"                <29A2>
130 PEN 1:LOCATE 6,24:PRINT"(c) 19
87 by Markus Schoengarth"          <2FCF>
135 LOCATE 14,1:PRINT"Highscore :"
;hpunkte                           <24C7>
140 a$=INKEY$                      <0C4A>
150 IF a$="1"THEN 20000            <1212>
```

```
160 IF a$="2"THEN punkte=0:GOTO 20
0                                   <1F81>
170 GOTO 140                        <09A7>
200 PEN 3:LOCATE 8,22:INPUT"Schwie
rigkeitsgrad (1-3)";zy:IF zy<1 OR
zy>3 THEN 200                       <4404>
201 IF zy=1 THEN gl=50              <16F4>
202 IF zy=2 THEN gl=40              <16D6>
203 IF zy=3 THEN gl=30              <16B8>
208 an=0:ges=0:hx=332:hy=368        <29D2>
209 SPEED INK 10,10                 <0C7B>
210 GOSUB 30000                     <09CC>
480 tr=32:MODE 0:BORDER 0:INK 0,0:
INK 2,24:INK 3,6:INK 4,1:INK 5,26:
INK 6,9,18:INK 7,18,9:INK 9,6:INK
10,1:INK 11,11:INK 12,2:INK 14,9,1
8                                   <5F4F>
490 FOR i=1 TO 100:PLOT INT(RND*68
0)+1,INT(RND*310)+90,INT(RND*14)+1
:NEXT                               <3976>
500 FOR i=1 TO 640 STEP 4:PLOT i,6
,INT(RND*3)+10:DRAW i,20+INT(RND*5
0):NEXT                             <3BBA>
505 PEN 2:LOCATE 20,25:PRINT"-";:L
OCATE 1,1                           <196B>
510 ORIGIN 485,250:FOR i=-PI/2 TO
PI*1.5 STEP 0.08:a=40*SIN(i):b=40*
COS(i):MOVE 0,0:IF INT(RND*2)=1 TH
EN fa=6 ELSE fa=7                   <6D8F>
520 MOVE 0,0:DRAW a,b,fa:NEXT       <1CCA>
525 ORIGIN 0,0                      <09A5>
530 po=INT(RND*400)+100:FOR i=1 TO
 gl:PLOT i+po,1,9:DRAW i+po,70:NEX
T                                   <4789>
540 PLOT 400,300,3:DRAW 550,200     <1719>
600 PLOT 1,1,0:DRAW 600,1           <1264>
610 PLOT 1,1,4:DRAW 600,1           <1280>
620 PLOT 1,5,4:DRAW 600,5           <1244>
700 PEN 2:LOCATE 1,25:PRINT"+":LOC
ATE 1,1                             <177F>
1010 PRINT CHR$(23)CHR$(1)          <1121>
1020 rau$=CHR$(129)                 <1235>
1021 feu$=CHR$(130)                 <12C2>
1022 cl$=CHR$(130)                  <113F>
1025 ORIGIN 0,0                     <098F>
1030 TAG                            <0653>
1800 MOVE hx,hy,2:PRINT rau$;       <1C4F>
1810 '                              <0736>
1815 h1x=hx:h1y=hy                  <1E65>
1820 TAGOFF:PRINT CHR$(23)CHR$(0);
:PLOT tr-5,3,14:DRAW tr,3:PRINT CH
R$(23)CHR$(1);:TAG                  <3B1D>
1825 IF INKEY(76)=0 THEN tr=tr+5:I
F tr>600 THEN 4000 ELSE SOUND 1,50
0,5,15,,,31:GOTO 2000               <428C>
1850 IF INKEY(74)=0 THEN hx=hx-5    <1CBB>
1860 IF INKEY(75)=0 THEN hx=hx+5    <1CEB>
1870 an=an+0.5+sch1:IF an>8+sch2 T
HEN an=8                            <385B>
1871 hy=hy+ges-an:IF hy<80 THEN 40
00 ELSE IF hy>86 AND hy<90 AND hx>
po-4 AND hx<gl+po+4 THEN 5000       <65DB>
1872 ges=ges-0.5:IF ges<0 THEN ges
=0                                  <2C66>
1880 MOVE h1x,h1y,2:PRINT rau$;:MO
VE hx,hy,2:PRINT rau$;              <3673>
1970 GOTO 1815                      <0935>
2000 MOVE h1x,h1y-20,3:PRINT feu$;
:MOVE hx,hy-20,3:PRINT feu$;        <3CC5>
2002 FOR m=1 TO 10:NEXT             <1125>
2005 ges=ges+1:IF ges>8+sch2 THEN
GOTO 3000                           <2BDF>
2006 hy=hy+ges                      <17D6>
2007 an=an-1:IF an<0 THEN an=0      <2313>
2010 MOVE h1x,h1y-20,0:PRINT cl$;:
MOVE hx,hy-20,0:PRINT cl$;          <3AEE>
2020 GOTO 1870                      <094B>
2100 '                              <077B>
2101 MOVE h1x,h1y,0:PRINT CHR$(143
);:MOVE hx,hy,0:PRINT CHR$(143);    <3678>
2102 ges=5:hy=hy+ges:h1y=hy         <2DF1>
2105 MOVE h1x,h1y,2:PRINT rau$;:MO
VE hx,hy,2:PRINT rau$;              <3636>
2120 GOTO 2100                      <094E>
3000 '                              <0785>
3010 TAGOFF:PRINT CHR$(23)CHR$(0):
TAG:MOVE h1x,h1y:PRINT CHR$(32);:T
AGOFF:PRINT CHR$(23)CHR$(1):TAG:FO
R i=h1y-20 TO 400 STEP 5:MOVE h1x,
i:PRINT rau$;:MOVE hx,i:PRINT rau$
;:NEXT                              <8194>
3020 GOTO 4000                      <09D1>
4000 '                              <0759>
4010 MOVE hx,hy,2:PRINT CHR$(238);
                                    <1C65>
4110 FOR i=12 TO 15:INK 0,i:SOUND
1,400,i,i,,,30:NEXT                 <3051>
4115 MOVE hx,hy,2:PRINT CHR$(203);
                                    <1C02>
4120 FOR i=15 TO 4 STEP-1:INK 0,i:
SOUND 1,500,i+20,i,,,31:SOUND 2,40
0,i+20,i,,,20:NEXT                  <4E74>
4125 MOVE hx,hy,2:PRINT CHR$(144);
                                    <1C5F>
4200 FOR i=15 TO 0 STEP-1:INK 0,i:
SOUND 1,200,i,i,,,30:SOUND 1,400,i
,i,,,30:BORDER INT(RND*26)+1:NEXT:
BORDER 0                            <576F>
4205 MOVE hx,hy,2:PRINT CHR$(32);   <1C08>
4210 TAGOFF:PRINT CHR$(23)CHR$(0):
LOCATE 5,10:PRINT"GAME OVER"        <261D>
4220 LOCATE 5,13:PRINT"<<SPACE>>"   <17C3>
4230 a$=INKEY$                      <0C4E>
4240 IF a$=" "THEN 20               <12F5>
4250 GOTO 4230                      <09A7>
5000 '                              <072D>
5005 bonus=600-tr:punkte=punkte+bo
nus                                 <348B>
5010 FOR i=1500 TO 100 STEP-10:SOU
ND 1,i,INT(RND*4)+1,INT(RND*5)+10:
BORDER INT(RND*25)+1:NEXT           <43EB>
```

Moonlander

```
5012 bo=0                                   (0C3A)
5015 PLOT tr-5,3,14:DRAW tr+bo,3:b
o=bo+10:SOUND 1,bo,5,15:IF tr+bo>6
00 THEN 5017 ELSE 5015                      (582D)
5017 PRINT CHR$(23)CHR$(0):TAGOFF:
LOCATE 7,13:PEN 1:PRINT"<<SPACE>>"
                                            (291D)
5018 a$=INKEY$                              (0C79)
5019 IF a$=" "THEN 5020 ELSE 5018           (1720)
5020 MODE 1:INK 1,6:INK 2,24:INK 3
5030 PEN 3:LOCATE 2,2:PRINT STRING
$(38,154):LOCATE 2,25:PRINT STRING
$(38,154);                                  (2908)
5040 FOR i=2 TO 25:LOCATE 1,i:PRIN
T CHR$(149):LOCATE 40,i:PRINT CHR$
(149);:NEXT                                 (3385)
5070 PEN 2:TAGOFF:PRINT CHR$(23)CH
R$(0):LOCATE 17,3:PRINT"Bravo !!!"
                                            (298F)
5080 PEN 1:LOCATE 6,5:PRINT"Sie ha
ben tatsaechlich";:PEN 3:PRINT" Le
vel";level:LOCATE 16,7:PEN 1:PRINT
"geschafft."                                (540B)
5090 LOCATE 4,9:PRINT"Sie erhalten
";INT((bonus/10)*zy);"Punkte fuer
die":LOCATE 7,11:PRINT"unverbrauch
te Energie "                                (631D)
5100 LOCATE 3,13:PRINT"Da Sie so g
ut landen koennen, duerfen":LOCATE
 3,15:PRINT"Sie sich nun an einer
schwierigeren":LOCATE 13,17:PRINT"
Landung pruefen."                           (7BE7)
5110 LOCATE 3,19:PRINT"Die Anziehu
ng des Planeten ist jetzt"                  (3204)
5115 LOCATE 17,21:PRINT"groesser."
                                            (1858)
5117 PEN 2:LOCATE 11,23:PRINT"Bitt
e Space druecken"                           (26B5)
5120 a$=INKEY$                              (0C47)
5125 IF a$=" "THEN 5130                     (12E2)
5127 GOTO 5120                              (0968)
5130 level=level+1:sch1=sch1+0.5:s
ch2=sch2+2:INK 3,6                          (4672)
5140 MODE 1:GOTO 200                        (0CC5)
10000 DATA /   /  ///   ///  /   /
                                            (1EEB)
10010 DATA // // /   / /   / //  /
                                            (1E15)
10020 DATA / / / /   / /   / / / /
                                            (1EB5)
10030 DATA / / / /   / /   / / / /
                                            (1EC9)
10040 DATA /   /  ///   ///  /  //
                                            (1EB8)
10050 '                                     (07B6)
10060 DATA \    \\\\ \  \ \\\  \\\
\ \\\\                                      (24C2)
10070 DATA \    \  \ \\ \ \  \ \
  \  \                                      (2444)
10080 DATA \    \  \ \ \\ \  \ \\\
\ \\\\                                      (246C)
10090 DATA \    \\\\ \  \ \  \ \
  \ \                                       (2392)
10100 DATA \\\\ \  \ \  \ \\\  \\\
\ \  \                                      (241F)
20000 '                                     (0799)
20001 INK 1,2                               (0977)
20010 GOSUB 30000                           (09CB)
20015 LOCATE 1,1                            (096E)
20020 PEN 3:a$="              Instru
ctions":GOSUB 32000                         (2CEE)
20030 PEN 2:a$="              ======
======":GOSUB 32000                         (2CA0)
20040 PRINT:PEN 1:a$="Ihr Raumschi
ff ist durch einen Kampf    schwer
 beschaedigt wurden.":GOSUB 32000           (5837)
20050 a$="Ihre einzige Moeglichkei
t ist, mit einerkleinen Raumfaehre
 auf dem Planeten     Alphi-Alphu
zu landen.":GOSUB 32000                     (7722)
20060 a$="Alphi-Alphu besitzt aber
 eine ziemlich  spitzte Oberflaech
e.Nur auf einer        kleinen Plat
tform ist es moeglich zu    landen
.":GOSUB 32000                              (9013)
20065 a$="Mit dem Joystick steuert
 man die        Raumfaehre nach re
chts und links.":GOSUB 32000               (5AE5)
20070 a$="Durch Druck auf den Feue
rknopf gibt man Gas.Aber Vorsicht
wenn man zuviel Gas   gibt fliegt
man fuer immer in den Welt- raum."
:GOSUB 32000                                (8EB6)
20080 PEN 2:PRINT:a$="Ich wuensche
 Ihnen noch viel Spass bei  Ihrer
Bruchlandung !!!":GOSUB 32000               (5447)
20090 PEN 3:PRINT:a$="       Bitte
Feuerknopf druecken !":GOSUB 32000
                                            (37BB)
20100 a$=INKEY$                             (0C88)
20110 IF a$="X"THEN INK 1,26:CLS:G
OTO 200 ELSE 20100                          (2099)
30000 FOR i=1 TO 49:PRINT:NEXT:RET
URN                                         (153D)
32000 FOR i=1 TO LEN(a$):PRINT MID
$(a$,i,1);:IF MID$(a$,i,1)=" "THEN
 32020                                      (3E87)
32010 SOUND 1,100+VAL(a$),1                 (1525)
32020 FOR b=1 TO 40:NEXT                    (11EB)
32030 NEXT                                  (067B)
32040 PRINT:FOR b=1 TO 240:NEXT:RE
TURN                                        (1594)
```

Keine Lust zum Abtippen? Softbox bestellen! Coupon Seite 81

Harry

Das Programm unseres Autors **Markus Rausch** ist in reinem BASIC geschrieben. Trotzdem ist es recht schnell und erinnert etwas an das berühmte Pac-Man-Spiel. Harry ist ein Geschicklichkeitsspiel, bei dem es darum geht, die Freundin aus einem Schloß zu befreien, das von Geistern bewacht wird, die Sie nicht vernichten können. Erwischt Sie ein Geist, so haben Sie ein Leben weniger.
Im Schloß liegt das Lösegeld verstreut, mit dem Harry seine Freundin freikaufen kann. Außer den Geistern gibt es noch Tretminen, die jeweils ein Leben kosten, wenn Sie sie berühren. Die Minen sind normalerweise unsichtbar. Sie können sie sichtbar machen, wenn Sie den Feuerknopf betätigen. Leider nimmt dadurch Harrys Energie, die am Bildschirmrand zu sehen ist, ab. Wenn sie ganz aufgebraucht ist, bleiben die Minen unsichtbar, selbst wenn Sie wie ein Wilder auf dem Joystick herumhämmern.
Wenn Sie das Geld in einem Screen gesammelt haben, erscheint ein Herz, das Sie ebenfalls aufnehmen müssen, um in den nächsten Screen zu kommen.
Gespielt wird Harry mit dem Joystick.
Scheint es Ihnen, ein Level sei nicht zu lösen, verlieren Sie nicht den Mut. *JE*□

```
1 '******************************  <24B1>
2 '*           HARRY              *  <24F6>
3 '*            VON               *  <24D0>
4 '*       MARKUS RAUSCH          *  <2472>
5 '*            FUER              *  <2424>
6 '*    SCHNEIDER CPC-WELT        *  <24EF>
7 '*     CPC 464/664/6128     JE*  <2453>
8 '******************************  <24BF>
10 MODE 1:INK 1,2,6:PEN 1:LOCATE 1
4,24:PRINT"BITTE WARTEN"            <2747>
20 SYMBOL 240,112,248,238,255,190,
200,248,112                         <20B7>
30 SYMBOL 241,60,126,219,255,195,1
89,255,219                          <20BA>
40 SYMBOL 242,14,31,119,255,125,19
,31,14                              <2000>
50 SYMBOL 243,60,66,153,137,137,13
7,66,60                             <20A6>
60 SYMBOL 244,255,141,231,160,231,
160,211,255                         <2046>
70 SYMBOL 245,0,0,24,0,0,0,0,0      <19C7>
80 MODE 1:BORDER 0                  <0A40>
90 INK 0,0:INK 1,0:INK 2,0:INK 3,0
                                    <1816>
100 y=270:g=50:j=150:k=250:l=350:m
=450                                <37BF>
110 FOR x=10 TO 28 STEP 3           <1255>
120 PLOT x+g,y,3:DRAWR 0,100:MOVE
x+g+75,y:DRAWR 0,100:MOVE x+g,y+50
:DRAWR 75,0:'H                      <5247>
130 PLOT x+j,y,1:DRAWR 0,100:DRAWR
 75,0:DRAWR 0,-100:MOVE x+j,y+50:D
RAWR 75,0:'A                        <4626>
140 PLOT x+k,y,2:DRAWR 0,100:DRAWR
 75,0:DRAWR 0,-50:DRAWR-75,0:DRAW
x+k+75,y:'R                         <47AA>
150 PLOT x+l,y,1:DRAWR 0,100:DRAWR
 75,0:DRAWR 0,-50:DRAWR-75,0:DRAW
x+l+75,y:'R                         <47D5>
160 PLOT x+m,y+100,3:DRAW x+m+38,y
+50:DRAW x+m+75,y+100:MOVE x+m+38,
y:DRAWR 0,50:'Y                     <6239>
170 y=y+3:NEXT                      <12C8>
180 LOCATE 19,12:PEN 1:PRINT"by"    <142B>
190 '****diamant****                <1607>
200 w=30:e=135:b=60:q=120           <2404>
210 FOR a=285 TO 270 STEP-1         <151F>
220 PLOT a,e,3:DRAWR w,0            <19AE>
230 w=w+2:e=e-1                     <1C66>
240 NEXT                            <06A2>
250 FOR a=270 TO 300                <1284>
260 PLOT a,q:DRAWR b,0              <17D8>
270 b=b-2:q=q-1                     <1C99>
280 NEXT                            <06F3>
290 '****strahlen****               <178F>
300 FOR a=1 TO 360 STEP 20          <1335>
310 DEG                             <06B3>
320 MOVE 300+40*COS(a),120+40*SIN(
a)                                  <2455>
330 DRAW 300+70*COS(a),120+70*SIN(
a)                                  <2478>
340 NEXT                            <066C>
350 '****kanten****                 <159D>
360 PLOT 300,120,0:DRAWR 0,-30      <15B8>
370 MOVE 270,120:DRAWR 60,0         <123A>
380 MOVE 286,135:DRAWR 15,-15:DRAW
R 15,16                             <1BF9>
390 MOVE 315,120:DRAWR-15,-32:DRAW
R-15,32                             <1D90>
400 '****laufschrift****           <1A96>
410 INK 0,0:INK 1,24:INK 2,14:INK
3,6                                 <1A18>
```

Harry

```
420 FOR x=1 TO 500:NEXT                    (1234)
430 y=33                                   (0CC2)
440 FOR x=1 TO 13                          (0FF0)
450 LOCATE x,24:PRINT" MARCUS ":LO
CATE y,24:PRINT"RAUSCH"                    (2C8F)
460 FOR v=1 TO 150:NEXT                    (116D)
470 y=y-1                                  (1058)
480 NEXT x                                 (0A6E)
490 CALL &BB06                             (0922)
500 '                                      (07F3)
510 '                                      (0708)
520 '                                      (071D)
530 '                                      (0731)
540 '                                      (0745)
550 '                                      (0759)
560 DIM feld(18,23)                        (1411)
570 CLS:INK 2,14:PEN 2                     (0F66)
580 a$="* HARRY *":b$="Brauchen Si
e Instruktionen (J/N) ?"                   (3FF6)
590 LOCATE 16,2                            (0ACA)
600 FOR p=1 TO 9:PRINT MID$(a$,p,1
);                                         (1F65)
610 SOUND 1,25,5,7:FOR x=1 TO 75:N
EXT x                                      (1F27)
620 NEXT p                                 (0A66)
630 FOR x=1 TO 1500:NEXT x                 (16C0)
640 LOCATE 4,4                             (0900)
650 FOR p=1 TO 34:PRINT MID$(b$,p,
1);                                        (202C)
660 SOUND 1,25,5,7:FOR x=1 TO 75:N
EXT x                                      (1F8B)
670 NEXT p                                 (0ACA)
680 ant$=INKEY$:IF ant$=""THEN 680
                                           (1D30)
690 an$=UPPER$(ant$)                       (1585)
700 LOCATE 20,6:PEN 3:PRINT an$            (1454)
710 IF an$="N"THEN 840 ELSE IF an$
="J"THEN 730                               (2310)
720 GOTO 680                               (09DC)
730 LOCATE 1,8:PEN 1                       (0CD6)
740 PRINT"         HALLO HIER SPRIC
HT HARRY         ":PRINT                   (3271)
750 PRINT"GANGSTER HALTEN MEINE FR
EUNDIN GEFANGEN.                           (2FDE)
760 PRINT"DAMIT ICH SIE LEBEND WIE
DERSEHE MUSS ICH                           (2F13)
770 PRINT"IHNEN EIN SEHR HOHES LOE
SEGELD ZAHLEN.DA                           (2F2D)
780 PRINT"ICH ABER SEHR ARM BIN MU
SS ICH VERSUCHEN                           (2FA6)
790 PRINT"DIE  VIELEN  SCHAETZE  I
N  DER BURGRUINE                           (2F49)
800 PRINT"AUSZUGRABEN .
                 "                         (3030)
810 PRINT"ES WIRD SEHR SCHWIERIG D
ENN DORT WIMMELT                           (2F9B)
820 PRINT"ES NUR SO VON GEISTERN U
ND !MIENEN!.                               (2B36)
830 CALL &BB06                             (09CC)
840 CLS                                    (06BD)
850 PEN 2:LOCATE 11,2:PRINT"SCHWIE
RIGKEITSGRAD"                              (23BD)
860 LOCATE 1,8:PEN 3:PRINT STRING$
(40,CHR$(154)):PEN 1                       (1E36)
870 PRINT" LEVEL 1 ..... >> ANFAE
NGER        <<"                            (2F7F)
880 PRINT" LEVEL 2 ..... >> FORTG
ESCHRITTENER <<"                           (2FCB)
890 PRINT" LEVEL 3 ..... >> PROFI
            <<"                            (2FC8)
900 PRINT" LEVEL 4 ..... >> SELBS
MOERDER     <<"                            (2FC9)
910 LOCATE 1,15:PEN 3:PRINT STRING
$(40,CHR$(154)):PEN 1                      (1F58)
920 LOCATE 11,20:PEN 2:PRINT"BITTE
  WAEHLEN SIE"                             (24D3)
930 IF INKEY(64)=0 THEN ll=0.39:GO
TO 980                                     (2090)
940 IF INKEY(65)=0 THEN ll=0.49:GO
TO 980                                     (2022)
950 IF INKEY(57)=0 THEN ll=0.59:GO
TO 980                                     (20E7)
960 IF INKEY(56)=0 THEN ll=0.69:GO
TO 980                                     (20B5)
970 GOTO 930                               (09A5)
980 as=0                                   (0CC8)
990 '***** ZUWEISUNGEN *****               (1E63)
1000 zu=3:level=1:live=3:mauer$=CH
R$(244):harl$=CHR$(242):harr$=CHR$
(240):geist$=CHR$(241):miene$=CHR$
(253):geld$=CHR$(243):truhe$=CHR$(
228):hv$=harr$:ypp=390                     (A7B3)
1010 '***** HAUPTPROGRAMM *****           (2031)
1020 IF level=4 AND ll=0.79 THEN l
l=0.69:as=2                                (320B)
1030 IF level=5 AND as=2 THEN as=0
:GOTO 3030                                 (25F3)
1040 IF level=5 AND as<>2 THEN lev
el=1:ll=ll+0.1                             (3651)
1050 CLS:ypp=390                           (115E)
1060 IF live<0 THEN 3290                   (136C)
1070 ON level GOTO 1690,1670,1700,
1680                                       (1E3D)
1080 GOSUB 1240:LOCATE hx,hy:PEN 1
:PRINT hv$:LOCATE hx,hy:PEN 3:PRIN
T CHR$(22)+CHR$(1);CHR$(245);CHR$(
22)+CHR$(0)                                (52AD)
1090 IF geld=gan THEN feld(heimx,h
eimy)=4:LOCATE heimx,heimy:PEN 4:P
RINT truhe$                                (51EC)
1100 '*** BEWEGUNG GEISTER ***             (1FE5)
1110 PEN 5                                 (07DA)
1120 FOR l=1 TO 2                          (0EC0)
1130 LOCATE gx(l),gy(l):PRINT CHR$
(32)                                       (25B1)
1140 IF gx(l)>hx THEN xx=-ll               (2497)
1150 IF gx(l)<hx THEN xx=ll                (23EF)
1160 IF gy(l)>hy THEN yy=-ll               (244C)
1170 IF gy(l)<hy THEN yy=ll                (238F)
1180 IF feld(gx(l)+xx,gy(l))=0 THE
```

```
N gx(l)=gx(l)+xx                    (4CB6)
1190 IF feld(gx(l),gy(l)+yy)=0 THE
N gy(l)=gy(l)+yy                    (4CF8)
1200 LOCATE gx(l),gy(l):PRINT geis
t$                                  (278D)
1210 IF CINT(gx(l))=hx AND CINT(gy
(l))=hy THEN 1610                   (350C)
1220 NEXT                           (064E)
1230 GOTO 1080                      (0967)
1240 '***** JOY-ABFRAGE *****       (1E5E)
1250 IF JOY(0)=0 THEN RETURN ELSE
LOCATE hx,hy:PRINT CHR$(32):SOUND
1,0,1,7,0,0                         (320D)
1260 IF JOY(0)=1 THEN 1340          (11BE)
1270 IF JOY(0)=2 THEN 1360          (11B3)
1280 IF JOY(0)=4 THEN 1350          (11F8)
1290 IF JOY(0)=8 THEN 1330          (1165)
1300 IF JOY(0)=16 THEN 1540         (1207)
1310 INK 7,0                        (0980)
1320 RETURN                         (067C)
1330 ri=1:ry=0:hv$=harr$:GOTO 1370
:'RECHTS                            (309A)
1340 ri=0:ry=-1:GOTO 1370:'OBEN     (21DC)
1350 ri=-1:ry=0:hv$=harl$:GOTO 137
0:'LINKS                            (30FF)
1360 ri=0:ry=1:'UNTEN               (1C76)
1370 IF feld(hx+ri,hy+ry)=0 THEN h
x=hx+ri:hy=hy+ry:RETURN             (4E80)
1380 IF feld(hx+ri,hy+ry)=1 THEN R
ETURN                               (2A5B)
1390 IF feld(hx+ri,hy+ry)=2 THEN g
eld=geld+1:hx=hx+ri:hy=hy+ry:SOUND
 1,400-20,10,7:feld(hx,hy)=0:RETUR
N                                   (86C0)
1400 IF feld(hx+ri,hy+ry)=3 THEN h
x=hx+ri:hy=hy+ry:GOTO 1610          (5179)
1410 IF feld(hx+ri,hy+ry)=4 THEN h
x=hx+ri:hy=hy+ry:level=level+1:liv
e=live+1:geld=0:feld(heimx,heimy)=
0:GOSUB 1430:GOTO 1010              (A3D8)
1420 '**** HERZ ERREICHT ****       (1EB1)
1430 RESTORE 1500                   (09FD)
1440 READ pe,la                     (1189)
1450 IF la=-1 THEN RETURN           (107D)
1460 la=la/2                        (12F6)
1470 SOUND 1,pe,la:SOUND 2,pe-1,la
:SOUND 4,pe-2,la                    (35BC)
1480 SOUND 1,0,5:SOUND 2,0,5:SOUND
 4,0,5                              (193F)
1490 GOTO 1440                      (09AF)
1500 DATA 379,50,379,50,358,50,319
,50,319,50,358,50,379,50,426,50     (3E5A)
1510 DATA 478,50,478,50,426,50,379
,50,379,50,426,50,426,100           (3846)
1520 DATA 0,-1                      (0B22)
1530 RETURN                         (0620)
1540 '***** ENERGYABZUG *****       (1E42)
1550 IF ypp<36 THEN 1310            (1326)
1560 LOCATE hx,hy:PEN 1:PRINT hv$:
LOCATE hx,hy:PEN 3:PRINT CHR$(22)+
CHR$(1);CHR$(245);CHR$(22)+CHR$(0)
                                    (4D38)
1570 SOUND 1,400-ypp,1,6,0,0        (1AFF)
1580 INK 7,6:MOVE 600,ypp:DRAWR 26
,0,0:ypp=ypp-2                      (2D86)
1590 GOTO 1300                      (091D)
1600 '***** TOD *****               (1645)
1610 SOUND 1,384,100,12,0,0,1:SOUN
D 2,184,100,12,0,0,1                (29CC)
1620 live=live-1                    (16EC)
1630 geld=0:feld(heimx,heimy)=0     (2B93)
1640 FOR qq=1 TO 1000:NEXT          (1318)
1650 GOTO 1050                      (09C1)
1660 '*** RESTORE-ANWEISUNG ***     (2072)
1670 RESTORE 1960:GOSUB 1720:RESTO
RE 2980:GOSUB 2920:GOTO 1080        (1D10)
1680 RESTORE 2200:GOSUB 1720:RESTO
RE 2990:GOSUB 2920:GOTO 1080        (1D11)
1690 RESTORE 2440:GOSUB 1720:RESTO
RE 3000:GOSUB 2920:GOTO 1080        (1D9D)
1700 RESTORE 2670:GOSUB 1720:RESTO
RE 3010:GOSUB 2920:GOTO 1080        (1D02)
1710 '***** AUFBAU *****            (19E6)
1720 MODE 0                         (071B)
1730 zu=zu+2                        (12BB)
1740 IF zu=9 THEN zu=19             (1622)
1750 IF zu>19 THEN zu=5             (16A8)
1760 INK 2,zu:INK 4,9:INK 5,26:INK
 6,10:INK 7,0                       (238D)
1770 FOR zy=1 TO 23                 (10A4)
1780 FOR zx=1 TO 18                 (1024)
1790 feld(zx,zy)=0                  (1B28)
1800 NEXT:NEXT                      (08E8)
1810 FOR zy=1 TO 23                 (10F5)
1820 READ aufb$                     (0DFF)
1830 FOR zx=1 TO 18                 (1089)
1840 ab$=MID$(aufb$,zx,1)           (1DB1)
1850 IF ab$="1"THEN PEN 2:feld(zx,
zy)=1:LOCATE zx,zy:PRINT mauer$     (40C6)
1860 IF ab$="2"THEN PEN 6:feld(zx,
zy)=2:LOCATE zx,zy:PRINT geld$      (3F48)
1870 IF ab$="3"THEN PEN 7:feld(zx,
zy)=3:LOCATE zx,zy:PRINT miene$     (40D5)
1880 NEXT                           (067A)
1890 NEXT                           (068E)
1900 LOCATE 1,24:PEN 5:PRINT"LEBEN
 :";live:LOCATE 1,25:PRINT"BILD  :
";level                             (3A24)
1910 FOR b=599 TO 631               (12C9)
1920 PLOT b,32,4:DRAWR 0,362        (1685)
1930 NEXT b                         (0A6F)
1940 RETURN                         (0655)
1950 '***** BILDERDATEN *****       (1EC4)
1960 DATA 1111111111111111          (1955)
1970 DATA 1200000000000021          (1981)
1980 DATA 1011111100111111          (1975)
1990 DATA 1010000300000101          (19F0)
2000 DATA 1010000000000101          (1997)
2010 DATA 1010011100111101          (19DD)
2020 DATA 1010012000021001          (19C0)
```

```
2030 DATA 1010010000000100101      (1914)
2040 DATA 1010010011001100101      (190F)
2050 DATA 1010010000000100101      (193D)
2060 DATA 1000010000000100001      (193F)
2070 DATA 1000000011000003301      (192F)
2080 DATA 1000010000000100001      (1967)
2090 DATA 1010010000000100101      (198D)
2100 DATA 1010010011001100101      (1988)
2110 DATA 1010010000000100101      (19B5)
2120 DATA 1010012000021100101      (1988)
2130 DATA 1010011100111100101      (19CF)
2140 DATA 1010000000000000101      (19B1)
2150 DATA 1010000000000000101      (19C5)
2160 DATA 1011111100111111101      (19DF)
2170 DATA 1200000000000000021      (1912)
2180 DATA 1111111111111111111      (190E)
2190 'BILD 1                       (0DDC)
2200 DATA 1111111111111111111      (1936)
2210 DATA 1200000211200000001      (19C3)
2220 DATA 1000000111100000001      (19D9)
2230 DATA 1000100000000010001      (1909)
2240 DATA 1001110000001110001      (198B)
2250 DATA 1001110011001110001      (1981)
2260 DATA 1000100011000102001      (1959)
2270 DATA 1000000000300000001      (19A0)
2280 DATA 1000003000000000001      (198D)
2290 DATA 1000000000000000001      (19CA)
2300 DATA 1000100020000100001      (1985)
2310 DATA 1001112000001110001      (19D6)
2320 DATA 1001110011001110001      (190D)
2330 DATA 1000100011300100001      (19EC)
2340 DATA 1000000000000000001      (1930)
2350 DATA 1000000000000000021      (1954)
2360 DATA 1000000011020000031      (1956)
2370 DATA 1000000011000000001      (1958)
2380 DATA 1020100011000100001      (1925)
2390 DATA 1001110011001110001      (1999)
2400 DATA 1001113011001110001      (190D)
2410 DATA 1000000000000000001      (19BC)
2420 DATA 1111111111111111111      (19F0)
2430 'BILD 2                       (0DC1)
2440 DATA 1111111111111111111      (1917)
2450 DATA 1000000011002000001      (19F7)
2460 DATA 1000000011300000001      (1931)
2470 DATA 1000000011000000001      (191F)
2480 DATA 1000000011000000001      (1933)
2490 DATA 1000100011000100001      (1905)
2500 DATA 1011111011011111101      (1991)
2510 DATA 1000100011000100001      (192E)
2520 DATA 1000100011000100001      (1942)
2530 DATA 1000100311000100001      (19D2)
2540 DATA 1000100011000100001      (196A)
2550 DATA 1011111000011111101      (1922)
2560 DATA 1000100011000100001      (1993)
2570 DATA 1000100011300100001      (19CD)
2580 DATA 1000100011000100001      (19BB)
2590 DATA 1000100011000100001      (19CF)
2600 DATA 1011111011011111101      (195B)
2610 DATA 1000100011000100001      (19F7)
2620 DATA 1000100011000100001      (190A)
2630 DATA 1000000011000000001      (1961)
2640 DATA 1000000311000000001      (1904)
2650 DATA 1000020011000000001      (1907)
2660 DATA 1111111111111111111      (19D1)
2670 ' Bild 5                      (0EA1)
2680 DATA 1111111111111111111      (19F9)
2690 DATA 1200010000001000021      (1964)
2700 DATA 1011010010011011101      (191D)
2710 DATA 1011010010011011101      (1931)
2720 DATA 1000000000000000001      (1928)
2730 DATA 1011113010311111101      (19A1)
2740 DATA 1000000000000000001      (1950)
2750 DATA 1000000000000000001      (1964)
2760 DATA 1011110010011111101      (198E)
2770 DATA 1010010010010001101      (1952)
2780 DATA 1000010000001000001      (1961)
2790 DATA 1200013000031000021      (19DA)
2800 DATA 1000010000001000001      (1989)
2810 DATA 1010010010010001101      (19A2)
2820 DATA 1011110010011111101      (1907)
2830 DATA 1000000000000000001      (1905)
2840 DATA 1000000000000000001      (191A)
2850 DATA 1011113010311111101      (1991)
2860 DATA 1000000000000000001      (1942)
2870 DATA 1011010010011011101      (1971)
2880 DATA 1011010010011011101      (1985)
2890 DATA 1200010000001000021      (19F6)
2900 DATA 1111111111111111111      (19B2)
2910 'STANDORT-KOORDINATEN LESEN   (210B)
2920 READ gx(1),gy(1),gx(2),gy(2),
hx,hy,heimx,heimy,gan              (4E29)
2930 LOCATE gx(1),gy(1):PRINT geis
t$                                 (21B5)
2940 LOCATE gx(2),gy(2):PRINT geis
t$                                 (2123)
2950 LOCATE hx,hy:PEN 1:PRINT harr
$                                  (1D24)
2960 RETURN                        (0651)
2970 'STANDORT-KOORDINATEN         (1B03)
2980 DATA 7,2,12,22,10,13,17,12,8  (1EE3)
2990 DATA 17,2,16,20,10,12,10,12,9
                                   (1F47)
3000 DATA 2,2,17,22,10,12,10,12,2  (1E0E)
3010 DATA 2,8,17,17,9,12,9,12,6    (1C7A)
3020 '***** HAPPY END *****        (1C6E)
3030 MODE 1:INK 3,6                (0C23)
3040 LOCATE 20,3:PEN 1:PRINT CHR$(
240):LOCATE 20,3:PEN 3:PRINT CHR$(
22)+CHR$(1);CHR$(245);CHR$(22)+CHR
$(0)                               (4055)
3050 LOCATE 22,3:PRINT CHR$(242):L
OCATE 22,3:PEN 1:PRINT CHR$(22)+CH
R$(1);CHR$(245);CHR$(22)+CHR$(0)   (3DD6)
3060 INK 2,6,0:LOCATE 21,2:PEN 2:P
RINT CHR$(228):PEN 3               (1FF6)
3070 LOCATE 1,5:PRINT STRING$(40,C
HR$(154))                          (18B9)
3080 LOCATE 19,6:PRINT"HURRA":PEN
1:LOCATE 1,8                       (1BCF)
3090 PRINT"SIE HABEN DAS  UNMOEGLI
```

```
CHE NUN  DOCH GE-"                    (3039)
3100 PRINT"SCHAFFT. SIE HABEN IHRE
 FREUNDIN BEFREIT"                    (3057)
3110 PRINT"UND VERBRINGEN DEN REST
 IHRES LEBENS MIT"                    (30D2)
3120 PRINT"IHR. AUF DAS DER NACHWU
CHS KOMME !!!!!!!"                    (3004)
3130 PEN 3:PRINT STRING$(40,CHR$(1
54))                                  (160A)
3140 RESTORE 3200                     (09A3)
3150 READ per,lan                     (13EE)
3160 IF lan=-1 THEN 3240              (1343)
3170 SOUND 1,per,lan:SOUND 2,per-1
,lan:SOUND 4,per-2,lan                (3BA2)
3180 SOUND 1,0,5:SOUND 2,0,5:SOUND
 4,0,5                                (198F)
3190 GOTO 3150                        (0989)
3200 DATA 319,25,319,25,284,50,319
,50,239,50,253,75,319,25              (3734)
3210 DATA 319,25,284,50,319,50,213
,50,239,75,319,25,319,25              (374A)
3220 DATA 159,50,190,50,239,50,253
,50,284,50,179,25,179,25              (3709)
3230 DATA 190,50,239,50,213,50,239
,75,0,-1                              (272D)
3240 PEN 1:PRINT"     NOCH EIN GEW
AGTES SPIEL ??????     "              (333A)
3250 PEN 3:PRINT STRING$(40,CHR$(1
54))                                  (16FB)
3260 nm$=UPPER$(INKEY$)               (11A2)
3270 IF nm$="J"THEN GOTO 570 ELSE
IF nm$="N"THEN CALL 0                 (2306)
3280 GOTO 3260                        (09AD)
3290 '*****  TRAGEDY  *****          (1CB9)
3300 MODE 1:PEN 2                     (0A47)
3310 FOR l=1 TO 40                    (0FCC)
3320 LOCATE l,1:PRINT geist$          (16BF)
3330 LOCATE l,24:PRINT geist$         (17C1)
3340 NEXT                             (06E7)
3350 FOR l=2 TO 23                    (0FF9)
3360 LOCATE 1,l:PRINT geist$          (16B4)
3370 LOCATE 40,l:PRINT geist$         (17FB)
3380 NEXT                             (0638)
3390 PRINT CHR$(22);CHR$(1)           (1257)
3400 LOCATE 16,3:PEN 1:PRINT"SCHAD
E !":LOCATE 1,6                       (1E3F)
3410 PRINT SPC(2)"Sie haben  es ni
cht  geschafft  Ihre                  (2F0F)
3420 PRINT SPC(2)"Freundin aus den
 Klauen der  Raeuber                  (2F3D)
3430 PRINT SPC(2)"zu befreien.        (1766)
3440 PRINT SPC(2)"Sie haben nur no
ch ein Chance um sie                  (2F14)
3450 PRINT SPC(2)"dort herauszuhol
en.":LOCATE 1,13                      (2576)
3460 PRINT SPC(2)"!!!!! SPIELEN SI
E  NOCH EINMAL !!!!! ":LOCATE 1,15
                                      (3750)
3470 PRINT SPC(2)"      Also was i
st JA oder NEIN                       (2A5D)
3480 a$=INKEY$:IF a$=""THEN 3480      (19C1)
3490 a$=UPPER$(a$)                    (1278)
3500 IF a$="N"THEN CLS:INK 1,24,6:
LOCATE 16,12:PRINT"SADIST!":FOR x=
1 TO 2000:NEXT:CALL 0                 (3B98)
3510 IF a$="J"THEN PRINT CHR$(22);
CHR$(1):GOTO 570 ELSE GOTO 3480       (2706)
```

Datenverwaltung mit Tabellen

In vielen Fällen ist es zweckmäßig, zur besseren Übersicht Daten in Tabellen darzustellen. Dafür wurde TADAT entwickelt, ein kompaktes, leistungsfähiges und anwendungsfreundliches Programm zur Verwaltung kleiner Datenmengen mit Hilfe von Tabellen. Es können maximal 170 Datensätze zu je zwölf Datenfeldern (hier Spalten) verarbeitet werden.

Verwendet man für kleine Datenmengen ein Dateiverwaltungsprogramm, das für größere zugeschnitten ist, so ist die Verarbeitungsdauer wegen der Umspeichervorgänge relativ lang.
Für kleine Datenmengen ist das Programm TADAT besser geeignet. Die gesamten Daten und die wichtigsten Programmroutinen befinden sich im RAM. Somit

können die Daten ohne zeitaufwendige Umspeicherung sofort bearbeitet werden. Lediglich beim Anlegen oder Verändern der Tabelle, bei der Festlegung der Spaltenverknüpfungen und beim Drucken müssen Programmroutinen von der Diskette in den Speicher geladen werden.
Das Programm ist so aufgebaut, daß eine externe Bedienungsanleitung nicht erforderlich ist. Der Computer führt Sie im Dialog durch das Programm. Der untere Teil des Bildschirmes enthält ein Menü mit den aktuellen Eingabemöglichkeiten.

1. Tabellen erstellen

Bevor eine Datei angelegt werden kann, müssen Sie zunächst eine Tabelle erstellen. Sie kann bis zu 12 Spalten erhalten. Die Spaltenbreiten und Spaltenüberschriften können frei gewählt werden. Sie können die Spaltenbreiten, Spaltenüberschriften und die Anzahl der Spalten auch nach dem Anlegen einer Datei ändern.

2. Daten eingeben, korrigieren, löschen und einfügen

In der Tabelle können Sie die Daten eingeben. Sie lassen sich auch korrigieren oder löschen. Es besteht auch die Möglichkeit, zwischen zwei anderen einen Datensatz einzufügen.
Achtung: Sind die Daten breiter als die Spalten, so werden die überzähligen Zeichen abgeschnitten. Um gekürzte Daten zu erkennen, wird auf dem Bildschirm die Spaltenbegrenzung überschrieben.

3. Arithmetische Verknüpfung von Spalten und Summenbildung

Die Spalten können untereinander arithmetisch verknüpft werden. Zwei Beispiele: Spalte 1 + Spalte 2 = Spalte 3 oder: Spalte 1 * 1.14 = Spalte 4. Es sind maximal drei arithmetische Verknüpfungen möglich. Die Werte in den Spalten können summiert werden. Das Ergebnis wird in einer Spalte angezeigt, zum Beispiel Summenbildung von Spalte 2 und das Ergebnis in Spalte 5. Es können bis zu zwei Summenspalten gewählt werden.
Achtung: Reicht die Spaltenbreite für die errechneten Werte nicht, so werden die überstehenden Zeichen abgeschnitten. Um gekürzte Daten zu erkennen, wird auf dem Bildschirm die Spaltenbegrenzung überschrieben.

4. Daten sortieren

Die Daten einer Spalte, ausgenommen die Summenspalten, können sortiert werden. Es besteht die Möglichkeit, nach Zahlenwerten oder nach Strings zu sortieren.

5. Datensätze suchen

Das Programm bietet auch die Möglichkeit, einzelne Daten in einer Spalte zu suchen. Zum Beispiel: alle Daten der Spalte zwei, die mit dem Buchstaben C anfangen, sollen gesucht werden.

6. Speicherplatz

Mit TADAT können maximal 170 Datensätze zu je zwölf Datenfeldern verwaltet werden.

7. Daten drucken

Die Tabelle mit den Daten können Sie sich ausdrucken lassen. Sie haben im Druckprogramm vielerlei Möglichkeiten, den Ausdruck zu gestalten, etwa Tabelle mit Rand, ohne Blattkopf und ohne Spaltenüberschrift. Sie können im Druckprogramm auch die Werte in den Spalten summieren lassen. Die Summe wird unterhalb der letzten Zeile ausgedruckt.

8. Erweiterungsmöglichkeiten

Das Programm TADAT ist übersichtlich strukturiert und enthält viele Unterprogramme. Ein Anwender mit BASIC-Kenntnissen kann es erweitern oder verbessern. Zusätzliche Erweiterungs-Programme könnten von TADAT mit CHAIN oder CHAIN MERGE aufgerufen werden, etwa ein Programm zur Darstellung der Daten in Balkendiagrammen.

9. Variablenliste

Die nachfolgende Liste enthält alle wichtigen Variablen:

usch$	Überschrift
datum$	Aktuelles Datum
stand$	Datum des letzten Zugriffes auf die Datei
seite	Seite der Bildschirmtabelle (angezeigt wird seite+1)
su$(n)	Spaltenüberschrift (n: 1–12)
sp(n,i)	X-Position der Spaltenbegrenzung (n: 1–12; i: 0–1)
nr	Spaltennummer (nr: 1–12)
lspnr	Anzahl der Spalten einer Tabelle (letzte Spaltennummer)
dat$(n,i)	Daten der Datei (n: 1–170; i: 1–12)
lznr	Letzte Zeilennummer einer Datei (letzter Datensatz)
name$	Mit name$ wird die Datei abgespeichert
xsp(n)	X-Position für die Dateneingabe, auch X-Position der Spaltenbegrenzung
spv(n)	Spaltennummern, deren Werte arithmetisch miteinander verknüpft werden
ar1$(n)	Arithmetisches Verknüpfungszeichen (n: 1–3)
kgr(n)	Konstante Größe der Spaltenverknüpfung
nkst(n)	Anzahl der Nachkommastellen
za	Zeilenanfang der

	Spaltenberechnung
ze	Zeilenende der Spaltenberechnung
zwsp(n)	Spaltennummern, deren Werte addiert werden
such$	Suchzeichen
suchsum(n)	Summenwerte einer Summenspalte beim Suchmodus
sdkor	sdkor=1: gesuchte Daten sollen korrigiert werden
kz	Korrektur-, Lösch- oder Einfügezeile
knr	Spaltennummer, ab der korrigiert werden soll.

10. Beispiel:

Aufgabe:
Es soll eine Tabelle mit fünf Spalten erstellt werden. Die Werte der Spalte zwei sollen mit den Werten der Spalte drei multipliziert werden. Zu diesem Wert ist die Zahl zehn zu addieren und das Ergebnis in Spalte vier darzustellen.
Die Werte der Spalte eins sollen summiert und in Spalte fünf dargestellt werden.

Lösung:
Starten Sie das Programm und geben Sie das Datum ein.
Im Hauptmenü müssen Sie die Betriebsart eins (Datei anlegen) wählen. Danach wird der Bildschirm neu aufgebaut. Die oberste Zeile ist reserviert für die Dateiüberschrift, den Stand der angelegten Datei, das aktuelle Datum und die aktuelle Seitenzahl. In Klammern wird die letzte mit Daten belegte Seitenzahl angezeigt.
Der mittlere Teil des Bildschirms ist für die Tabelle vorgesehen. In den untersten zwei Zeilen wird stets ein Menü mit den aktuellen Eingabemöglichkeiten angezeigt. In diesem Fall wird die Eingabe einer Überschrift gefordert. Falls Sie sie nicht korrigieren müssen, können Sie mit der Eingabe der Spaltenüberschriften beginnen. Der Cursor erscheint zu diesem Zweck an der Stelle, wo die erste Spaltenüberschrift stehen soll. Geben Sie diese ein und schließen Sie die Eingabe mit einem Punkt ab. An der Stelle des Punktes wird dann vom CPC ein senkrechter Strich gezogen. Möchten Sie einen Doppelstrich, ist ein Doppelpunkt einzugeben. Die Eingabe wird mit Enter abgeschlossen. Danach kann die zweite Spaltenüberschrift in gleicher Weise eingegeben werden.
Soll die Beschriftung der Spaltenüberschriften abgeschlossen werden, ist Enter zu drücken. Sie sehen dann, wie der Cursor zur Dateneingabe in der Spalte eins bereitsteht. Geben Sie aber zunächst noch keine Daten ein, sondern drücken Sie nochmals Enter. Im unteren Menü erkennen Sie jetzt die verschiedenen Eingabemöglichkeiten. Wählen Sie das Dienstmenü an (Taste sechs) und im Dienstmenü die Taste vier (Spaltenverknüpfung und Summenbildung). Auf dem Bildschirm erscheinen zwei Eingabetabellen. Die oberste Tabelle ist für die Spaltenverknüpfung, die unterste für die Summenbildung zuständig. Beginnen Sie mit der Spaltenverknüpfung. Der Cursor erscheint dann in der ersten Spalte der Eingabetabelle.
Gemäß der Aufgabenstellung soll folgende Rechnung durchgeführt werden:

```
(Spalte 2 * Spalte 3) + 10
  = Spalte 4
```

Sie müssen deshalb in der ersten Spalte der Eingabetabelle die Spaltennummer zwei eingeben. In der nächsten Spalte geben Sie das Multiplikationszeichen ein. In die dritte Spalte schreiben Sie die Zahl zwei (für Spalte), da ja Spalte zwei mit Spalte drei multipliziert werden soll.
Als nächstes geben Sie die Spaltennummer drei ein. Das Ergebnis der Multiplikation ist ein Zwischenwert und soll nicht dargestellt werden. Damit er für die weitere Verarbeitung zur Verfügung steht, muß er zunächst in der Spalte vier zwischengespeichert werden. Geben Sie deshalb in der fünften Spalte der Eingabetabelle die Zahl vier ein. In der letzten Spalte wählen Sie die Nachkommastellen. Nun ist die erste Verknüpfung fertig.
Da laut Aufgabenstellung zu diesem Ergebnis die Zahl zehn addiert werden soll, muß eine weitere Verknüpfung durchgeführt werden. Für die zweite Verknüpfung müssen Sie in der ersten Spalte der Eingabetabelle die Spaltennummer vier eingeben, da das in Spalte vier zwischengespeicherte Multiplikations-Ergebnis weiter verarbeitet werden soll. Geben Sie nacheinander das Pluszeichen, die Eins (für konstante Größe) und die Zahl zehn in die Eingabetabelle ein. In die fünfte Spalte schreiben Sie eine Vier, weil das Ergebnis in der Spalte vier dargestellt werden soll. Nach Eingabe der Nachkommastellen ist diese Spaltenverknüpfung fertig.
Gehen Sie dann durch Eingabe von Enter wieder in das Menü zurück. Weil noch eine Summenspalte angelegt werden soll, wählen Sie die Zwei im Menü. Der Cursor erscheint jetzt in der untersten Eingabetabelle. Laut Aufgabenstellung sollen die Werte der Spalte eins summiert und in Spalte fünf dargestellt werden. Geben Sie deshalb in der ersten Spalte eine Eins und in der zweiten Spalte eine Fünf ein.
Damit sind Summenbildung und Spaltenverknüpfung eingegeben. Gehen Sie mit Enter wieder in das Untermenü und betätigen Sie dann die Taste vier. Auf dem Bildschirm erscheint wieder die konstruierte Tabelle. Um dort die Daten einzugeben, betätigen Sie Taste drei.
Testen Sie die vielfältigen Möglichkeiten des Programmes durch, bevor Sie mit „echten" Daten anfangen. *ME*□

Datenverwaltung

```
10 '*****************************  (2397)
20 '*          TADAT            *  (2394)
30 '*           VON             *  (23B1)
40 '*      HEINZ BEHRENS        *  (23F6)
50 '*          FUER             *  (23B6)
60 '*   SCHNEIDER CPC-WELT      *  (23D8)
70 '*      CPC 664/6128      ME*  (2338)
75 '*****************************  (2318)
76 '                                (07A3)
80 GOSUB 4250:TAGOFF                (0B77)
90 LOCATE 7,8:PRINT"Bitte geben Si
e das Datum ein!"                   (2BA2)
100 CLEAR INPUT:LOCATE 7,12:INPUT"
TT.MM.JJ )",datum$                  (2420)
110 IF VAL(MID$(datum$,1,2))=0 OR
MID$(datum$,3,1)<>"."THEN 130       (330B)
120 IF MID$(datum$,6,1)<>"."THEN 1
30 ELSE 160                         (2244)
130 SOUND 2,12,20,15                (10E5)
140 LOCATE 7,12:PRINT SPACE$(20)    (127B)
150 GOTO 100                        (093C)
160 LOCATE 7,16:PRINT"Datum korrig
ieren (J/N)?"                       (26A3)
170 a$=INKEY$:a$=UPPER$(a$)         (1ABE)
180 IF a$=CHR$(13)THEN 280          (15FA)
190 ON INSTR(" JN",a$)GOTO 170,80,
280:GOTO 170                        (250A)
200 a=VAL(MID$(datum$,1,2)):b=VAL(
MID$(datum$,4,2)):c=VAL(MID$(datum
$,7,2))                             (4F82)
210 POKE 49600,a:POKE 49601,b:POKE
 49602,c                            (2BA3)
220 CLEAR                           (06D3)
230 a=PEEK(49600):b=PEEK(49601):c=
PEEK(49602)                         (3493)
240 IF a<10 THEN da$=MID$(STR$(a/1
00),4,2)ELSE da$=MID$(STR$(a),2,2)
                                    (3D04)
250 IF b<10 THEN db$=MID$(STR$(b/1
00),4,2)ELSE db$=MID$(STR$(b),2,2)
                                    (3D3D)
260 IF c<10 THEN dc$=MID$(STR$(c/1
00),4,2)ELSE dc$=MID$(STR$(c),2,2)
                                    (3DBB)
270 datum$=da$+CHR$(46)+db$+CHR$(4
6)+dc$                              (2D8D)
280 DIM su$(12),sp(12,1),dat$(170,
12),xsp(13),s(12)                   (3D62)
300 sp(0,0)=28                      (12BC)
310 GOSUB 4250                      (0950)
320 PRINT"Datum:";datum$;           (183D)
330 TAGOFF                          (06DC)
340 LOCATE 5,8:PRINT"Datei anlegen
       (1)"                         (26AD)
350 LOCATE 5,10:PRINT"Datei zeigen
/aendern  (2)"                      (2791)
370 LOCATE 5,12:PRINT"Programm bee
nden      (3)"                      (27E3)
380 LOCATE 5,17:PRINT"Bitte waehle
n        )>"                        (257C)
390 CLEAR INPUT                     (07D0)
400 hm$=INKEY$                      (0DF1)
410 ON INSTR(" 1234",hm$)GOTO 400,
430,430,420:GOTO 400                (2CAB)
420 CLS:END                         (085B)
430 MODE 2                          (070A)
440 WINDOW#1,5,80,3,3:WINDOW#2,1,8
0,24,25                             (2072)
450 WINDOW#3,5,80,6,22:WINDOW#4,1,
80,6,22                             (2027)
460 IF hm$="2"THEN 1040             (1392)
470 ON ERROR GOTO 490               (0B36)
475 GOSUB 4900                      (09F9)
480 CHAIN MERGE"MASKE.TPR",230,DEL
ETE-4410                            (1B68)
490 GOSUB 5000:GOTO 470             (0E69)
510 REM Daten eingeben              (159F)
520 ON ERROR GOTO 0:GOSUB 4790      (0B50)
530 seite=0                         (0F1B)
540 GOSUB 4670                      (0946)
550 PRINT#2,"Geben Sie die Daten e
in.   ""Eingaben beenden/Korrektur
en >>ENTER (in der Spalte 1)"       (5E29)
560 FOR z=1 TO 17                   (0FF1)
570 IF seite*17+z>170 THEN 660      (1D2F)
580 IF dat$(seite*17+z,1)<>""THEN
710                                 (27E9)
590 LOCATE 1,5+z:PRINT MID$(STR$(z
+seite*17),2)                       (293B)
600 FOR nr=1 TO lspnr               (16ED)
610 GOSUB 4100                      (09FC)
620 IF se=1 THEN 690                (11BB)
630 LOCATE#3,xsp(nr),z              (1B01)
640 LINE INPUT#3,"",dat$(seite*17+
z,nr)                               (2B2C)
650 IF dat$(seite*17+z,1)=""THEN 6
60 ELSE 680                         (2CAB)
660 IF seite*17+z-1>lznr THEN lznr
=seite*17+z-1                       (3B50)
670 GOSUB 3350:GOTO 1440            (0EBD)
680 dat$(seite*17+z,nr)=LEFT$(dat$
(seite*17+z,nr),xsp(nr+1)-xsp(nr))
                                    (64A3)
690 NEXT nr                         (0B3C)
700 za=seite*17+z:ze=za:GOSUB 3020
                                    (2C9F)
710 NEXT z                          (0A43)
720 seite=seite+1:GOTO 540          (1DD5)
730 REM      Daten speichern        (1AFF)
740 CLS#2:PRINT#2,"Unter welchem N
amen soll die Datei abgespeichert
werden?"                            (4771)
750 IF hm$="2"THEN LOCATE#2,58,1:P
RINT#2,"Alter Name:";UPPER$(name$)
                                    (36E7)
760 LOCATE#2,1,2:INPUT#2,"max. 8 Z
eichen (keine Satzzeichen)      Zu
rueck durch ENTER          ",a$     (5C26)
770 IF a$=""THEN 1420               (1165)
780 name$=UPPER$(a$)                (152D)
790 IF INSTR(name$,".TDT")=0 THEN
name$=name$+".TDT"                  (3134)
800 OPENOUT name$                   (0D70)
810 PRINT#9,usch$                   (10B4)
820 PRINT#9,datum$                  (11D8)
830 PRINT#9,lspnr,lznr              (1966)
840 FOR n=1 TO 9                    (0E2E)
850 PRINT#9,spv(n):NEXT             (1708)
860 FOR n=1 TO 3                    (0E3E)
870 PRINT#9,ar1$(n)                 (15EE)
880 PRINT#9,kgr(n),nkst(n):NEXT     (254C)
890 FOR n=1 TO 4                    (0E7E)
900 PRINT#9,zwsp(n):NEXT            (184F)
910 FOR nr=1 TO lspnr               (1659)
920 PRINT#9,sp(nr,0),sp(nr,1)       (2647)
930 PRINT#9,su$(nr)                 (1551)
940 NEXT nr                         (0B31)
950 FOR n=1 TO lznr                 (144B)
960 FOR nr=1 TO lspnr               (16BD)
970 PRINT#9,dat$(n,nr)              (1B7E)
980 NEXT nr,n                       (10E9)
990 CLOSEOUT                        (06E6)
1000 PRINT#2,"Daten gespeichert!":
PRINT#2,"Zurueck ins Hauptmenue (1
)    Daten nochmals betrachten (2)
"                                   (5FEE)
1010 CLEAR INPUT                    (07A9)
1020 a$=INKEY$                      (0C2E)
1030 ON INSTR(" 12",a$)GOTO 1020,2
00,1420:GOTO 1020                   (2575)
1040 REM     Daten lesen            (162C)
1050 CLS:PRINT"Datein auf der im S
tandardlaufwerk eingelegten Disket
te!"                                (42AC)
1060 |DIR,"*.TDT"                   (121C)
1070 MOVE 0,88:DRAW 640,88:MOVE 0,
25:DRAW 640,25                      (2064)
1080 LOCATE 2,21:PRINT"Falls die g
ewuenschte Datei auf der eingelegt
en Diskette nicht vorhanden ist,""
 wechseln Sie die Diskette und bet
aetigen Sie dann die ENTER-Taste."
" Zurueck ins Hauptmenue durch Ein
gabe von ++"                        (CE60)
1090 LOCATE#2,1,2:INPUT#2,"Name de
r Datei? > ",name$                  (2D9D)
1100 IF name$=""THEN 1050           (14D9)
1110 IF name$="++"THEN 200          (1649)
1120 IF INSTR(name$,".TDT")=0 AND
INSTR(name$,".tdt")=0 THEN name$=n
ame$+".TDT"                         (4624)
1130 ON ERROR GOTO 1150             (0B0B)
1140 GOTO 1180                      (09CF)
1150 GOSUB 5000:GOTO 1040           (0E73)
1180 OPENIN name$                   (0D7E)
1190 LINE INPUT#9,usch$             (11A1)
1200 INPUT#9,stand$                 (11EB)
1210 INPUT#9,lspnr,lznr             (19B6)
1220 FOR n=1 TO 9                   (0E26)
1230 INPUT#9,spv(n):NEXT            (1720)
1240 FOR n=1 TO 3                   (0E36)
1250 LINE INPUT#9,ar1$(n)           (1645)
1260 INPUT#9,kgr(n),nkst(n):NEXT    (25B1)
1270 FOR n=1 TO 4                   (0E76)
1280 INPUT#9,zwsp(n):NEXT           (188E)
1290 FOR nr=1 TO lspnr              (1654)
1300 INPUT#9,sp(nr,0),sp(nr,1)      (2625)
1310 LINE INPUT#9,su$(nr)           (16A5)
1320 NEXT nr                        (0B2C)
1330 FOR n=1 TO lznr                (1446)
1340 FOR nr=1 TO lspnr              (16B8)
1350 LINE INPUT#9,dat$(n,nr)        (1CE0)
1360 NEXT nr,n                      (10E4)
1370 CLOSEIN                        (06DB)
1380 GOSUB 4790                     (0997)
1420 REM     Daten zeigen           (177B)
1430 seite=0                        (0F25)
1440 ON ERROR GOTO 0:GOSUB 4670     (0BC0)
1450 PRINT#2,"    umblaettern (1)
       bestimmte Seite zeigen (2)
     Daten eingeben (3)""Daten ei
nfuegen (4)     Drucken (5)    Die
nstmenue (6)          Hauptmenue (
7)"                                 (A9E7)
1460 CLEAR INPUT                    (072F)
1470 a$=INKEY$                      (0CB3)
1480 ON INSTR(" 1234567",a$)GOTO 1
470,1720,1740,1790,1530,1484,1950,
1500:GOTO 1470                      (3E90)
1484 ON ERROR GOTO 1490             (0B75)
1485 GOSUB 4900                     (09E1)
1486 CHAIN"DRUCK.TPR"               (1187)
1490 GOSUB 5000:GOTO 1485           (0E05)
1500 PRINT#2," Achtung! Beim Rueck
sprung ins Hauptmenue werden die a
ktuellen Daten geloescht.  Sollen
die Daten vorher abgespeichert wer
den (J/N) ?"                        (90B6)
1510 a$=INKEY$:a$=UPPER$(a$)        (1A3C)
1520 ON INSTR(" JN",a$)GOTO 1510,7
30,200:GOTO 1510                    (25D7)
1530 IF lznr<170 THEN 1550          (143D)
1540 GOSUB 4360:GOTO 1450           (0EE8)
1550 PRINT#2," Zurueck durch ENTER
!"                                  (2003)
1560 INPUT#2," In welcher Zeile so
llen die Daten eingefuegt werden?
>",kz                               (481C)
1570 IF kz=0 THEN 1440              (1163)
1580 GOSUB 3950                     (09DB)
1590 IF y=0 THEN 1530               (100A)
1600 GOSUB 4150                     (094C)
1610 GOSUB 4670                     (09A6)
1620 PRINT#2," Geben Sie die Daten
 ein!   Zurueck durch ENTER in Spa
lte 1!"                             (4743)
1630 FOR nr=1 TO lspnr              (16FD)
1640 GOSUB 4100:IF se=1 THEN 1690   (1614)
1650 LOCATE#3,xsp(nr),kz-seite*17   (2818)
1660 LINE INPUT#3,"",dat$(kz,nr)    (20D5)
1670 IF dat$(kz,1)=""THEN 1700      (1CA4)
1680 dat$(kz,nr)=LEFT$(dat$(kz,nr)
,xsp(nr+1)-xsp(nr))                 (4E93)
1690 NEXT                           (06FD)
1700 za=kz:ze=kz:GOSUB 3020:GOSUB
3350                                (2660)
1710 GOTO 1440                      (0968)
1720 seite=seite+1                  (180F)
1730 GOTO 1440                      (0990)
1740 CLS#2                          (08A1)
1750 INPUT#2,"Welche Seite? > ",se
ite                                 (24BB)
```

```
1760 seite=seite-1                                    (1867)
1770 IF seite<0 THEN 1780 ELSE 144
0                                                     (1994)
1780 SOUND 2,12,20,15:CLS#2:PRINT#
2,"Die Seite"seite+1"gibt es nicht
!":GOTO 1750                                          (43D8)
1790 IF seite*17+z<=170 THEN 1810                     (1DC7)
1800 GOSUB 4360:GOTO 1450                             (0EF1)
1810 FOR z=1 TO 17                                    (0FBA)
1820 IF dat$(seite*17+z,1)=""THEN
550                                                   (27A6)
1830 NEXT                                             (0616)
1840 SOUND 2,12,20,15                                 (1048)
1850 PRINT#2,"Keine freien Zeilen
zur Dateneingabe auf dieser Seite
vorhanden!""naechste Seite (1)
   bestimmte Seite (2)       zurue
ck (3)"                                               (8B7E)
1860 CLEAR INPUT                                      (0752)
1870 a$=INKEY$                                        (0CD6)
1880 ON INSTR(" 123",a$)GOTO 1870,
1890,1920,1450:GOTO 1870                              (2AE7)
1890 seite=seite+1                                    (1863)
1900 GOSUB 4670                                       (09EB)
1910 GOTO 1790                                        (09EF)
1920 CLS#2:INPUT#2,"Welche Seite?
) ",seite                                             (28D5)
1930 seite=seite-1                                    (18BB)
1940 IF seite<0 THEN SOUND 2,12,20
,15:GOTO 1920 ELSE 1900                               (26FB)
1950 REM     Dienstmenue                              (1652)
1960 PRINT#2,"Ueberschrift/Spalten
/Daten korrigieren (1)     speiche
rn (2)     sortieren (3)  Spalten
verknuepfen u. Summenbildung   (4)
        suchen (5)       zurueck (
6)"                                                   (A90B)
1970 CLEAR INPUT                                      (072D)
1980 a$=INKEY$                                        (0CB1)
1990 ON INSTR(" 123456",a$)GOTO 19
80,2000,730,2020,2350,2010,1450:GO
TO 1980                                               (3928)
2000 GOSUB 2410:GOTO 1450                             (0EE8)
2010 GOSUB 3500:GOTO 1450                             (0E1F)
2020 REM     Sortieren                                (14DD)
2030 CLS#2                                            (08E6)
2040 IF zwsp(3)=0 AND zwsp(4)=0 TH
EN 2060                                               (23D6)
2050 PRINT#2,"Die Daten in den Sum
menspalten koennen nicht sortiert
werden!"                                              (4807)
2060 INPUT#2,"In welcher Spalte so
llen die Daten sortiert werden?
Spaltennummer )",sortsp                               (5A6D)
2070 IF sortsp=0 THEN 1950                            (1551)
2080 IF sortsp<0 OR sortsp)lspnr O
R sortsp=zwsp(3)OR sortsp=zwsp(4)T
HEN SOUND 2,12,20,15:GOTO 2030                        (5FD0)
2090 PRINT#2,"Sollen Zahlenwerte s
ortiert werden (1)   oder Strings
(2) ?          )"                                     (503A)
2100 a$=INKEY$                                        (0CA3)
2110 IF a$="1"OR a$="2"THEN 2120 E
LSE 2100                                              (2058)
2120 IF a$="1"THEN ON ERROR GOTO 2
330                                                   (1579)
2130 teil=lznr:e=0                                    (1B48)
2140 CLS#2:PRINT#2," Bitte warten!
"                                                     (1DF8)
2150 WHILE NOT e                                      (0BF4)
2160 teil=INT(teil/2)                                 (1AFD)
2170 IF teil<=1 THEN e=-1                             (1787)
2180 FOR i=1 TO lznr-teil                             (1C92)
2190 FOR j=i TO 1 STEP-teil                           (1A97)
2200 IF a$="2"THEN 2220                               (12E3)
2210 IF VAL(dat$(j+teil,sortsp))>=
VAL(dat$(j,sortsp))THEN j=1:GOTO 2
280 ELSE 2230                                         (548E)
2220 IF dat$(j+teil,sortsp))=dat$(
j,sortsp)THEN j=1:GOTO 2280                           (475E)
2230 FOR k=1 TO lspnr                                 (15F9)
2240 hilf$=dat$(j+teil,k)                             (26CE)
2250 dat$(j+teil,k)=dat$(j,k)                         (305B)
2260 dat$(j,k)=hilf$                                  (1EBF)
2270 NEXT                                             (0686)
2280 NEXT j,i                                         (0FD6)
2290 WEND                                             (0643)
2300 ON ERROR GOTO 0                                  (06D2)
2310 GOSUB 3350:GOSUB 4670                            (0E9C)
2320 GOTO 1450                                        (0980)
2330 PRINT#2,"Achtung!  Sortiervor
gang musste abgebrochen werden, we
il die Spalte";sortsp;"Stringgroes
sen enthaelt.     Weiter durch Be
tatigen einer Taste!"                                 (9CC4)
2340 IF INKEY$<>""THEN RESUME 2300
 ELSE 2340                                            (1504)
2350 REM     Spalten verknuepfen u
nd                                                    (222D)
2360 REM     Summenbildung                            (185E)
2370 ON ERROR GOTO 2390                               (0B92)
2375 GOSUB 4900                                       (09DA)
2380 CHAIN"VERKNUPF.TPR"                              (146F)
2390 GOSUB 5000:GOTO 2380                             (0E1E)
2410 REM     UP Korrekturen                           (1984)
2420 CLEAR INPUT:CLS#2                                (0B30)
2430 IF sdkor<>1 THEN 2480                            (149F)
2440 PRINT#2,"Daten korrigieren (1
)   Datensatz loeschen (2)   zurue
ck (3)"                                               (4744)
2450 a$=INKEY$                                        (0C5F)
2460 ON INSTR(" 123",a$)GOTO 2450,
2510,2850,2470:GOTO 2450                              (2AC1)
2470 sdkor=0:RETURN                                   (1118)
2480 PRINT#2,"Ueberschrift korrigi
eren (1)    Spalten aendern (2)
Daten korrigieren (3)     Daten lo
eschen          (4)    Zurueck
     (5)"                                             (8E3B)
2490 a$=INKEY$                                        (0CAF)
2500 ON INSTR(" 12345",a$)GOTO 249
0,2800,2710,2510,2850,2470:GOTO 24
90                                                    (349E)
2510 CLS#2:INPUT#2,"Zurueck durch
ENTER      Zeilennummer der Korrek
turgroesse ) ",kz                                     (52D9)
2520 IF kz=0 THEN 2410                                (1130)
2530 GOSUB 3950                                       (094A)
2540 IF y=0 THEN 2510                                 (1029)
2550 PRINT#2,"Spaltenwert korrigie
ren (k)      Keine korrekturen (EN
TER)"                                                 (45B1)
2560 FOR nr=1 TO lspnr                                (1645)
2570 GOSUB 4100:IF se=1 THEN 2660                     (16BB)
2580 LOCATE xsp(nr)+4,y+5                             (1CF2)
2590 CURSOR 1,1                                       (0911)
2600 a$=INKEY$:a$=UPPER$(a$)                          (1AC5)
2610 IF a$=CHR$(13)THEN 2660                          (1580)
2620 IF a$="K"THEN 2630 ELSE 2600                     (17FA)
2630 LOCATE#3,xsp(nr),y:PRINT#3,ST
RING$(xsp(nr+1)-xsp(nr)-1,32)                         (46A7)
2640 LOCATE#3,xsp(nr),y:LINE INPUT
#3,"",dat$(kz,nr)                                     (3753)
2650 dat$(kz,nr)=LEFT$(dat$(kz,nr)
,xsp(nr+1)-xsp(nr))                                   (4E2B)
2660 NEXT                                             (0695)
2670 CURSOR 0                                         (072C)
2680 za=kz:ze=kz:GOSUB 3020:GOSUB
3350                                                  (260C)
2690 IF sdkor=0 THEN GOSUB 4670:GO
TO 2510                                               (1A49)
2700 sdkor=0:RETURN                                   (11E4)
2710 IF sp(lspnr,0)>635 THEN CLS#2
:GOTO 2730                                            (24CD)
2720 PRINT#2,"Neue Spalten hinzufu
egen  )>  ("lspnr+1")"                                (37C6)
2730 INPUT#2,"     Spalten korrigi
eren  )>(Spaltennr.) ",knr2                           (3C79)
2740 IF knr2=0 THEN RETURN                            (115D)
2750 IF knr2)lspnr+1 OR knr2<0 OR(
sp(lspnr,0)>635 AND knr2)lspnr)THE
N SOUND 2,12,20,15:GOTO 2710                          (5C22)
2760 ON ERROR GOTO 2820                               (0B1C)
2765 GOSUB 4900                                       (09E6)
2770 CHAIN MERGE"MASKE.TPR",40,DEL
ETE-4410                                              (1B87)
2780 ON ERROR GOTO 0                                  (0694)
2790 GOSUB 4670:GOTO 1450                             (0EEE)
2800 ON ERROR GOTO 2820                               (0B6C)
2805 GOSUB 4900                                       (0937)
2810 CHAIN MERGE"MASKE.TPR",190,DE
LETE-4410                                             (1B6D)
2820 GOSUB 5000:GOTO 2840                             (0EE1)
2840 IF a$="1"THEN 2800 ELSE 2770                     (1791)
2850 PRINT#2,"Geben Sie die Zeilen
nummer der zu loeschenden Zeile ei
n!"                                                   (4327)
2860 INPUT#2,"
        Zurueck durch ENTER
 )> ",kz                                              (4B5F)
2870 IF kz=0 THEN RETURN                              (0F26)
2880 GOSUB 3950                                       (0908)
2890 IF y=0 THEN 2850                                 (1092)
2900 FOR z=kz TO lznr-1                               (1AF5)
2910 FOR s=1 TO lspnr                                 (154F)
2920 dat$(z,s)=dat$(z+1,s)                            (2A22)
2930 NEXT                                             (06B2)
2940 NEXT                                             (06C6)
2950 FOR s=1 TO lspnr                                 (159F)
2960 dat$(lznr,s)=""                                  (1CDF)
2970 NEXT                                             (0602)
2980 lznr=lznr-1                                      (16BA)
2990 GOSUB 3350                                       (0921)
3000 IF sdkor=0 THEN GOSUB 4670:GO
TO 2850                                               (1A57)
3010 sdkor=0:RETURN                                   (1152)
3020 REM     UP Werte in den Spalt
en                                                    (2201)
3030 REM     berechnen                                (1446)
3040 FOR n=1 TO 3                                     (0E4D)
3050 IF spv(n)=0 THEN 3330                            (18C2)
3060 IF ze-za>5 THEN CLS#2:PRINT#2
," Bitte warten!"                                     (2C17)
3070 FOR zs=za TO ze                                  (1745)
3080 IF dat$(zs,1)=""THEN 3325                        (1C8A)
3090 ON ERROR GOTO 3155                               (0B29)
3100 f1=VAL(dat$(zs,spv(n)))                          (2930)
3130 ON ERROR GOTO 3160                               (0BA1)
3140 f2=VAL(dat$(zs,spv(n+3)))                        (2B18)
3150 ON ERROR GOTO 0:GOTO 3170                        (0B4A)
3155 f1=0:RESUME 3130                                 (1160)
3160 f2=0:RESUME 3150                                 (1106)
3170 IF kgr(n)=0 THEN 3250                            (1821)
3180 IF ar1$(n)="+"THEN 3220                          (1A63)
3190 IF ar1$(n)="-"THEN 3230                          (1AC6)
3200 IF ar1$(n)="*"THEN 3240                          (1AAB)
3210 dat$(zs,spv(n+6))=STR$(ROUND(
f1/kgr(n),nkst(n))):GOTO 3320                         (4FDF)
3220 dat$(zs,spv(n+6))=STR$(ROUND(
f1+kgr(n),nkst(n))):GOTO 3320                         (4FAA)
3230 dat$(zs,spv(n+6))=STR$(ROUND(
f1-kgr(n),nkst(n))):GOTO 3320                         (4F35)
3240 dat$(zs,spv(n+6))=STR$(ROUND(
f1*kgr(n),nkst(n))):GOTO 3320                         (4F62)
3250 IF ar1$(n)="+"THEN 3290                          (1A18)
3260 IF ar1$(n)="-"THEN 3300                          (1A7D)
3270 IF ar1$(n)="*"THEN 3310                          (1A60)
3280 dat$(zs,spv(n+6))=STR$(ROUND(
f1/f2,nkst(n))):GOTO 3320                             (481D)
3290 dat$(zs,spv(n+6))=STR$(ROUND(
f1+f2,nkst(n))):GOTO 3320                             (4893)
3300 dat$(zs,spv(n+6))=STR$(ROUND(
f1-f2,nkst(n))):GOTO 3320                             (48B7)
3310 dat$(zs,spv(n+6))=STR$(ROUND(
f1*f2,nkst(n))):GOTO 3320                             (48B7)
3320 dat$(zs,spv(n+6))=LEFT$(dat$(
zs,spv(n+6)),xsp(spv(n+6)+1)-xsp(s
pv(n+6)))                                             (721A)
3325 NEXT zs                                          (0B41)
3330 NEXT n                                           (0A96)
3340 RETURN                                           (064C)
3350 REM     UP Spaltenaddition                       (1DBF)
3360 FOR n=1 TO 2                                     (0ECB)
3370 IF zwsp(n)=0 THEN 3480                           (1991)
3380 IF lznr>10 THEN CLS#2:PRINT#2
," Bitte warten!"                                     (2956)
3390 FOR zn=1 TO lznr                                 (15EE)
3400 IF dat$(zn,1)=""THEN f4=f4+VA
L(dat$(zn-1,zwsp(n+2))):GOTO 3470                     (4D1A)
3410 ON ERROR GOTO 3440                               (0B95)
3420 f3=VAL(dat$(zn,zwsp(n)))                         (2A33)
3430 ON ERROR GOTO 0:GOTO 3450                        (0B42)
```

Datenverwaltung mit Tabellen

```
3440 f3=0:RESUME 3430                    <11F6>
3450 dat$(zn,zwsp(n+2))=STR$(f3+f4
+VAL(dat$(zn-1,zwsp(n+2))))              <5621>
3455 dat$(zn,zwsp(n+2))=LEFT$(dat$
(zn,zwsp(n+2)),xsp(zwsp(n+2)+1)-xs
p(zwsp(n+2)))                            <76BE>
3460 f4=0                                <0CDA>
3470 NEXT zn                             <0B4F>
3480 NEXT n                              <0AC1>
3490 RETURN                              <0677>
3500 REM     UP Suchprogramm             <1A3E>
3510 seiteh=seite                        <1701>
3520 CLS#2:LOCATE#2,20,1:INPUT#2,"
Geben Sie das/die Suchzeichen ein!
 ))",such$                               <45A9>
3530 INPUT#2,"In welcher Spalte so
ll gesucht werden?  Spaltennummer
))",spn                                  <4A27>
3540 seite=0:zws1=0:zws2=0:sanf=1        <2DE5>
3550 suchsum(1)=zws1:suchsum(2)=zw
s2:zh=1:y=0                              <3F77>
3560 GOSUB 4420                          <0915>
3570 FOR sz=sanf TO lznr                 <1BBF>
3580 IF MID$(dat$(sz,spn),1,LEN(su
ch$))=such$THEN dat$(sz,0)="x":GOT
O 3670 ELSE dat$(sz,0)=""                <60E4>
3590 NEXT                                <06DC>
3600 PRINT#2,"Suchzeichen: ";such$
,"Spaltennummer: ";spn"Daten nochm
als betrachten (1)  zurueck (2)  D
aten korrigieren (3)  Drucken (4)"
                                         <8930>
3610 CLEAR INPUT                         <0704>
3620 a$=INKEY$                           <0C89>
3630 ON INSTR(" 1234",a$)GOTO 3620
,3540,3660,3640,3631:GOTO 3620           <2FEE>
3631 GOSUB 4900                          <09AF>
3632 ON ERROR GOTO 3634                  <0B6C>
3633 CHAIN"DRUCK.TPR"                    <1157>
3634 GOSUB 5000:GOTO 3631                <0E05>
3640 sdkor=1:GOSUB 2410                  <1431>
3650 IF y()0 THEN 3550 ELSE 3600         <159B>
3660 seite=seiteh:GOSUB 4670:RETUR
N                                        <1ECF>
3670 IF zh=18 THEN 3680 ELSE 3740        <1782>
3680 PRINT#2,"Suchzeichen: ";such$
,"Spaltennummer: ";spn"umblaettern
 (1) zurueckblaettern (2) zurueck
(3) Daten korrigieren/loeschen (4)
"                                        <8A29>
3690 CLEAR INPUT                         <07A5>
3700 a$=INKEY$                           <0C28>
3710 ON INSTR(" 1234",a$)GOTO 3700
,3720,3540,3660,3930:GOTO 3700           <2FBC>
3720 seite=seite+1:zh=1                  <2049>
3730 zws1=suchsum(1):zws2=suchsum(
2):GOSUB 4420                            <3574>
3740 LOCATE 1,5+zh:PRINT MID$(STR$
(sz),2)                                  <1FCE>
3750 IF zwsp(1)=0 AND zwsp(2)=0 TH
EN 3830                                  <233E>
3760 FOR sh=1 TO 2                       <0FB8>
3770 ON ERROR GOTO 3800                  <0BAE>
3780 f=VAL(dat$(sz,zwsp(sh)))            <2AA3>
3790 ON ERROR GOTO 0:GOTO 3810           <0B59>
3800 f=0:RESUME 3790                     <1012>
3810 suchsum(sh)=suchsum(sh)+f           <2DBA>
3820 NEXT                                <06A8>
3830 FOR nr=1 TO lspnr                   <1635>
3840 LOCATE#3,xsp(nr),zh                 <1CD0>
3850 IF zwsp(3)=nr AND zwsp(4)=nr
THEN sh=2:GOTO 3900                      <3449>
3860 IF zwsp(3)=nr THEN sh=1:GOTO
3900                                     <23AA>
3870 IF zwsp(4)=nr THEN sh=2:GOTO
3900                                     <235B>
3880 PRINT#3,dat$(sz,nr)                 <1CF6>
3890 GOTO 3910                           <09CF>
3900 PRINT#3,suchsum(sh)                 <1ABC>
3910 NEXT                                <065E>
3920 zh=zh+1:GOTO 3590                   <17A7>
3930 sdkor=1:GOSUB 2410                  <1476>
3940 IF y()0 THEN 3550 ELSE 3680         <1561>
3950 REM     UP Y-Position einer Z
eile                                     <24D9>
3960 FOR y=1 TO 17                       <0F0E>
3970 zeil$=""                            <0FB7>
3980 FOR n=1 TO 3                        <0EAA>
3990 LOCATE n,y+5                        <114B>
4000 zeil$=zeil$+COPYCHR$(#0)            <1B40>
4010 NEXT n                              <0AE7>
4020 IF y=1 THEN sanf=VAL(zeil$)         <2032>
4030 IF kz=VAL(zeil$)THEN RETURN         <19A7>
4040 NEXT                                <0661>
4050 SOUND 2,12,20,15                    <1093>
4060 CLS#2:PRINT#2,"Die Zeilennumm
er"kz"ist nicht auf dieser Seite!
  Taste betaetigen!"                     <552B>
4070 IF INKEY$()""THEN 4080 ELSE 4
070                                      <141E>
4080 y=0                                 <0BB4>
4090 RETURN                              <062A>
4100 REM     UP Spalten von der Ei
ngabe                                    <25B3>
4110 REM     ausschliessen               <1802>
4120 se=0                                <0CB9>
4130 IF(zwsp(3)=nr OR zwsp(4)=nr O
R spv(7)=nr OR spv(8)=nr OR spv(9)
=nr)AND nr)1 THEN se=1                   <6991>
4140 RETURN                              <068F>
4150 REM     UP Zeilen einfuegen         <1ED9>
4160 FOR z=lznr+1 TO kz STEP-1           <1D51>
4170 FOR s=1 TO lspnr                    <152C>
4180 dat$(z,s)=dat$(z-1,s)               <2AFC>
4190 NEXT:NEXT                           <089C>
4200 FOR s=1 TO lspnr                    <1568>
4210 dat$(kz,s)=""                       <1A9A>
4220 NEXT                                <06CB>
4230 lznr=lznr+1                         <167B>
4240 RETURN                              <0656>
4250 REM     UP Eingangsbild             <1ABA>
4260 MODE 1:BORDER 9                     <0A1D>
4270 INK 2,0:INK 3,10                    <0FE7>
4280 ORIGIN 19,387,0,640,400,365         <1A97>
4290 CLG 1:GRAPHICS PEN 2                <0B2F>
4300 TAG:PRINT"Datenverwaltung mit
 Hilfe von Tabellen";                    <31E5>
4310 ORIGIN 55,352,0,640,365,330         <1A2F>
4320 CLG 3:GRAPHICS PEN 1                <0BE5>
4330 PRINT CHR$(164);" by Heinz Be
hrens        1988";                      <2F48>
4340 ORIGIN 1,16,0,640,90,0:CLG 3        <1827>
4350 RETURN                              <0633>
4360 REM     UP Speicher voll            <1B18>
4370 SOUND 2,12,20,15                    <1016>
4380 PRINT#2,"Weitere Datensaetze
koennen nicht eingegeben werden.-M
it diesem Programm        koennen
max. 170 Datensaetze verarbeitet w
erden!    Beliebige Taste betaetig
en!"                                     <AAD4>
4390 CLEAR INPUT                         <0720>
4400 IF INKEY$=""THEN 4400               <0F05>
4410 RETURN                              <06AC>
4420 REM     UP Tabelle zeichnen         <1E7B>
4430 CLS                                 <06D7>
4440 ORIGIN 0,0,1,640,400,384:CLG
1                                        <1A87>
4450 PAPER 1:PEN 0:LOCATE 1,1:PRIN
T usch$;TAB(34)"Stand:"stand$;TAB(
50)"Datum:"datum$;TAB(66)"Seite:";
seite+1;"(";MID$(STR$(INT((lznr-0.
5)/17)+1),2);")"                         <8827>
4460 PAPER 0:PEN 1                       <0A84>
4470 ORIGIN 0,0,0,640,400,0              <1516>
4480 MOVE 0,40:DRAW 640,40               <1286>
4490 MOVE 0,328:DRAW 640,328             <14E1>
4500 LOCATE 1,3:PRINT"Z-"                <0F37>
4510 MOVE 26,40:DRAW 26,383:MOVE 2
7,383:DRAW 27,40:MOVE 28,40:DRAW 2
8,383:MOVE 29,383:DRAW 29,40             <4002>
4520 LOCATE 1,4:PRINT"Nr"                <0FC1>
4530 PRINT#1                             <08C5>
4540 FOR nr=1 TO lspnr                   <16C3>
4550 IF nr=lspnr THEN PRINT#1,su$(
nr);:GOTO 4570                           <2B85>
4560 PRINT#1,su$(nr);" ";                <1AA4>
4570 GOSUB 4620                          <0943>
4580 MOVE sp(nr,0),40:DRAW sp(nr,0
),366                                    <2B41>
4590 MOVE sp(nr,1),40:DRAW sp(nr,1
),366                                    <2BC1>
4600 NEXT                                <06C3>
4610 RETURN                              <063D>
4620 REM     UP Spaltennummer zeic
hnen                                     <2420>
4630 xsnr=INT((sp(nr-1,0)+7)/8)+(I
NT((sp(nr,0)-sp(nr-1,0))/8))/2           <5305>
4640 IF(nr)1 AND sp(nr,0)=28)OR xs
nr=4 THEN RETURN                         <2DD2>
4650 LOCATE xsnr,4:PRINT MID$(STR$
(nr),2)                                  <1F77>
4660 RETURN                              <06A1>
4670 REM     UP Listen einer Seite
                                         <204C>
4680 GOSUB 4420                          <09DA>
4690 FOR z=1 TO 17                       <0F44>
4700 IF seite*17+z)170 THEN PRINT#
3,"Das Programm verarbeitet maxima
l 170 Datensaetze!""Weitere Eingab
en sind nicht moeglich.":RETURN          <7A4A>
4710 LOCATE 1,5+z:PRINT MID$(STR$(
z+seite*17),2)                           <297B>
4720 FOR nr=1 TO lspnr                   <162C>
4730 LOCATE#3,xsp(nr),z                  <1B18>
4740 PRINT#3,dat$(seite*17+z,nr)         <2753>
4750 NEXT                                <06F1>
4760 IF seite*17+z)=lznr THEN RETU
RN                                       <2074>
4770 NEXT z                              <0A0A>
4780 RETURN                              <0690>
4790 REM     UP Cursor-Position f.
 die                                     <241C>
4800 REM     Dateneingabe                <178E>
4810 FOR n=1 TO lspnr+1                  <1714>
4820 xsp(n)=INT(sp(n-1,0)/8)-2           <291E>
4830 NEXT                                <0690>
4840 RETURN                              <0608>
4900 REM UP Programmdiskette einle
gen                                      <2383>
4910 CLS#2:PRINT#2,"Programmdisket
te einlegen und anschliessend eine
 Taste betaetigen!"                      <511D>
4920 IF INKEY$()""THEN RETURN ELSE
 4920                                    <12CE>
5000 REM UP Fehlermeldung                <178F>
5010 CLS#2:PRINT#2,"Diskettenfehle
r!  Fehler beheben und anschliesse
nd eine Taste betaetigen!"               <588B>
5020 IF INKEY$()""THEN RESUME 5030
 ELSE 5020                               <1531>
5030 RETURN                              <0685>
```




Haben Sie Probleme? CPC Welt beantwortet Ihre Fragen
Postfach 1161
D 8044 Lohhof

Ordnung leicht gemacht

Sicher kennen Sie die Situation: Sie wußten genau, daß Sie auf irgendeiner Diskette das Programm noch hatten – oder haben Sie es doch schon auf Kassette ausgelagert? Nach stundenlangem Suchen haben Sie es gefunden. Und haben zum soundsovielten Mal beschlossen, Ordnung zu halten. Aber wie?

Unser Programm ARCHIVATOR bringt System in Ihre Diskettensammlung. Hier eine Übersicht der Leistungsdaten:

- 223 Diskseiten mit bis zu 935 Dateien können erfaßt werden;
- Sortieren durch MCode, dadurch nur kurze Wartezeiten;
- Beschleunigte Bildschirmausgabe;
- Erfassen der Dateien direkt aus dem CATalog. Dabei Abfangen von Diskfehlern;
- Manuelles Erfassen und Ändern des Archivs möglich;
- Ausgabe auf Bildschirm oder Drucker. Steuerzeichen-Sequenzen sind frei definierbar;
- Drei Listen können erstellt werden: alphabetisch nach Dateinamen, nach Seiten geordnet und eine Übersicht über die Disknamen;
- Suchen nach Dateinamen;
- Diskoperationen wie Löschen und Umbenennen vom Programm aus, dabei sind Massenoperationen möglich;
- Verwendung als Kassettenarchiv, zum Beispiel für Backup-Kassetten.

Während andere Archivprogramme verlangen, daß man zu den Dateinamen noch andere Daten wie etwa die Länge, wie man sie startet, ihre Bewertung und das Aufnahmedatum eingibt, wurde bei ARCHIVATOR darauf verzichtet. Dieses Drumherum hindert erfahrungsgemäß nur daran, ein Archiv auf dem neuesten Stand zu halten, weil die Eingaben per Hand zuviel Arbeit machen.

Das Eintippen

Sie benötigen einen CPC mit einem oder zwei Laufwerken. Vortex-Laufwerke sind nicht geeignet.
Tippen Sie **Listing 1 (ARCH.BAS)** ein und speichern Sie es auf Diskette ab. Dann tippen Sie **Lsting 2 (ARCH.LAD)** und machen eine Sicherheitskopie auf Diskette. Lassen Sie die Diskette im Laufwerk. Starten Sie ARCH.LAD. Wenn alles in Ordnung ist, speichert der Hexlader ein Binärfile mit dem Namen ARCH.BIN auf Diskette. Sie können jetzt den Hexlader löschen. ARCHIVATOR besteht aus den beiden Teilen ARCH.BAS und ARCH.BIN.

Die Bedienung

1. Starten

Sie starten ARCHIVATOR mit RUN"ARCH". Wenn Sie ein bereits erstelltes Archiv bearbeiten wollen, das denselben Namen wie der Default-Wert (oben links angezeigt) trägt, drücken Sie <L>. Wollen Sie ein anders benanntes Archiv bearbeiten, müssen Sie vorher den Default-Namen ändern (<N>).

2. CATALOG

Das Herzstück des Programms ist die Option „Catalog". Wenn Sie die Taste <C> drücken, zeigt Ihnen das Programm das Inhaltsverzeichnis der eingelegten Diskette an. Dieses können Sie jetzt ganz oder teilweise in das Archiv aufnehmen und Einträge aus ihm löschen oder umbenennen. Dazu müssen Sie die Einträge markieren und danach die Taste der jeweiligen Funktion drücken.
Einträge werden mit den Cursortasten ausgewählt und mit <COPY> markiert. Mit <DEL> wird die Markerung rückgängig gemacht. Außerdem stehen Ihnen folgende Hilfen zur Verfügung:
<I>: Alle Markierungen invertieren, das heißt, Markiertes wird unmarkiert und Unmarkiertes markiert.
<K>: Alle Markierungen löschen.
<G> Alle Einträge markieren.
<L> (Löschen), <U> (Umbenennen) und <A> (Archivieren) be-

Ordnung leicht gemacht

ziehen sich auf die markierten Dateien. Wenn Sie sie archivieren wollen, werden Sie aufgefordert, den Namen der Diskseite einzugeben.
Ein Tip zu den Namen: ARCHIVATOR sortiert alphabetisch. Das heißt, daß „Spiele 11" vor „Spiele 2" kommt. Diesen Effekt können Sie vermeiden, wenn Sie die Zahlen stellenrichtig eingeben, also führende Stellen mit Blanks auffüllen: „Spiele<Blank>11" und „Spiele<2 Blanks>2". Damit wird so sortiert, wie Sie es erwarten.
Manche Disketten sind fremdorientiert und enthalten keine von AMSDOS aus lesbare Directory. Den Lesefehler, der dadurch entsteht, fängt ARCHIVATOR ab. Genauso den Fehler, der auftritt, wenn keine Diskette im Laufwerk ist. Sie haben dadurch die Möglichkeit, nur den Disknamen in das Archiv aufzunehmen. Das wäre eine Diskettenseite, die null Dateien enthält und null K-Byte frei hat. Sie können aber auch abbrechen, also die Diskseite übergehen und eine neue Diskette einlegen oder noch einen Leseversuch starten. Solche fremdformatierten Diskettenseiten sollten Sie per Hand erfassen. Sollten Sie eine Diskseite zum zweiten Mal eingeben, bemerkt ARCHIVATOR das und fragt, ob Sie den alten Inhalt überschreiben wollen. Wenn nicht, müssen Sie einen neuen Disknamen eingeben. Zwei Diskseiten können nicht denselben Namen haben; zwei Dateien aber wohl.

3. Manuelles Erfassen

Diesen Punkt können Sie mit <M> aus dem Hauptmenü wählen. Manuelles Erfassen kommt in Frage, wenn Sie eine fremdformatierte Diskette archivieren wollen oder ARCHIVATOR als Kassettenarchiv benutzen. Eine Leereingabe, also <ENTER> ohne vorhergehende Zeichen, bricht die Option ab.

4. Ausgabe des Archivs

Es gibt drei Formen, in der das Archiv ausgegeben werden kann.
a) **Gesamtliste:** In ihr werden alle Dateinamen alphabetisch geordnet ausgegeben und angezeigt, auf welcher Diskseite sich die Datei befindet. In dieser Form werden Sie ARCHIVATOR wohl am häufigsten benutzen. Mit <G> erhalten Sie die Gesamtliste auf dem Bildschirm. Mit den Cursortasten können Sie in ihr blättern. Mit <SHIFT> und den Rechts-/Links-Cursortasten läßt sich das Blättern beschleunigen. Außerdem können Sie Dateinamen aus dem Archiv löschen oder sie ändern. Die Zeit, die ARCHIVATOR braucht, einen Namen neu einzusortieren, werden Sie kaum bemerken.
b) **Seitenliste:** In ihr werden alle Diskseiten, alphabetisch geordnet, aufgeführt. Dazu wird angegeben, wieviel KBytes auf der jeweiligen Seite noch frei und welche Dateien dort abgespeichert sind. Hier kann wie in der Gesamtliste geblättert werden.
c) **Übersicht:** Hier werden die Diskseiten ohne die Dateien aufgeführt. Sinn dieser Liste ist es, sich über den noch verfügbaren Speicherplatz auf den Disketten zu orientieren. Außerdem lassen sich hier Diskseiten aus dem Archiv löschen.
Alle drei Listen können auch auf dem Drucker ausgegeben werden. Dazu müssen Sie vom Hauptmenü aus das Druckmenü wählen (<A>). Die Druckroutine ist für den NLQ 401 und Zwölf-Zoll-Endlospapier eingerichtet. Sie können aber über den Punkt <C> eigene Werte festlegen. Wenn Sie <C> drücken, werden Sie gebeten, einen Steuerstring anzugeben. Dieser String hat die Syntax:
<Steuerzeichen <,Steuerzeichen <...>>>;Zeilenbreite <;Zeilen pro Seite>
Die Angaben in spitzen Klammern können weggelassen werden. Achten Sie genau darauf, wann Kommas und wann Semikolons gesetzt werden müssen. Die Angabe der Zeilenbreite ist zwingend. Wenn nur sie eingegeben werden soll, müssen Sie ein Semikolon voransetzen, sonst würde das Programm den eingegebenen Wert als Steuerzeichen interpretieren.
Die oben angesprochene Voreinstellung hat als Steuerstring folgende Form:
27,120,0,15,27,67,0,12;120;66
Das bedeutet:
CHR$(27)+CHR$(120)+
CHR$(0): Entwurfsdruck ein
CHR$(15): Verdichtete Schrift ein
CHR$(27)+CHR$(67)+CHR $(0)
CHR$(12): Zwölf-Zoll-Formular
Zeilenbreite 120
66 Zeilen pro Seite

5. Weitere Kommandos

<F>: Suchen nach einem Dateinamen. Auf Groß- und Kleinschreibung kommt es dabei nicht an. Gesucht wird nach dem Dateinamen, der das eingegebene Suchwort enthält.
<D>: Bezugslaufwerk wird B beziehungsweise wieder A.
<Z>: Programm wird neu gestartet – alle Daten gehen verloren.
<Q>: Rückkehr zum BASIC.
<K>: Umschalten zwischen Benutzung als Disk- und Kassettenarchiv. Wenn ARCHIVATOR als Kassettenarchiv verwendet wird, wird die Option <C> gesperrt, keine Angabe des freien Speicherplatzes pro Seite verlangt und bei der Eingabe der Dateinamen zwischen Klein- und Großschreibung unterschieden. Dateinamen können dann bis zu zwölf Zeichen lang sein. Beachten Sie, daß sich der Defaultname ändert – die Endung DSC wird durch die Endung CAS ersetzt.

Thomas Naumann □

```
1 '*****************************  <24B1>
2 '*        ARCHIVATOR         *  <244D>
3 '*            VON            *  <2449>
4 '*      THOMAS NAUMANN       *  <242F>
5 '*           FUER            *  <2407>
6 '*     SCHNEIDER CPC-WELT     *  <24EF>
7 '*      CPC 464/664/6128   JE*  <2453>
8 '*****************************  <24BF>
10 MODE 1:LOCATE 8,10:PRINT"A R C
H I V A T O R":LOCATE 5,15:PRINT"(
c) 1988 by Thomas Naumann"        <488D>
20 IF PEEK(&9200)<>42 THEN OPENOUT
"d":MEMORY HIMEM-1:CLOSEOUT:MEMORY
 &91FF:LOAD"arch.bin":CALL &9200:/
FAST:POKE &BE78,1:POKE &BE47,10   <499E>
30 DEFINT a-z:md=220:mf=1000:DIM d
r$(63),mk(63),lae(63),d$(md),f$(mf
)                                 <55DC>
40 e$=CHR$(200):el$=e$:p$=CHR$(31)
+CHR$(1)+CHR$(2)+CHR$(20):z$=CHR$(
18):b12$=SPACE$(12):b13$=SPACE$(13
):b16$=SPACE$(16):cur$=CHR$(240)+C
HR$(241)+CHR$(242)+CHR$(243)      <93FC>
50 az=0:s=0:free=0:f=0:fz=0:dz=0  <3473>
60 DEF FNx(xxx)=(xxx MOD 4)*20+3:D
EF FNy(yyy)=yyy\4+3               <3DF7>
70 FOR i=0 TO 63:dr$(i)=b12$:NEXT:
d$(0)=b16$:f$(0)=b13$             <4211>
80 dn$=b16$:g$=b13$:t$="Keine Tite
lzeile":mk$=CHR$(246)             <42A9>
90 h$="lsmcgpufatkdznq":drm$="gsut
c"+CHR$(13):df$="GESAMT   .DSC"   <46D8>
100 fi$="Fehler - Eingabe ignorier
t."+CHR$(7):utb$=" Blaettern ":utl
$="<L> Loeschen ":uta$="<A> Aender
n ":utesc$="<ENTER> Abbruch":tst$=
" (<TASTE>)"                      <9FCA>
110 s$=CHR$(27)+CHR$(120)+CHR$(0)+
CHR$(15)+CHR$(27)+CHR$(67)+CHR$(0)
+CHR$(12):drz=64:zbr=120          <53EA>
120 MODE 2                        <079C>
290 'Hauptmenue                   <1163>
300 GOSUB 4200:PRINT p$;:LOCATE 1,
5                                 <1534>
310 PRINT TAB(20);"H A U P T M E N
 U E":LOCATE 1,10:ZONE 40         <2BA1>
320 PRINT"<L> Archiv laden","<S> A
rchiv speichern","<M> Manuell erfa
ssen","<C> Catalog","<G> Gesamtlis
te","<P> Seitenliste","<U> Uebersi
cht","<A> Druckerausgabe"         <9E42>
330 PRINT:PRINT"<F> Finden","<T> K
opfzeile","<N> Neuer Archivname","
<K> Kass/Disk-Archiv (z. Z. ";:IF
cass THEN PRINT"Kass.)"ELSE PRINT"
Disk)"                            <78FF>
350 PRINT"<D> Laufwerkwechsel":PRI
NT:PRINT"<Z> Zurueecksetzen","<Q> A
bbruch"                           <40AF>
360 GOSUB 5000:i=INSTR(h$,el$):IF
i=0 THEN 360 ELSE IF i=LEN(h$)THEN
 END                              <3BE7>
370 ON i GOSUB 1800,1900,700,3300,
1200,1500,900,2400,2500,3200,3800,
4600,4500,2200:GOTO 300           <4727>
690 'Manuell erfassen             <17E9>
700 IF dz>220 OR fz>935 THEN 780 E
LSE GOSUB 4000:IF q THEN 780      <2CE9>
710 IF cass THEN MID$(dn$,16,1)="
"ELSE LOCATE 1,5:PRINT CHR$(20);:L
INE INPUT"Freier Platz: ",free$:IF
 free$=""THEN 780 ELSE free=VAL(fr
ee$):MID$(dn$,16,1)=CHR$(free+32) <8B1C>
720 GOSUB 4100:/SLOW:WINDOW#0,1,80
,VPOS(#0)+2,25                    <25E4>
730 LINE INPUT"Dateiname (<ENTER>:
 fertig): ",n$:IF n$=""THEN 770   <3843>
740 IF NOT cass THEN GOSUB 4300:IF
 ic THEN PRINT fi$:GOTO 730       <268F>
750 IF cass THEN IF LEN(n$)>12 THE
N PRINT fi$:GOTO 730 ELSE g$=n$+MI
D$(b13$,1,13-LEN(n$))             <49A1>
760 MID$(g$,13,1)=CHR$(ds+32):/SI,
@f$(0),fz,@g$,@s:fz=fz+1:GOSUB 420
0:GOTO 730                        <56A6>
770 WINDOW#0,1,80,1,25:/FAST:GOTO
700                               <1EB7>
780 RETURN                        <0642>
890 'Uebersicht                   <1113>
900 IF dz=0 THEN PRINT CHR$(7);:GO
TO 1050 ELSE erst=0:letzt=MIN(erst
+31,dz-1):ZONE 40                 <4986>
910 PRINT p$;:LOCATE 1,3:FOR i=ers
t TO letzt:PRINT USING"###:";i;:PR
INT MID$(d$(i),1,15);:IF NOT cass
THEN PRINT USING" ### K";ASC(MID$(
d$(i),16,1))-32;                  <7B46>
920 PRINT,:NEXT                   <09B3>
930 LOCATE 1,25:PRINT z$;cur$;utb$
;uta$;utl$;utesc$;z$;             <3B46>
940 GOSUB 5000:ON INSTR(cur$,el$)G
OTO 950,960,970,980:GOTO 990      <304A>
950 erst=0:letzt=MIN(erst+31,dz-1)
:GOTO 910                         <3351>
960 letzt=dz-1:erst=MAX(letzt-31,0
):GOTO 910                        <3452>
970 erst=MAX(erst-31,0):letzt=MIN(
erst+31,dz-1):GOTO 910            <42C6>
980 letzt=MIN(dz-1,letzt+31):erst=
MAX(letzt-31,0):GOTO 910          <44D4>
990 IF el$<>"a"THEN 1040 ELSE LOCA
TE 1,25:PRINT z$;:LINE INPUT"Aende
rn - Nr.: ",nr$:IF nr$=""THEN 1050
 ELSE nr=VAL(nr$):IF nr>dz-1 OR nr
<0 THEN GOTO 990 ELSE LOCATE 1,25:
PRINT z$;:LINE INPUT"Name: ",n$   <9483>
1000 IF n$=""THEN 1050 ELSE GOSUB
4400                              <17CE>
1010 MID$(dn$,16,1)=MID$(d$(nr),16
,1):/RP,nr+32,dz+32,@f$(0),fz:/MV,
```

Ordnung leicht gemacht

```
@d$(nr+1),@d$(nr),(dz-nr-1)*3:FOR
i=nr+1 TO dz-1:/RP,i+32,i+31,@f$(0
),fz:NEXT                                <BBC3>
1020 /SI,@d$(0),dz-1,@dn$,@s:FOR i
=dz-2 TO s STEP-1:/RP,i+32,i+33,@f
$(0),fz:NEXT:/RP,dz+32,s+32,@f$(0)
,fz                                      <885E>
1030 GOTO 910                            <097E>
1040 IF el$<>"l"THEN 1050 ELSE LOC
ATE 1,25:PRINT z$;:LINE INPUT"Loes
chen - Nr.: ",nr$:IF nr$=""THEN 10
50 ELSE nr=VAL(nr$):IF nr>dz-1 OR
nr<0 THEN GOTO 990 ELSE s=nr:GOSUB
 3900:GOTO 900                           <8D97>
1050 RETURN                              <065F>
1190 'Gesamtliste                        <1273>
1200 IF fz=0 THEN PRINT CHR$(7);:G
OTO 1360 ELSE ZONE 40                    <1F49>
1210 erst=0:letzt=MIN(fz-1,39)           <2628>
1220 PRINT p$;:LOCATE 1,3:FOR i=er
st TO letzt:PRINT USING"###:";i;:P
RINT MID$(f$(i),1,12);" -> ";MID$(
d$(ASC(MID$(f$(i),13,1))-32),1,15)
,:NEXT:LOCATE 1,25:PRINT cur$;utb$
;utl$;uta$;utesc$;z$;                    <AE06>
1230 GOSUB 5000:ON INSTR(cur$,el$)
GOTO 1210,1240,1250,1260:GOTO 1270
                                         <3036>
1240 letzt=fz-1:erst=MAX(0,letzt-3
9):GOTO 1220                             <34A0>
1250 erst=MAX(0,erst-39):letzt=MIN
(fz-1,erst+39):GOTO 1220                 <42DD>
1260 letzt=MIN(fz-1,letzt+39):erst
=MAX(0,letzt-39):GOTO 1220               <44CD>
1270 IF el$=CHR$(246)THEN erst=MAX
(0,erst-150):letzt=MIN(fz-1,erst+3
9):GOTO 1220                             <50A8>
1280 IF el$=CHR$(247)THEN letzt=MI
N(fz-1,letzt+150):erst=MAX(0,letzt
-39):GOTO 1220                           <5202>
1290 IF el$<>"a"THEN 1340                <1379>
1300 LOCATE 1,25:PRINT z$;:LINE IN
PUT"Aendern - Nr.: ",nr$:IF nr$=""
THEN 1220 ELSE nr=VAL(nr$):IF nr>f
z-1 OR nr<0 THEN 1300 ELSE alt=ASC
(MID$(f$(nr),13,1))                      <8543>
1310 LOCATE 1,25:PRINT z$;:LINE IN
PUT"Name: ",n$:IF n$=""THEN 1220 E
LSE IF NOT cass THEN GOSUB 4300:IF
 ic THEN LOCATE 1,25:PRINT z$;fi$;
tst$:GOSUB 5000:GOTO 1310                <68D7>
1320 IF cass THEN IF LEN(n$)>12 TH
EN LOCATE 1,25:PRINT z$;fi$;tst$:G
OSUB 5000:GOTO 1310 ELSE g$=n$+MID
$(b13$,1,13-LEN(n$))                     <6031>
1330 MID$(g$,13,1)=CHR$(alt):/MV,@
f$(nr+1),@f$(nr),(fz-nr-1)*3:fz=fz
-1:/SI,@f$(0),fz,@g$,@s:fz=fz+1:GO
TO 1220                                  <90AC>
1340 IF el$<>"l"THEN 1360                <13FB>
1350 LOCATE 1,25:PRINT z$;:LINE IN
PUT"Loeschen - Nr.: ",nr$:IF nr$="
"THEN 1220 ELSE nr=VAL(nr$):IF nr>
fz-1 OR nr<0 THEN 1350 ELSE/MV,@f$
(nr+1),@f$(nr),(fz-nr-1)*3:fz=fz-1
:GOSUB 4200:letzt=MIN(letzt,fz-1):
GOTO 1220                                <D0B3>
1360 RETURN                              <06CC>
1490 'Seitenweise geordnet               <1BC9>
1500 IF dz=0 THEN PRINT CHR$(7);:G
OTO 1630 ELSE c=0                        <2248>
1510 PRINT p$;:LOCATE 1,3:cc=c           <1B73>
1520 PRINT"Nr.";cc;"* ";MID$(d$(cc
),1,15);:IF NOT cass THEN PRINT" -
";ASC(MID$(d$(cc),16,1))-32;"K fre
i"ELSE PRINT                             <62F6>
1530 /DP,cc+32,@f$(0),fz:PRINT TAB
(1);:PRINT:IF VPOS(#0)<16 THEN IF
cc<dz-1 THEN cc=cc+1:GOTO 1520           <57A2>
1540 LOCATE 1,25:PRINT cur$;utb$;"
<N> Nr. waehlen ";utesc$                 <35B8>
1550 GOSUB 5000:ON INSTR(cur$,el$)
GOTO 1560,1570,1580,1590:GOTO 1600
                                         <304B>
1560 c=0:GOTO 1510                       <101A>
1570 c=dz-1:GOTO 1510                    <1642>
1580 c=MAX(0,c-1):GOTO 1510              <1B61>
1590 c=MIN(dz-1,cc+1):GOTO 1510          <2205>
1600 IF el$=CHR$(246)THEN c=MAX(0,
c-10):GOTO 1510                          <2A8B>
1610 IF el$=CHR$(247)THEN c=MIN(dz
-1,cc+5):GOTO 1510                       <3017>
1620 IF el$="n"THEN LOCATE 1,25:PR
INT z$;:LOCATE 1,25:LINE INPUT"Dir
ekt anwaehlen - Nr.: ",nr$:IF nr$<
>""THEN nr=VAL(nr$):c=MAX(0,MIN(nr
,dz-1)):GOTO 1510                        <81F2>
1630 RETURN                              <06E9>
1790 'Laden                              <0CA5>
1800 LOCATE 20,25:PRINT"BITTE WART
EN":df$=df$:OPENIN df$:LINE INPUT#
9,t$:i=0:r=0:WHILE NOT r:LINE INPU
T#9,d$(i):r=(d$(i)="\\\"):i=i+1:WE
ND                                       <8362>
1810 dz=i-1:i=0:WHILE NOT EOF:LINE
 INPUT#9,f$(i):i=i+1:WEND:fz=i:CLO
SEIN:GOSUB 4200:LOCATE 9,1:PRINT d
f$;:LOCATE 20,25:PRINT z$;:RETURN        <6A18>
1890 'Speichern                          <108F>
1900 LOCATE 20,25:PRINT"BITTE WART
EN":df$=df$:OPENOUT df$:PRINT#9,t$
:FOR i=0 TO dz-1:PRINT#9,d$(i):NEX
T:PRINT#9,"\\\":FOR i=0 TO fz-1:PR
INT#9,f$(i):NEXT:CLOSEOUT:LOCATE 2
0,25:PRINT z$;:RETURN                    <95E3>
2190 'Neuer Archivdateiname              <1C7B>
2200 LOCATE 1,25:PRINT z$;:LINE IN
PUT"Neuer Archivname: ",n$:IF n$="
"THEN 2230 ELSE GOSUB 4300:IF ic T
HEN LOCATE 40,25:PRINT fi$;tst$:GO
```

```
SUB 5000:GOTO 2200                          (6668)
2210 IF cass THEN df$=MID$(g$,1,8)
+".CAS"ELSE df$=MID$(g$,1,8)+".DSC
"                                           (409D)
2220 GOSUB 4200                             (09B0)
2230 RETURN                                 (069A)
2390 'Finden                                (0D9A)
2400 IF fz=0 THEN PRINT CHR$(7);:G
OTO 2440 ELSE n=0                           (22A5)
2410 PRINT p$;:LOCATE 1,3:LINE INP
UT"Gesucht: ",sn$:IF sn$=""THEN 24
40 ELSE i=0:MID$(sn$,1,LEN(sn$))=U
PPER$(sn$)                                  (592D)
2420 IF INSTR(UPPER$(f$(i)),sn$)>0
 THEN n=i:erst=MAX(0,i-10):letzt=M
IN(erst+32,fz-1):ZONE 40:GOSUB 122
0:PRINT p$;:LOCATE 1,3:PRINT"Weite
rsuchen nach ";sn$;" (<J>/<N>)?":G
OSUB 5000:IF el$="j"THEN i=n ELSE
2440                                        (BEBC)
2430 i=i+1:IF i<fz THEN 2420 ELSE
PRINT:PRINT"Nicht gefunden!";tst$:
GOSUB 5000                                  (4276)
2440 RETURN                                 (063F)
2490 'Ausgabe auf Drucker                   (1A00)
2500 IF dz=0 THEN PRINT CHR$(7);:G
OTO 2540 ELSE PRINT p$;:LOCATE 20,
4:PRINT"D r u c k m e n u e":LOCAT
E 20,5:PRINT"-------------------":
LOCATE 1,8                                  (610A)
2510 PRINT TAB(20);"<G>";TAB(30);"
Gesamtliste":PRINT:PRINT TAB(20);"
<S>";TAB(30);"Seitengeordnete List
e":PRINT:PRINT TAB(20);"<U>";TAB(3
0);"Uebersicht":PRINT:PRINT TAB(20
);"<T>";TAB(30);"Titelzeile"        (963D)
2520 PRINT:PRINT TAB(20);"<C>";TAB
(30);"Steuercodes":PRINT:PRINT TAB
(20);"<ENTER>";TAB(30);"Hauptmenue
"                                           (4D0A)
2530 GOSUB 5000:i=INSTR(drm$,el$):
IF i=0 THEN 2530 ELSE IF i<LEN(drm
$)THEN ON i GOSUB 2600,2700,2800,2
900,3000:GOTO 2500                          (5C9B)
2540 RETURN                                 (0607)
2590 'Gesamtliste                           (1268)
2600 PRINT#8,s$;:se=1:ZONE 40:WIDT
H zbr:GOSUB 2630:FOR i=0 TO fz-1:P
RINT#8,"          ";MID$(f$(i),1,12
);" - ";MID$(d$(ASC(MID$(f$(i),13,
1))-32),1,15),                              (89E2)
2610 IF(i+1)MOD((drz-2)*(zbr\40))=
0 THEN PRINT#8,CHR$(12);:se=se+1:G
OSUB 2630                                   (48EE)
2620 NEXT:PRINT#8,CHR$(12);:RETURN
                                            (1424)
2630 PRINT#8,TAB(9);t$;" - Seite";
se:PRINT#8:RETURN                           (29A1)
2690 'Seitenorientierte Liste               (1EB8)
2700 PRINT#8,s$;:se=1:ZONE 20:WIDT
H zbr:GOSUB 2630:PRINT#8:n=0:FOR i
=0 TO dz-1:PRINT#8,TAB(8);MID$(d$(
i),1,15);:IF cass THEN PRINT#8 ELS
E PRINT#8," (";ASC(MID$(d$(i),16,1
))-32;"K frei)"                             (9A90)
2710 j=0:nn=0                               (1377)
2720 /FL,i+32,@f,@f$(j),fz-j:IF f<
>-1 THEN f=f+j:PRINT#8,"        ";M
ID$(f$(f),1,12),:j=f+1:IF nn MOD(z
br\20)=0 THEN n=n+1                         (9587)
2730 IF j<fz AND f<>-1 THEN nn=nn+
1:GOTO 2720                                 (2BCD)
2740 PRINT#8,TAB(1):PRINT#8:n=n+2:
IF n>drz-2-(64\(zbr\20))THEN PRINT
#8,CHR$(12);:se=se+1:GOSUB 2630:n=
0                                           (633D)
2750 NEXT:PRINT#8,CHR$(12);:RETURN
                                            (1429)
2790 'Uebersicht                            (11F2)
2800 PRINT#8,s$;:se=1:GOSUB 2630:Z
ONE 40:WIDTH zbr:FOR i=0 TO dz-1:P
RINT#8,"          ";MID$(d$(i),1,15)
;:IF cass THEN PRINT#8,ELSE PRINT#
8," (";ASC(MID$(d$(i),16,1))-32;"K
 frei)",                                    (9741)
2810 IF(i+1)MOD((drz-3)*(zbr\40))=
0 THEN PRINT#8,CHR$(12);:se=se+1:G
OSUB 2630                                   (48E8)
2820 NEXT:PRINT#8,CHR$(12);:RETURN
                                            (14B6)
2890 'Titelzeile                            (1115)
2900 LOCATE 1,25:LINE INPUT"Titelz
eile: ",n$:IF n$<>""THEN t$=n$              (33ED)
2910 RETURN                                 (06EE)
2990 'Steuercodes                           (129F)
3000 LOCATE 1,24:PRINT"Steuerzeich
en (Syntax: ";CHR$(34);"xx<,xx>;Ze
ilenbreite;Zeilen pro Seite";CHR$(
34)+")"                                     (5FE7)
3010 LINE INPUT"",n$:IF n$=""THEN
3050 ELSE i=INSTR(n$,";"):j=LEN(n$
):IF i<>0 THEN IF j>i+1 THEN zbr=V
AL(MID$(n$,i+1)):i=INSTR(i+1,n$,";
"):IF i<>0 THEN IF j>i+1 THEN drz=
VAL(MID$(n$,i+1))                           (B3C2)
3020 i=INSTR(n$,";"):IF i>1 THEN n
$=LEFT$(n$,i-1)ELSE IF i=1 THEN 30
50                                          (40E5)
3030 s$="":i=1                              (1375)
3040 s$=s$+CHR$(VAL(MID$(n$,i))):i
=INSTR(i,n$,",")+1:IF LEN(n$)-i>0
AND i<>1 THEN 3040                          (589F)
3050 RETURN                                 (0605)
3190 'Titel-/Datumszeile                    (194B)
3200 LOCATE 1,25:LINE INPUT"Neue T
itelzeile: ",n$:IF n$<>""THEN t$=n
$                                           (38E5)
3210 RETURN                                 (0646)
3290 'Directory zeigen und bearbei
ten                                         (26A8)
```

Ordnung leicht gemacht

```
3300 PRINT p$:/CT,@dr$(0),@lae(0),
@az,@free:IF az<>-1 THEN 3330 ELSE
 PRINT"LESEFEHLER - (<N> Nochmal
<D> Disknamen aufnehmen  <Q> Abbre
chen)":GOSUB 5000:IF el$="n"THEN G
OTO 3300                          <9E84>
3310 IF el$="q"THEN 3610 ELSE GOSU
B 4000:IF NOT q THEN MID$(dn$,16,1
)=" ":GOSUB 4100:GOSUB 4200       <3C19>
3320 PRINT"Diskwechsel -";tst$:GOS
UB 5000:GOTO 3300                 <26EE>
3330 FOR i=0 TO az-1:LOCATE FNx(i)
,FNy(i):PRINT dr$(i);:PRINT USING"
 ###K";lae(i),:NEXT:PRINT TAB(1):P
RINT:PRINT USING"### K frei";free;
:PRINT TAB(40);:PRINT USING"## Dat
eien";az                          <92DF>
3340 w=0:FOR i=0 TO az-1:mk(i)=0:N
EXT                               <2B4B>
3350 LOCATE 1,23:PRINT"<CURSOR> Wa
ehlen <COPY> Markieren <DEL> M. lo
eschen <I> M. invertieren <K> Kei-
 ne M. <G> Alles m. <B> BAK-Dateie
n loeschen <L> Loeschen <U> Umbene
nnen"                             <A596>
3360 PRINT"<A> Archivieren <ENTER>
 Abbruch"                         <272A>
3370 IF az<>0 THEN x=FNx(w):y=FNy(
w):LOCATE x,y:/SLOW:PRINT CHR$(24)
;dr$(w);CHR$(24);:/FAST:ow=w       <6F8B>
3380 GOSUB 5000:IF az=0 THEN 3560
ELSE IF el$<CHR$(127)THEN 3460    <29D7>
3390 IF el$=CHR$(&E0)THEN LOCATE x
-2,y:PRINT mk$:mk(w)=-1:w=MIN(w+1,
az-1)                             <4E40>
3400 IF el$=CHR$(127)THEN LOCATE x
-2,y:PRINT" ":mk(w)=0:w=MIN(w+1,az
-1)                               <4AB2>
3410 IF el$=CHR$(240)THEN w=MAX(0,
w-4)                              <24EA>
3420 IF el$=CHR$(241)THEN w=MIN(az
-1,w+4)                           <2A34>
3430 IF el$=CHR$(242)THEN w=MAX(0,
w-1)                              <24BA>
3440 IF el$=CHR$(243)THEN w=MIN(az
-1,w+1)                           <2A33>
3450 LOCATE x,y:PRINT dr$(ow):GOTO
 3370                             <220F>
3460 IF el$<>"i"THEN 3480 ELSE FOR
 i=0 TO az-1:mk(i)=NOT mk(i):LOCAT
E FNx(i)-2,FNy(i):IF mk(i)THEN PRI
NT mk$ELSE PRINT" "               <724E>
3470 NEXT                         <06EC>
3480 IF el$="k"THEN FOR i=0 TO az-
1:mk(i)=0:LOCATE FNx(i)-2,FNy(i):P
RINT" ":NEXT                      <4FF8>
3490 IF el$="g"THEN FOR i=0 TO az-
1:mk(i)=-1:LOCATE FNx(i)-2,FNy(i):
PRINT mk$:NEXT                    <52EE>
3500 IF el$="b"THEN n$="*.bak":PRI
NT p$:/ERA,@n$:GOTO 3300          <3367>
3510 IF el$<>"l"THEN 3530 ELSE PRI
NT p$:FOR i=0 TO az-1:IF mk(i)THEN
/ERA,@dr$(i)                      <4A91>
3520 NEXT:GOTO 3300               <0B4A>
3530 IF el$<>"u"THEN 3560 ELSE WIN
DOW 1,80,3,25:CLS:FOR i=0 TO az-1:
IF mk(i)THEN PRINT"Alter Name: ";d
r$(i);" --- ";:LINE INPUT"Neuer Na
me: ",n$:IF n$<>""THEN/REN,@n$,@dr
$(i)                              <9B1C>
3540 NEXT:WINDOW 1,80,1,25:GOTO 33
00                                <1615>
3560 IF el$<>"a"THEN 3590 ELSE IF
dz>222 OR fz>935 OR cass THEN PRIN
T CHR$(7);:GOTO 3600 ELSE GOSUB 40
00:IF q THEN 3300 ELSE MID$(dn$,16
,1)=CHR$(free+32):GOSUB 4100      <704B>
3570 FOR i=0 TO az-1:IF mk(i)THEN
g$=dr$(i)+CHR$(ds+32):/SI,@f$(0),f
z,@g$,@s:fz=fz+1                  <6D4C>
3580 NEXT:GOSUB 4200:PRINT"Diskwec
hsel (falls gewuenscht) - ";tst$:G
OSUB 5000:GOTO 3300               <41F6>
3590 IF el$=CHR$(13)THEN 3610     <16EA>
3600 GOTO 3380                    <09F9>
3610 RETURN                       <0669>
3790 'Umschalten Kass/Disk        <1BDE>
3800 IF NOT cass THEN MID$(df$,10,
3)="CAS":GOTO 3820                <2720>
3810 MID$(df$,10,3)="DSC"         <1848>
3820 cass=NOT cass:RETURN         <17E2>
3890 'Disk stelle loeschen        <1BCF>
3900 /MV,@d$(s+1),@d$(s),(dz-1-s)*
3:dz=dz-1:i=0                     <49A5>
3910 /FL,s+32,@f,@f$(i),fz-i:IF f<
>-1 THEN/MV,@f$(f+i+1),@f$(f+i),(f
z-f-i-1)*3:fz=fz-1:IF f+i<=fz THEN
 i=f+i:GOTO 3910                  <A9E3>
3920 FOR i=s+1 TO dz:/RP,i+32,i+31
,@f$(0),fz:NEXT:GOSUB 4200:RETURN <4409>
3990 'Name der Diskseite          <19FE>
4000 q=0:PRINT p$;:LOCATE 1,3:LINE
 INPUT"Name der Seite: ",n$:IF n$=
""THEN q=-1:GOTO 4030 ELSE GOSUB 4
400:/FF,@dn$,@s,@d$(0),dz         <6ED1>
4010 IF s=-1 THEN 4030 ELSE LOCATE
 1,5:PRINT"Diese Seite ist bereits
 im Archiv. Alter Inhalt:":PRINT:/
DP,s+32,@f$(0),fz                 <6948>
4020 PRINT TAB(1):PRINT:PRINT"<U>
Ueberschreiben <ENTER> Ignorieren"
:GOSUB 5000:IF el$<>"u"THEN 4000 E
LSE GOSUB 3900                    <4F32>
4030 RETURN                       <06B1>
4090 'Diskname einsortieren       <1C10>
4100 /SI,@d$(0),dz,@dn$,@s:ds=s:FO
R i=dz-1 TO ds STEP-1:/RP,i+32,i+3
3,@f$(0),fz:NEXT:dz=dz+1:RETURN   <7D5A>
4190 'Statuszeile                 <1261>
```

```
4200 LOCATE#1,1,1:PRINT#1,"Archiv:
 ";df$;" * ";:PRINT#1,USING"### ";
dz;:IF cass THEN PRINT#1,"Kass. ";
ELSE PRINT#1,"Disks ";                (61AE)
4210 PRINT#1,USING"### Dateien ";f
z;:PRINT#1,"* Lfw. ";CHR$(65+PEEK(
&A700));" * ";MID$(t$,1,MIN(LEN(t$
),23));z$;:RETURN                     (64BD)
4290 'fbez$ aus name$ bilden          (1D15)
4300 ic=0:i=INSTR(n$,"."):j=LEN(n$
):IF(i=0 AND j>8)OR i>9 OR i=1 OR
j>12 OR(i>0 AND j-i>3)THEN ic=-1:G
OTO 4330 ELSE g$=b13$                 (830C)
4310 IF i=0 THEN MID$(g$,1,j)=n$EL
SE MID$(g$,1,i-1)=MID$(n$,1,i-1):M
ID$(g$,9,1)=".":MID$(g$,10,j-i)=MI
D$(n$,i+1,j-i)                        (811B)
4320 MID$(g$,1)=UPPER$(g$)            (17E1)
4330 RETURN                           (060A)
4390 'Aus name$ dname$ machen         (1ED8)
4400 dn$=b16$:MID$(dn$,1,15)=MID$(
n$,1,MIN(15,LEN(n$))):RETURN          (3BC0)
4490 'Programm zuruecksetzen          (1D9F)
4500 LOCATE 1,25:PRINT"SICHER? (J)
/(N)":GOSUB 5000:IF el$="j"THEN RU
N                                     (2FC4)
4510 RETURN                           (0673)
4590 'Laufwerkwechsel                 (161C)
4600 i=NOT-PEEK(&A700):POKE &A700,
ABS(i):RETURN                         (2338)
4990 'Warten auf Taste                (1787)
5000 MID$(e$,1,1)=CHR$(200):WHILE
e$=CHR$(200):MID$(e$,1,1)=INKEY$:W
END:MID$(el$,1,1)=LOWER$(e$):RETUR
N                                     (4D2A)
```

```
100 '****************************    (2234)
110 '*  ARCHIVATOR-DATALADER    *    (2257)
120 '*   ERZEUGT (ARCH.BIN)     *    (2257)
130 '*          VON             *    (22F5)
140 '*     THOMAS HAUMANN       *    (229E)
150 '*          FUER            *    (2288)
160 '*   SCHNEIDER CPC-WELT     *    (2247)
170 '*    CPC 464/664/6128   JE*    (2256)
180 '****************************    (22D4)
190 MEMORY &91FF                      (09CC)
650 a=&9200:e=&964A:zb=1000:e=e+1     (2C34)
660 FOR i=a TO e:IF i=e THEN SAVE"
ARCH.BIN",B,&9200,&44A:END            (3A22)
670 READ d$:POKE i,VAL("&"+d$)        (1D5A)
730 IF i<e THEN NEXT i                (1526)
1001 DATA 2A,A6,BB,CB,BC,22,6C,92     (1E85)
1002 DATA 2A,1B,BC,CB,BC,22,71,92     (1EDC)
1003 DATA 2A,12,BC,CB,BC,22,62,92     (1EBF)
1004 DATA 2A,D4,BD,22,68,92,01,28     (1E0D)
1005 DATA 92,21,45,92,CD,D1,BC,C9     (1E5E)
1006 DATA 49,92,C3,3A,93,C3,80,92     (1E7E)
1007 DATA C3,87,92,C3,FA,93,C3,6D     (1EC7)
1008 DATA 94,C3,2B,94,C3,9D,94,C3     (1E1F)
1009 DATA D4,94,C3,95,95,00,00,00     (1EE1)
1010 DATA 00,53,C9,46,41,53,D4,53     (1EE7)
1011 DATA 4C,4F,D7,4D,D6,46,C6,46     (1E0D)
1012 DATA CC,52,D0,44,D0,43,D4,00     (1E46)
1013 DATA 4F,CD,00,00,FE,02,79,C2     (1ECD)
1014 DATA 00,00,E5,CD,00,00,EB,E1     (1E8D)
1015 DATA CD,00,00,EB,06,08,7E,12     (1E9E)
1016 DATA 7A,C6,08,57,23,10,F7,C9     (1EE8)
1017 DATA 21,60,92,22,D4,BD,C9,2A     (1E62)
1018 DATA 68,92,22,D4,BD,C9,21,94     (1E24)
1019 DATA 92,C3,CF,92,50,61,72,61     (1EA6)
1020 DATA 6D,65,74,65,72,66,65,68     (1E9A)
1021 DATA 6C,65,72,0D,0A,07,00,21     (1E4D)
1022 DATA AD,92,C3,CF,92,57,65,72     (1EB8)
1023 DATA 74,66,65,68,6C,65,72,0D     (1EC8)
1024 DATA 0A,07,00,21,C1,92,C3,CF     (1EEC)
1025 DATA 92,44,69,73,6B,66,65,68     (1E26)
1026 DATA 6C,65,72,0D,0A,07,00,7E     (1ED0)
1027 DATA B7,C8,CD,5A,BB,23,C3,CF     (1E73)
1028 DATA 92,00,00,00,00,00,00,00     (1E9F)
1029 DATA 00,F5,C5,D5,E5,DD,E5,FD     (1EE4)
1030 DATA E5,EB,7E,23,4E,23,46,EB     (1E67)
1031 DATA 5F,1C,C5,FD,E1,56,14,23     (1E74)
1032 DATA 4E,23,46,C5,DD,E1,DD,7E     (1EC4)
1033 DATA 00,DD,23,15,CA,1D,93,CD     (1EA9)
1034 DATA 31,93,47,FD,7E,00,FD,23     (1E2C)
1035 DATA 1D,CA,27,93,CD,31,93,B8     (1E29)
1036 DATA CA,FE,92,38,0A,FD,E1,DD     (1EF7)
1037 DATA E1,E1,D1,C1,F1,B7,C9,FD     (1E22)
1038 DATA E1,DD,E1,E1,D1,C1,F1,37     (1E95)
1039 DATA C9,FE,61,D8,FE,7B,D0,CB     (1EF8)
1040 DATA AF,C9,FE,04,C2,8E,92,DD     (1E5D)
1041 DATA 6E,00,DD,66,01,22,DF,92     (1ED8)
1042 DATA DD,6E,02,DD,66,03,22,D9     (1E52)
1043 DATA 92,DD,6E,04,DD,66,05,22     (1E81)
1044 DATA DD,92,E5,DD,6E,06,DD,66     (1EB5)
1045 DATA 07,22,DB,92,E1,01,00,00     (1EAB)
1046 DATA B7,ED,42,CA,E6,93,2B,EB     (1E01)
1047 DATA 21,00,00,E5,D5,EB,B7,ED     (1EA3)
1048 DATA 52,CB,7C,D1,E1,C2,A9,93     (1EF8)
1049 DATA E5,D5,19,B7,CB,1C,CB,1D     (1E29)
1050 DATA 4D,44,09,09,EB,2A,DB,92     (1EB9)
1051 DATA 19,EB,2A,D9,92,CD,E1,92     (1E0F)
1052 DATA D1,E1,D2,A3,93,03,69,60     (1EF0)
1053 DATA C3,73,93,0B,59,50,C3,73     (1E90)
1054 DATA 93,E5,EB,2A,DF,92,73,23     (1E21)
1055 DATA 72,2A,DD,92,B7,ED,52,CA     (1E15)
1056 DATA D2,93,5D,54,19,19,4D,44     (1EF2)
1057 DATA 2A,DD,92,EB,2A,DB,92,19     (1EEA)
1058 DATA 19,19,2B,5D,54,13,13,13     (1E2D)
1059 DATA ED,B8,D1,6B,62,19,19,EB     (1E99)
1060 DATA 2A,DB,92,19,EB,2A,D9,92     (1E88)
1061 DATA 01,03,00,ED,B0,C9,2A,DB     (1E5B)
1062 DATA 92,EB,2A,D9,92,01,03,00     (1E26)
1063 DATA ED,B0,2A,DF,92,AF,77,23     (1E8C)
1064 DATA 77,C9,FE,03,C2,8E,92,DD     (1E7E)
1065 DATA 4E,00,DD,46,01,21,00,00     (1E75)
1066 DATA B7,ED,42,C8,DD,5E,02,DD     (1EF8)
```

Ordnung leicht gemacht

```
1067 DATA 56,03,DD,6E,04,DD,66,05 <1E01>
1068 DATA E5,B7,ED,52,E1,38,03,ED <1E87>
1069 DATA B0,C9,09,2B,EB,09,2B,EB <1EEC>
1070 DATA ED,B8,C9,FE,04,C2,8E,92 <1E16>
1071 DATA DD,4E,00,DD,46,01,DD,6E <1E56>
1072 DATA 02,DD,66,03,E5,DD,6E,04 <1E2E>
1073 DATA DD,66,05,22,DF,92,DD,7E <1E11>
1074 DATA 06,21,6B,94,77,EB,26,0C <1EB3>
1075 DATA 2E,01,E5,FD,E1,E1,CD,45 <1EBC>
1076 DATA 95,38,09,2A,DF,92,36,FF <1E4D>
1077 DATA 23,36,FF,C9,2A,DF,92,71 <1EE9>
1078 DATA 23,70,C9,00,00,FE,04,C2 <1E9D>
1079 DATA 8E,92,DD,4E,00,DD,46,01 <1ECB>
1080 DATA DD,6E,02,DD,66,03,E5,DD <1EEF>
1081 DATA 6E,04,DD,66,05,22,DF,92 <1E72>
1082 DATA DD,6E,06,DD,66,07,23,5E <1EBE>
1083 DATA 23,56,26,00,2E,0F,E5,FD <1E91>
1084 DATA E1,E1,C3,56,94,FE,04,C2 <1E7F>
1085 DATA 8E,92,DD,4E,00,DD,46,01 <1ED7>
1086 DATA 21,00,00,B7,ED,42,C8,DD <1E20>
1087 DATA 6E,02,DD,66,03,E5,DD,7E <1E31>
1088 DATA 04,32,6C,94,DD,7E,06,21 <1ECB>
1089 DATA 6B,94,77,EB,21,CF,94,22 <1E36>
1090 DATA 2D,95,E1,CD,19,95,C9,3A <1EED>
1091 DATA 6C,94,77,C9,FE,03,C2,8E <1EDF>
1092 DATA 92,DD,4E,00,DD,46,01,21 <1EBA>
1093 DATA 00,00,B7,ED,42,C8,DD,6E <1E8A>
1094 DATA 02,DD,66,03,E5,DD,7E,04 <1E98>
1095 DATA 21,6B,94,77,EB,21,00,95 <1E4F>
1096 DATA 22,2D,95,E1,CD,19,95,C9 <1E37>
1097 DATA 01,0C,00,B7,ED,42,06,0C <1E9D>
1098 DATA 7E,CD,5A,BB,23,10,F9,06 <1E32>
1099 DATA 08,3E,09,CD,5A,BB,10,FB <1ED9>
1100 DATA C9,D5,E5,26,0C,2E,01,E5 <1E2F>
1101 DATA FD,E1,E1,E5,C5,CD,45,95 <1EA9>
1102 DATA D2,41,95,C5,CD,00,00,D1 <1E8B>
1103 DATA E1,13,B7,ED,52,4D,44,6B <1E98>
1104 DATA 62,19,19,D1,19,D1,C3,19 <1E19>
1105 DATA 95,E1,E1,E1,C9,F5,E5,C5 <1E26>
1106 DATA E1,22,DD,92,E1,22,DB,92 <1EE0>
1107 DATA 01,00,00,2A,DD,92,B7,ED <1E6A>
1108 DATA 42,CA,92,95,69,60,09,09 <1E7C>
1109 DATA D5,EB,2A,DB,92,19,23,5E <1E3F>
1110 DATA 23,56,EB,FD,E5,D1,5A,16 <1E33>
1111 DATA 00,19,D1,D5,C5,FD,E5,C1 <1ECE>
1112 DATA 41,1A,BE,C2,8C,95,23,13 <1E61>
1113 DATA 10,F7,06,00,B7,ED,42,C1 <1E2D>
1114 DATA D1,F1,37,C9,C1,D1,03,C3 <1E02>
1115 DATA 53,95,F1,B7,C9,FE,04,C0 <1EAD>
1116 DATA DD,6E,00,DD,66,01,22,DF <1E00>
1117 DATA 92,DD,6E,02,DD,66,03,22 <1E55>
1118 DATA DD,92,DD,6E,04,DD,66,05 <1E82>
1119 DATA 22,35,96,DD,6E,06,DD,66 <1EB6>
1120 DATA 07,22,DB,92,3E,C9,32,D9 <1E31>
1121 DATA BD,11,00,98,CD,9B,BC,3E <1EEC>
1122 DATA C3,32,D9,BD,38,5E,28,5C <1E63>
1123 DATA 0E,00,DD,21,01,98,2A,DB <1E4A>
1124 DATA 92,23,5E,23,56,06,0B,DD <1EE4>
1125 DATA 7E,00,B7,28,35,E6,7F,FE <1EAA>
1126 DATA 20,38,41,12,DD,23,13,3E <1E3A>
1127 DATA 04,B8,20,04,3E,2E,12,13 <1E3E>
1128 DATA 10,E5,2A,35,96,DD,7E,00 <1E50>
1129 DATA 77,DD,23,DD,23,DD,23,23 <1E87>
1130 DATA AF,77,23,22,35,96,0C,2A <1E64>
1131 DATA DB,92,23,23,23,22,DB,92 <1E4E>
1132 DATA 18,BC,2A,DD,92,71,23,AF <1E7D>
1133 DATA 77,3A,B4,BF,2A,DF,92,77 <1E6B>
1134 DATA 23,AF,77,C9,2A,DD,92,3E <1E66>
1135 DATA FF,77,23,77,C9,00,00,DD <1E82>
1136 DATA 23,DD,23,DD,23,DD,23,C9 <1EE9>
1137 DATA FD,23,FD,23,FD,23,FD,23 <1ED4>
1138 DATA C9,00,00                <0F70>
```

Alle Programme auf Disc erhältlich!

Bestell-Coupon Seite 81

BEBAUEN · BEWAHREN



Brot für die Welt

Postgiro Köln 500 500-500

2500 PROGRAMMIERSPRACHEN ZUR AUSWAHL

Wie in Babylon

Von Dipl.-Ing. (FH) Oliver Rosenbaum

Der vielschichtige, schnell wachsende Markt der Programmiersprachen ist auch für Insider kaum noch überschaubar. Unsere Serie erläutert die Hauptunterschiede. In der zweiten Folge: Zwanzig weitere Programmiersprachen – von Exapt bis OPS-5.

29. Exapt

Extended Subset of **APT**.
Weiterentwicklung der Werkzeugmaschinen-Steuersprache APT (für numerische Steuerungen).
Exapt ermöglicht neben der zweidimensionalen Steuerung auch die Berücksichtigung des Werkstoffes (zweieinhalbdimensionale Steuerung).
Neben APT (siehe dort) ist Exapt eine der wichtigsten Sprachen zur Steuerung von NC-Maschinen.
Exapt wird ebenfalls eingesetzt für automatisches Zeichnen, ist aber problemorientiert, also maschinenunabhängig zu programmieren.
Ein sinnvoller Einsatz von Exapt setzt eine Großrechenanlage voraus.
Exapt wurde in Deutschland auf der Basis des amerikanischen APT erstellt und ist daher auch in Englisch gehalten.
1965 wurde Exapt an den Technischen Hochschulen in Aachen, Berlin und Stuttgart in Zusammenarbeit mit der Industrie entwickelt.
Man unterscheidet drei Sprachbereiche:

Exapt 1 für Bohr- und Fräsbearbeitung mit Punkt- oder Streckensteuerung,
Exapt 2 für Drehmaschinen mit Strekken- und Bahnsteuerung,
Exapt 3 für Fräsbearbeitung mit zweieinhalbdimensionaler Bahnsteuerung.

Exapt-Programme bestehen aus drei Sprachteilen. Bei der Übersetzung werden daher drei Übersetzungsprogramme benötigt.
Die jeweiligen Übersetzungen werden getrennt voneinander vorgenommen, da jeder Teil für sich sehr umfangreich ist.
Man unterscheidet:

- geometrischer Prozessor
- technologischer Prozessor
- Postprozessor.

Im geometrischen Prozessor werden im ersten Maschinendurchlauf alle geometrischen Definitionen, die das Programm enthält, über das Werkstück und die Peripherie verarbeitet.
Beim zweiten Maschinendurchlauf wird die Technologie der Bearbeitung des Werkstückes bestimmt.
Zur Auswertung kommen Schnittwerte, Werkzeuge, Stoff- und Werkzeugdaten, die meist schon vorab gespeichert wurden.
Der Postprozessor stellt die Verbindung des Programmes mit der NC-Maschine vor; dies ist notwendig, da Exapt problemorientiert ist (siehe oben).
Der Vorteil dieser Dreiteilung liegt in der größeren Beweglichkeit, der Einsparung von Speicherplatz und der besseren Fehlersuche und -korrektur.
Programmbeispiel siehe Tabelle 8.

Programmbeispiel zu Exapt
Listingausschnitt

```
 1 PARTNO/FLANSCH RO 83-17          Die Anweisung kennzeichnet den
                                    Programmanfang. Hinter dem "/"
                                    steht der Name des Werkstückes mit
                                    der Zeichnungsnummer.
 2 MACHIN/PP1,SETPOS,-150,-150,1    Hier steht die Werkzeugmaschine,
                                    auf der das Teil angefertigt werden
                                    soll.
 3 TRANS/167.2,199.3,0.5            Angegeben sind die drei Koordinaten
                                    der Position, in der das Werkstück
                                    eingespannt wurde.
 4 P1=POINT/100,60,25               Definition von Punkt 1
 5 P2=POINT/15,20,15                Definition von Punkt 2
 6 P3=POINT/185,20,15               Definition von Punkt 3
 7 ZSURF/25                         Z-Koordinate des Bohrplans
 8 K1=CIRCLE/CENTER,P1,RADIUS,(80/2) Definition des Teilkreises K1
 9 K2=PATERN/ARC,K1,8,CLW,4         Definition der Gewindelöcher auf K1
10 ZSURF/15                         Definition der Z-Koordinate
                                    von L1 und L2
11 L1=PATERN/LIN,P2,ATANGL,90,INCR,2,AT40  Definition der Bohrung L1
12 L2=PATERN/LIN,P3,ATANGL,90,INCR,2,AT40  Definition der Bohrung L2
13 PART/MATERL.27                   Materialnummer
14 A1=REAM/DIAMET,30,DEPTH,25       Arbeitszyklus "Reiben" zur
                                    Herstellung einer Bohrung,
                                    Durchmesser 30 mm
15 A2=TAP/DIAMET,10,DEPTH,15,TAT,1,BLIND,1  Herstellung eines M10 Gewindes
16 A3=DRILL/DIAMET,10,DEPTH,15      Definition von 10mm Bohrungen
17 A4=SINK/SO,DIAMET,20,DEPTH,5,SPIRET,1  Senkoperation mit Durchmesser 20mm
18 CLDIST/2                         Sicherheitsabstand von 2mm
19 COOLNT/ON                        Kühlmittelzugabe wird gestartet
20 FROM/10,10,0                     Definition der Ausgangsposition der
                                    ersten Bewegung
21 WORK/A1                          Die Reibebearbeitung wird gestartet
22 GOTO/P1                          bei Punkt 1
23 WORK/A2                          Die Gewindeoperation wird gestartet
24 GOTO/K2                          bei Punkt K2 (Teilkreis)
25 WORK/A3,A4                       Die Bohrbearbeitung (10mm) und die
                                    darauffolgende Senkbearbeitung wird
                                    gestartet.
26 GOTO/L1                          bei L1
27 CODLTA/12,NOCUT                  Anheben des Werkzeuges beim
                                    überschweben von Hindernissen
28 GOTO/L2                          Zweite Bohrbearbeitung (wie 25)
29 FINI                             Ende des Programms
```

Tabelle 8

30. FOCUS

Sprache für Expertensysteme, siehe auch MANTIS, oder andere DL-Sprachen wie SQL.

31. Forth

Forth ist ein stackorientiertes Programmsystem, das etwa seit 1980 einem breiteren Anwenderkreis durch die Implementierungen für Micro-Computer zugänglich gemacht wurde.
Entwickelt wurde Forth zehn Jahre früher von Charles Moore auf der legendären IBM 1130.
Forth sollte das System der „vierten Generation" werden, daher der Name (die IBM 1130 ließ maximal fünfstellige Namen zu).
Heute gibt es preiswerte Implementationen für die verschiedensten Rechnertypen und -größen:

- das Standard-Forth der Forth-Interest-Group, auch Fig Forth genannt;
- Forth 79, eine Erweiterung des Fig Forth;
- C-64 Forth für den Commodore, das speziell den C64 und seinen grafischen Eigenschaften angepaßt wurde, –

um nur die bekanntesten zu nennen.
Das Forth-System ist äußerst portabel. Implementationen gibt es für fast alle Prozessortypen.
Die hier vorliegende Be-

schreibung von Forth basiert auf der C64-Version. Forth unterscheidet sich in einigen markanten Punkten von anderen Programmiersprachen:

- Forth arbeitet mit der RPN (Reverse Polish Notation, umgekehrten Polnischen Notation) hier auch Lifo (Last in first out);
- Fort ist stackorientiert und damit maschinennäher als andere höhere Programmiersprachen; daraus ergibt sich eine höhere Ausführungsgeschwindigkeit der Programme sowie eine bessere Speicherausnutzung;
- Bei der Programmierung in Forth wird zugleich mit einem Interpreter und einem Compiler gearbeitet (interpretierender Compiler).
  Forth besitzt zudem ein eigenes Betriebssystem;
- Das Forth-System ist relativ klein und benötigt nur etwa acht bis zwölf KByte;
- Es können eigene Befehle entworfen und in das System integriert werden;
- Das Forth-System ist äußerst transparent und in sich selbst definiert, was auch Änderungen am System selbst erlaubt.

DIE REGISTER IN FORTH

Forth hat grundsätzlich zwei verschiedene Stapelregister: eines für die Verarbeitung von Zahlen und ein weiteres für definierte Wörter, Befehle und anderes.
In Forth gibt es nur wenige fertig vordefinierte Befehle; sie beschränken sich auf ein unbedingt notwendiges Minimum, ähnlich wie bei Assembler-Sprachen.
Neu definierte Befehle (sie basieren auf den vorhandenen) werden in ein „Wörterbuch" abgelegt. Der TOD (Top Of Dictionary) zeigt immer auf den zuletzt definierten Befehl.
Forth-Programme werden in sogenannten Screens (Bildschirmseiten) auf Diskette abgespeichert. Ein Screen entspricht dem Inhalt eines kompletten Bildschirmes.
Mit Hilfe des im Forth-Betriebssystem enthaltenen Editors können einzelne Screens beschrieben, geändert oder gelöscht werden.

DER BEFEHLSVORRAT IN FORTH

DUP	dupliziert den obersten Stackwert TOS (=Top of Stack);
SWAP	vertauscht die beiden obersten Stackwerte;
DROP	löscht die oberste Position im Stack;
OVER	kopiert die vorletzte Zahl in den obersten Stack (TOS);
ROT	vertauscht die oberen drei Stackwerte zyklisch (ROTation);
n PICK	kopiert die n-te Zahl in den TOS;
n ROLL	vertauscht die ersten n-Stackwerte (zyklisch);
MOD	liefert den Rest einer Division;
ABS	Absolutwert einer Zahl;
MIN	bestimmt die kleine von zwei Zahlen;
MAX	liefert den größeren von zwei Werten.

In Forth gibt es außerdem vergleichende Operatoren wie in BASIC: größer, kleiner, gleich.

DO...LOOP
begrenzt Schleifen in Forth analog der For-Next-Schleife in BASIC;

IF..THEN..ELSE
bedingte Verzweigung:

BEGIN...UNTIL
BEGIN...AGAIN
Schleifen mit bedingtem Ende;

```
----------------------------------------------
Programmbeispiel zu Forth:
Listingausschnitt
----------------------------------------------
3 $VARIABLE A$  3 $VARIABLE B$  10 $VARIABLE C$
: CKFILE
  ?FILE IF
    CR."FILENAME " PAD $.. "NOT FOUND"  0
  ELSE 1 THEN ,

: COPYSCR
  CR ."SCREENCOPY FROM : "  A$  INPUT$
  CR. "SCREENCOPY TO   : "  B$  INPUT$
  OPENCHN
  " SCR" A$ $+ C$ $!
  14 8 14 C$ " ,S,R" $+ COUNT OPEN
  CKFILE IF
    " SCR" B$ $+ C$ $!
    13 8 13 C$ " ,S,S" $+ COUNT OPEN
    CKWRSTAT
    14 HERE     1024 GET#     CKWRSTAT
    13 HERE  1+ 1024 PRINT#   CKWRSTAT
  THEN
  13 CLOSE  14 CLOSE  CLOSCHN  CR;
----------------------------------------------
```

Tabelle 9

BEGIN..WHILE..REPEAT
bedingte Schleife;

INPUT
Eingabe von Variablen in das laufende Programm;

GET INKEY
Eingabe eines Zeichens in das laufende Programm.
Außerdem sind Stringmanipulationen möglich wie in BASIC (siehe dort):
LEFT$, RIGHT$, MID$, LEN, ASC, CHR$

FILEHANDLING

OPEN
Öffnen eines logischen Files;

CLOSE
Schließen eines logischen Files;

ABORTIO
Schließen aller noch offenen logischen Files;

GETDEV
holt die Devicenummer eines eröffneten Files zurück;

RDSTAT
liest den Diskettenstatus von der Disk;

CKST
liest den I/O-Status (ST-Variable);

CKRDSTAT
Lesen des Fehlerkanals;

OPENCHN
eröffnet den Fehlerkanal der Diskette;

CLOSCHN
schließt den Fehlerkanal.

Programmbeispiel siehe Tabelle 9.

32. Fortran

Formular Translator (Formel-Übersetzer).
Wie aus dem Namen schon hervorgeht, ist Fortran in erster Linie zur Lösung von technisch-mathematischen Problemen entwickelt worden. Man spricht hier auch von einer „Formelsprache". Fortran ist die heute am meisten verbreitete Programmier-Sprache in diesem Anwendungsbereich.
Fortran bietet aufgrund guter Compilerversionen eine hervorragende Fehlersuche bei der Pro-

grammerstellung. Erkannt und gemeldet werden:

- Anweisungsfehler: Syntaxfehler, semantische Fehler, konstruktive Fehler;
- Programmfehler: Programmumfangsfehler, Compilerfehler, nicht verwendete Variable oder Unterprogramme;
- Laufzeitfehler: Speicherbereichsfehler, Ein-/Ausgabefehler.

Angegeben vom Compiler wird die Art des Fehlers und die Stelle im Programm, an der er auftritt *(vgl. Tabelle 10).*

Es gibt vier verschiedene Arten von *Anweisungen* in Fortran:

1. *Eingabe- und Ausgabeanweisungen.*
   Sie regeln den Datenfluß innerhalb der Rechenanlage und zu oder von den Peripheriegeräten.
2. *Steueranweisungen.*
   Hierdurch wird der Programmablauf manipuliert. Die Steueranweisungen bestimmen die Reihenfolge der Abarbeitung der Befehle und Programmteile.
3. *Spezifikationsanweisungen,* welche die Datentypen festlegen (mit oder ohne Nachkommastellen, numerische oder alphanumerische Daten, Texte . . .).
4. *Arithmetische Anweisungen.*
   Hier finden die eigentlichen Berechnungen statt.

Der Nachteil von Fortran ist hauptsächlich in der etwas schwierigen Handhabung größerer Datenmengen zu sehen, wie sie beispielsweise in kaufmännischen Anwendungsbereichen anfallen.
Fortran wurde 1957 eingeführt und ständig überarbeitet. Die heute gebräuchlichste Form ist Fortran IV mit den ANSI-Erweiterungen.
(ANSI = American National Standards Institute, amerikanischer Normenausschuß).

Tabelle 10:

```
Beispiel eines Anweisungsfehlers in Fortran

SUMME = K1 + K2 + - K3              Programmzeile
E+E+E+E+E+E+E+E+E+$+E+E             Fehlermeldung

E = Error und $ = Stelle des Fehlers / Hier ist daß Minuszeichen der Fehler.
```

Tabelle 10b:

```
Programmbeispiel zu Fortran
Listing

   REAL*8 XA,XE,D,E,XL,ZL,ZR,X1,X*,X,Y
   F(X) = 0.1 * X**3 + 0.5 * X**2 - X - 3.5
   READ (9,1) XA,XE,D,E
 1 FORMAT ($ F10.3)
   XL = XA
   XR = XL + D
   YL = F(XL)
20 YR = F(XR)
   WRITE (10,2) XL,ZL
 2 FORMAT (1H,2X,2 F10.3)
   IF (YL*YR) 10,10,30
10 X1 = XL
   X2 = XR
40 X = 0.5 * (XL + XR)
   Y = F(X)
   IF (ABS(Y) -E) 79,50,50
50 IF (Y*YL) 60,80,80
80 XL = X
   YL = Y
   GO TO 40
60 XR = X
   YR = Y
   GO TO 40
70 WRITE (10,2) X,Y
   XR = X2
   YR = F(XR)
30 XL = XR
   YL = XL + D
   IF (XL-XE) 20,90,90
90 WRITE (10,2) XL,YL
   STOP
   END
```

Zwar ist die Sprache Fortran wie viele andere Sprachen genormt (s.o.), aber im Laufe der Zeit wurden beim Erstellen der Compiler verschiedene Computerhersteller einige Anweisungen modifiziert oder anlagenspezifische Befehle mit aufgenommen, so daß man heute nicht mehr von der Sprache Fortran sprechen kann. Es existieren viele Versionen, man spricht auch von Dialekten.

Normen

- DIN 66.027 „Informationsverarbeitung; Programmiersprache Fortran" (BRD) in deutsch;
- ANSI X3.9.1978 (USA), (früher USAS X3.9-1966);
- ISO 1539–1980 „Programming Language Fortran" (international).

In neuester Zeit ist versucht worden, die Vorteile der verschiedenen Versionen in einem Komplex zusammenzufassen: Fortran 77.
Jedoch wird diese Version auch nur eine von verschiedenen Fortran-Dialekten bleiben, da sie zu spät auf den Markt kommt. Die bestehenden Versionen lassen sich kaum verdrängen und viele Studenten, die heute die Grundlagen des Programmierens erlernen sollen, studieren dies mit einer alten Fortran-Fassung. Hinzu kommt, daß Anlagen, die beispielsweise mit Fortran IV-Compilern laufen, sich in den meisten Fällen zwar auf Fortran 77 umstellen lassen, dies aber mit hohen Kosten verbunden ist.
Folgende **Befehle** gehören zum Fortran-Vokabular:

Steueranweisungen:
GO TO
IF
DO
CONTINUE
CALL
RETURN
PAUSE
STOP
END
DIMENSION

E/A-Anweisungen:
READ
WRITE
ACCEPT
DISPLAY
ENCODE
DECODE
FORMAT
REWIND
BACKSPACE
END FILE
DEFINE FILE

Operatoren:
\+ Addition
– Subtraktion
* Multiplikation
/ Division
**Potenzierung

Bibliotheks-Unterprogramme:

SQRT	Wurzelfunktion
SIN	Sinusfunktion
COS	Cosinusfunktion
ATAN	Arcustanges
EXP	Exponentialfunktion
ALOG	Logarithmusfunktion.

Variablen:
In Fortran wird streng unterschieden zwischen den sogenannten Integer-Variablen (ganze Zahlen) und den Real-Variablen (Zahlen mit Nachkommastellen).
Schon bei der Programmierung muß festgelegt werden, welche Variablen und Konstanten zum Typ Integer oder zum Typ Real gehören.

Es gibt zwei Möglichkeiten, die Variablen (und Konstanten) zu definieren:

1. *Implizit:*
   Beginnt ein Variablenname oder Konstantenname (Buchstaben A bis Z und Zahlen 0 bis 9) mit einem der folgenden Buchstaben:
   I J K L M N
   so gehört er automatisch zum Typ Integer, wenn keine anderen Vereinbarungen getroffen werden. Alle anderen Namen gehören dem Typ Real an.
   Beispiel:

SUM	= 5.11	Real
P	= 6.9	
X	= 1.2	
PI	= 3.14	
I	= 2	Integer
JA	= 1	
NUMMER	= 123	
NN	= 8.	

2. *Explizit:*
   Am Programmanfang werden die verwendeten Variablen oder Konstanten definiert. Für allen übrigen Variablen und Konstanten gilt die implizite Typ-Vereinbarung.
   Beispiel:
   ■ Anweisung:
   INTEGER SUMME, X, Y, NN
   REAL MM, Z, KA.
   ■ Ganzzahl-Variable:
   SUMME, X, Y, NN und alle übrigen Variablen, deren Name mit I J K L M oder N beginnt.
   ■ Gebrochene Variable:
   MM, Z, K und alle anderen Variablen mit Ausnahme der oben genannten.

In Fortran werden häufig verwendete Programmteile in sogenannte Unterprogramme geschrieben, die über das Hauptprogramm aufgerufen werden können. Es gibt hier verschiedene Unterprogrammformen, die je nach den Erfordernissen Einsatz finden.
Die sogenannten Bibliotheksunterprogramme (siehe oben) gehören standardmäßig zum Fortran-Vokabular und werden einfach mit ihrem Namen und dem Parameter aufgerufen, zum Beispiel wird der Sinus von 15.5° berechnet mit dem UP-Aufruf:

```
x = sin (15.5)
```

Der Sinuswert von 15.5 wird daraufhin berechnet und in der Variablen X abgespeichert und kann dort abgerufen werden.
Die Möglichkeit der Unterprogrammbildung gibt es auch in anderen Programmiersprachen, jedoch ist bei Fortran zu beachten, daß Variablen im Hauptprogramm nicht den gleichen Wert enthalten wie im Unterprogramm, es sei denn, es wird ausdrücklich gewünscht (Parameterübergabe). Anders ist dies beispielsweise in BASIC, wo eine einmal belegte Variable ihren Wert beibehält (auch in Unterprogrammen), bis sie durch eine Anweisung geändert wird.
Sogar nach der Beendigung des BASIC-Programmes können die Variablen noch abgefragt werden, was in Fortran nicht mehr möglich ist.

Programmbeispiel:
Nullstellenermittlung

Die Nullstellen einer vorgegebenen Funktion sollen innerhalb eines gewählten Intervalles berechnet werden. Die Funktion wird im Programm definiert mit:

```
f(X) = 0,1 * X**3 + 0,5
       * X**2 – X – 3,5
```

Die Grenzen, in denen die Funktion untersucht werden soll, werden auf einer Datenkarte angegeben, ebenso die Schrittweite, denn es soll außerdem eine Wertetabelle ausgedruckt werden.

XA Untere Grenze
= – 8,00
XE Obere Grenze
= + 5,00
D Schrittweite
= 1,00
E Genauigkeit
= 0,001.

Programmbeispiel siehe Tabelle 10b.

Der Drucker hat hier im Beispiel die Geräte Nr. 10, der Kartenleser die Nr. 9 (READ und WRITE-Anweisungen). Bei der vorgegebenen Funktion und den gegebenen Grenzen sowie der Schrittweite erhält man folgende Ausgabe auf dem Drucker:

–8.000	–14.700	
–7.000	–6.300	
–6.000	–1.100	
–5.675	–0.000	– –
–5.000	1.499	erste Nullstelle
–4.000	2.100	
–3.000	1.300	
–2.168	0.000	– –
–2.000	–0.299	zweite Nullstelle
–1.000	–2.099	
–0.000	–3.499	
0.999	–3.900	
2.843	0.000	– –
2.999	0.699	dritte Nullstelle
3.999	6.899	
4.999	16.499	

Im Rahmen der hier vorgegebenen Rechengenauigkeit ist Zahl 0.999 = 1.000, die 1.999 = 2.000 und so weiter.
Das negative Vorzeichen vor der 0.000 hat für die Auswertung keine Bedeutung.
Die Nullstellen dieser Funktion liegen demnach bei:

x1 = –5.675
x2 = –2.168
x3 = +2.843

Ob alle Nullstellen der Funktion getroffen werden, hängt bei diesem Programm im wesentlichen davon ab, welches Intervall gewählt wird. Wird die Genauigkeit erhöht, verlangsamt sich die Rechengeschwindigkeit.

33. GPL
Graphical Programming Language.
Spezialentwicklung für die Programmierung grafischer Probleme.

34. HOL
System-Implementationssprache für Prozeßrechner des US-Verteidigungsministeriums, die sich noch in der Entwicklung befindet.
HOL ist maschinenunabhängig und dürfte aufgrund dessen von großer Bedeutung sein, da eine einheitliche Systemimplementierung bisher noch nicht möglich war. Man hofft, die mit Hilfe von HOL erreichen zu können.

35. HOPE
Deklarative Sprache.
HOPE wurde entwickelt an der University of Edinburgh. Über ein sogenanntes Regelwerk werden Beziehungen zwischen den betrachteten Daten sowie der Datenfluß an sich beschrieben (siehe auch LISP oder PROLOG).

36. Lisp
List Processing Language.
Höhere Programmiersprache mit freier Operatorwahl und Meta-Compilation.
Lisp hat einen compilierenden Interpreter (siehe auch Forth).
Lisp ist eine Programmiersprache, die zur Erforschung der künstlichen

Intelligenz eingesetzt wird, das heißt, in Lisp werden unter anderem Lernprogramme für Maschinen und Systeme geschrieben. Die Programme ändern sich je nach den Anforderungen, sie können „vergessen", was nicht benötigt wird, um Speicherplatz für das „Dazugelernte" zu schaffen.
Lisp ermöglicht hochpräzise Rechnungen in jedem Zahlensystem mit einer Basis aus dem Bereich von 2 bis 36. Durch eine dynamische Speicherverwaltung wird der benutzte Speicherplatz optimiert, so daß auch extrem komplexe Fragestellungen gelöst werden können. Die erste Implementierung von Lisp auf einen Microcomputer war Lisp 80 von der Firma Microsoft.
Lisp gibt es bereits seit Ende der 50er Jahre, wurde von John McCarthy am Massachusetts Institute of Technologie geschaffen. Sie gehört seit Anfang 1970 zu den grundlegenden Werkzeugen für die Erforschung der künstlichen Intelligenz.

KÜNSTLICHE INTELLIGENZ

Lisp unterscheidet sich von anderen höheren Programmiersprachen in erster Linie darin, daß numerische Operationen nicht die Basis für die Programmausführung sind. Lisp ist daher besonders prädestiniert für die Manipulation symbolischer Daten.
Stellt man sich Symbole als Worte vor, dann sind Listen von Symbolen nichts anderes als Sätze und Aussagen. Da die Symbolverarbeitung und die Fähigkeit zur Wissensdarstellung relativ einfach zur Repräsentation von Schablonen und zu Assoziationen des menschlichen Denkens genutzt werden können, wurde Lisp zu einem wesentlichen Werkzeug bei den

```
Programmbeispiel zu Lisp
Listing Ausschnitt
-----------------------------------------------------
(defun exchange-minus(expression)
(selectq(expression-type expresssion)
(expression-with-1-arg)
(undefinded 'undefinded)
(0 0)     ;(-0) -0
(otherwise(let((first-arg(exchange(operrand1 expression))))
(if(equal(operrand1 expression)first arg)
expression
(exchange-minus(make-expression-with-1-arg)
operator 'minus
operand1 first-arg))))))
(expression-with2-args
(if(or(eq (operand1 expression) 'undefinded)
'undefinded
(let((first-arg (operand1 expression)))
(if equal first-arg 0)
(exchange (make-expression-with-1-erg)
operator 'minus
operand1 (operand2 expression)))
(if(equal first arg (operand2 expression))
(selectq (operand2 expression)
(0(exchange (operand1 expression)))
.
.
.
```

Tabelle 11

Versuchen, menschliches Verhalten und Denken zu simulieren. Darüber hinaus ist Lisp allerdings auch eine Sprache für andere universelle Anwendungen.
Wenn auch die Programmierung in Lisp nicht gerade als einfach bezeichnet werden kann, so stehen dem Programmierer doch eine Vielzahl nützlicher Hilfs-Werkzeuge zur Verfügung; so sind compilierter und interpretierter Code jederzeit ausführbar, auch Teilprogramme können ausgeführt und zur Fehlersuche ausgetestet werden.
Die Konzeption von Lisp ist, wie der Name schon sagt, auf die Verarbeitung von Listen abgestellt, aber auf eine ganz andere Art wie beispielsweise bei den Listen-Generatoren (siehe RPG oder LPG).
Der Programmierer kann hier darauf verzichten, eine Programmstruktur entwickeln zu müssen. Diese ist Bestandteil der Programmiersprache, und man kann sich bei der Programmierung ganz auf die logische Seite des Problems konzentrieren. Siehe hierzu auch die Ausführungen zu PROLOG.
Aus Frankreich kommt die dort sehr stark verbreitete Version Le-Lisp dieser schnellen KI-Sprache. Le-Lisp gibt es seit 1986 auch in Deutschland.

Programmbeispiel siehe Tabelle 11.

37. Logo

Logo ist ein weiterentwickelter Dialekt der Programmiersprache Lisp, jedoch wesentlich weiter verbreitet als diese. Logo wurde 1967/68 von Seymour Papert für Kinder und Anfänger konzipiert.
In Europa hatte Logo einige Startschwierigkeiten, da es nur auf Großcomputern sinnvoll angewandt werden konnte, denn für diese wurde es entwickelt. Doch Kinder haben eben selten Zugang zu Großcomputern.
In den USA hatte Logo jedoch große Erfolge in einem Großversuch in den Jahren 1970/80.
Hier in Europa ist Logo erst seit etwa 1980 bekannt. Dennoch kann sich diese hervorragende Programmiersprache kaum gegen die verbreiteten konventionellen Sprachen durchsetzen, obwohl es für die meisten Microcomputersysteme Implementierungen von Logo gibt, sogar für einige Homecomputer.
Logo ist eine typische Interpretersprache wie BASIC. Besonders unterstützt werden grafische Anwendungen und vor allem die Computerperipherie bis hin zum Joystick.

GRAFISCHE ANWENDUNGEN

Logo ist leicht lesbar und kurz. Die Programme werden in Prozeduren aufgeteilt (modulare Blöcke) und sind top-down strukturiert, das ist ein entscheidender Vorteil gegenüber BASIC.
Die Nachteile von Logo liegen eindeutig in seinen Schwächen beim interaktiven Rechnen. Mathematisch-naturwissenschaftliche Probleme bleiben Probleme in Logo.
Logo stellt folgende Grundoperationen und Funktionen zur Verfügung:

\+ Addition
– Subtraktion
* Multiplikation
/ Division.

Negative Zahlen sind in Klammern zu schreiben, ansonsten gilt „Punktrechnung vor Strichrechnung".

Ganzzahloperationen:
ROUND :X

rundet die reelle Zahl X auf die nächste ganze Zahl;

INTEGER :X
schneidet von der reellen Zahl X die Nachkommastellen ab;

QUOTIENT :X :Y
ergibt den ganzzahligen Anteil der Division von X durch Y an.

Funktionen:
SQRT :X
Quadratwurzel aus der Zahl X;

SIN :W
Sinuswert des Winkels W (in Grad);

COS :W
Cosinuswert;

ATAN :X
Umkehrfunktion des Tangens;

RANDOMIZE
Initialisierung des Zufallszahlengenerators;

RANDOM :N
erzeugt eine ganze Zufallszahl zwischen 0 und N.

```
Programmbeispiel zu Logo
Listing

TO PRIMZAHLEN
     CLEARTEXT
     PRINT (PRIMZAHLENTABELLE) PRINT()
     PRINT (UNTERE GRENZE?) MAKE "M FIRST
     REQUEST
     PRINT (OBERE GREZE?) MAKE "N FIRST
     REQUEST
     IF (REMAINDER :M2) = 0 THEN MAKE "M :M+1
     PRINT()
END

TO PRIM :M :N
     IF :M ) :N THEN PRINT () STOP
     MAKE "WURZEL SQRT :M
     MAKE "TEILER 3
     PRIMTEST :M :WURZEL :Teiler
     PRIM :M + 2 :N
END

TO PRIMTEST :M : WURZEL : TEILER
     IF : TEILER³ :WURZEL + 1 THEN PRINT1 : M PRINT (,) STOP
     IF REMAINDER :M :TEILER) = 0 STOP
     PRIMTEST :M : WURZEL : TEILER + 2
END
```

Tabelle 12

In Logo sind keine Variablen-Deklarationen notwendig. Logo kennt globale und lokale Variable. Eine Variable kann einen Zahlenwert, einen Booleschen Wert, ein Wort, einen Satz beliebiger Länge, eine Liste von Elementen der verschiedensten Typen und sogar Prozedurnamen enthalten. Gerade Programmieranfängern dürfte dies sehr entgegen kommen. Das Handling von Listen, eine höhere Form von Arrays, ist das geeignete Mittel für komplexere Probleme der künstlichen Intelligenz – mit Logo sogar schon auf Homecomputern (in gewissen Grenzen, versteht sich).
Logo nimmt unter den höheren Programmiersprachen eine Sonderstellung ein. Logo wurde nicht für bestimmte Anwendungen entwickelt, sondern als Trainings-Sprache zum Erlernen des Programmierens in Form einer geistigen Disziplin; also zum Schulen von bestimmten Denkmechanismen, die für eine moderne Programmierung in jeder Programmiersprache von Vorteil sind.
Logo sollte daher nach den Wünschen ihres Entwicklers die allererste Sprache sein, mit der ein Anfänger in die Welt der Programmierung einsteigt, um später eine andere zu lernen.
Genau dieses Konzept wurde versuchsweise an amerikanischen, englischen und französischen Schulen eingeführt, mit großem Erfolg.
Es gibt einige Vorteile von Logo, die es sinnvoll machen, den oben beschriebenen Weg einzuschlagen. Im folgenden werden die wichtigsten aufgezählt:

- In Logo sind die Prozeduren unabhängig voneinander aufrufbar und müssen nicht über ein Hauptprogramm miteinander verknüpft sein. Ein Logo-Programm kann also direkt am Bildschirm entworfen werden.
- Die kurzen Module können während der Eingabe fortlaufend überprüft und festgehalten werden. Problemlos lassen sie sich korrigieren und verfeinern. Bei Aufruf des Programms mit dessen Hauptnamen werden die Module automatisch verkettet.
- Logo eignet sich ähnlich wie BASIC für den Dialogbetrieb, da es nicht compiliert, sondern interpretiert wird, was andererseits zu Lasten der Geschwindigkeit geht. Dies ist jedoch für eine „Lernsprache" untergeordnet.
- Logo unterstützt Rekursionen. Schleifen werden nicht iterativ, sondern rekursiv dargestellt.
- Logo stellt dem Anwender standardmäßig eine hochauflösende Grafik zur Verfügung, was viele andere Sprachen nicht können.
- In den meisten Programmiersprachen ist die Syntax und deren Umfang genau vorgegeben. Auch in Logo gibt es natürlich einen festgelegten Grundwortschatz, jedoch kann der Anwender beliebig neue Grundwörter definieren, ähnlich wie in Forth (siehe dort). Dadurch kann der Sprachschatz erweitert und die Programmierumgebung den spezifischen Anwenderanforderungen angepaßt werden.

Den oben genannten Vorteilen stehen aber auch eine Reihe von Nachteilen gegenüber:

- Logo kennt keine indizierten Variablen, so daß nicht wie in anderen Sprachen bequem auf bestimmte Elemente eines Feldes zugegriffen werden kann. Logo stellt hierzu die rekursive Listenbearbeitung bereit.
- Es stehen kaum vernünftige Ausgabe-Formatierungs-Befehle zur Verfügung, so daß es in Logo schwierig werden kann, Listen und Tabellen übersichtlich auf dem Bildschirm darzustellen. Es fehlen auch einige, je nach Anwendung, wichtige mathematische Funktionen, beispielsweise zum Potenzieren, Tangensfunktionen, Umkehrfunktionen zu Sinus und Cosinus sowie logarithmische Funktionen und Exponentialfunktionen. Ohne diese sind allgemeine mathematisch-naturwissenschaftliche Programme kaum zu erstellen. Daher gibt es einige Ergänzungen zu Logo, aber nicht für jeden Rechner. Logo ist eben nicht die für diesen Bereich prädestinierte Programmiersprache.

Vor- und Nachteile dürften sich weitgehend auf-

heben, so daß wieder die oben angesprochenen Gründe für Logo als „Einsteiger-Sprache“ zählen.

Programmbeispiel siehe Tabelle 12.

38. LPG

Listen Programm Generator (siehe auch RPG). Ein Programm-Generator ist ein System, welches von Dateien eingelesene Datensätze nach vorbestimmten, gleichartigen Umformungen bearbeitet und in Form einer Liste ausgibt. Haupteinsatzgebiet für solche Programmiersprachen liegen im kommerziellen Bereich.

39. MACRO 80

MACRO 80 ist ein verschiebbarer Assembler (siehe dort) für den Prozessor 8080, 8085 und Z80. Dieser Assembler unterstützt alle Macros nach dem Intel-Standard. Dabei wird die Schachtelung der Macros nur durch den vorhandenen Speicherplatz begrenzt.
Der Objektcode wird in verschiebbarer Form generiert, die durch den Lader weiterverarbeitet wird. Eine bedingte Assemblierung wird ermöglicht durch Pseudo-Operationen. Bedingungen können bis zu 255 Stufen geschachtelt werden.
MACRO-Definitionen erlauben es, mehrere Befehlszeilen als Macro in einer Bibliothek abzuspeichern und jeweils durch einen Aufruf in das Quellprogramm einzuschließen. Besonders interessant ist auch die Möglichkeit, in der Bibliothek Macros abzuspeichern, die Maschinencode für jede beliebige Acht- oder 16-Bit-Maschine unabhängig vom 8080-Code generieren können.
MACRO-80 benötigt rund 14 KByte Speicherplatz und es werden etwa 1000 Zeilen pro Minute assembliert.

```
-----------------------------------------------
Programmbeispiel zu Modula
Listing
-----------------------------------------------
  FROM TERM1 IMPORT
     WriteString, WriteCard, Writeln;

  CONST
     Size       = 7244;        (* Arraygröße *)
     Iterations = 15;          (* minimum 1  *)

  VAR
     count, i, iter, k, prime:  CARDINAL;
     flags: ARRAY(0..Size-1) OF BOOLEAN,

BEGIN
  WriteString('15 Iterations'); Writeln
  FOR iter :=1 TO Iterations DO
     count := 0;
     FOR i := 0 TO Size DO flags(i) := TRUE; END;
     FOR i := 0 TO Size DO
        IF flags(i) THEN
           prime := i * 2 + 3;
           k := i + prime;
           WHILE k = Size DO
              flags(k) := FALSE,
              INC(k, prime);
           END;
           INC(count);
        END;
     END; (* FOR *)
  END; (* FOR *)
  WriteString(' There were ');
  WriteCard(count, 0), WriteString(' primes ');
End BEISPIEL
-----------------------------------------------
```

Tabelle 13

40. MANTIS

MANTIS ist eine höhere interpretative, prozedurale Sprache, welche die Vorteile von BASIC (leicht erlernbar, interpretativ) und Pascal (Strukturierung und String-Manipulation) in sich vereint. Hinzu kommen Elemente aus APL: Operatoren und Mächtigkeit.
Aus diesen Gründen ist MANTIS sehr flexibel.
MANTIS wurde entwikkelt von „Cincom Systems“ zur Produktivitätssteigerung beim Programmieren. Es wird daher oft mit den Tools verglichen, die mit Sprachen wie Cobol zusammenarbeiten. MANTIS ist jedoch eine richtige höhere Programmiersprache. Integrierte Help-Funktionen und Menüsteuerungen machen MANTIS sehr benutzerfreundlich.

Der vollständige Befehlsvorrat ermöglicht eine übergreifende Anwendungsbereich-Abdeckung. Die Stärke von MANTIS liegt in seiner Effizienz. Leistungsfähige Gleitkomma-Arithmetik und ein spezieller Interpreter, der mit speziellen Techniken arbeitet und sich deutlich von anderen Interpretern abzeichnet, sind die Vorteile dieser relativ neuen Programmiersprache.
Innerhalb kurzer Zeit gab es von MANTIS Implementationen für die gängigsten Rechnertypen. Von IBM-Rechnern unter DOS über VM-CMS und MVS gibt es MANTIS nun auch für die VAX von DEC, WANG/VS und Honeywell-Bull.
MANTIS-Programme sind sehr flexibel und weisen eine erstaunlich hohe Portabilität auf. Sie sind völlig unabhängig vom Betriebssystem.
MANTIS vereinigt in sich alle fortschrittlichen Technologien: vom interaktiven Bildschirmentwurf über die Darstellungsform des sogenannten „Filmablaufes“ bis hin zur vollständigen Simulation von Anwendungen, einschließlich der Darstellung aller Datenflüsse und der programmierten Entscheidungslogik. Die dabei angewandte Verbindung zwischen Top-down- und Bottom-up-Lösungsansatz führt zu ausgezeichneten Programmen mit relativ geringem Wartungsaufwand.
MANTIS erlaubt auch bei komplexeren Problemen schnellere Lösungen als mit herkömmlichen Programmiersprachen.

41. Modula 2

Der Name dieser höheren Programmiersprache weist schon hin auf den modularen Aufbau der Programme.
Modula 2 entstand aus einer Studie zur Programmierung gleichzeitig ablaufender Prozesse: Modula 1.
In Modula 1 haben die Module die Aufgabe sogenannter Monitore, Gruppierungen gemeinsamer Variablen und ihrer Operatoren. Minitore sind als kritische Programmabschnitte gedacht, welche nicht gleichzeitig von verschiedenen Prozeduren durchlaufen werden dürfen.

Die allgemeine Bedeutung und letztlich wichtigere für die Gliederung von Programmsystemen wurde bei der Entwicklung erst später erkannt.
Modula ist als „Nachfolger" von Pascal von Prof. Klaus Wirth am Institut für Informatik (IfI) in Zürich entwickelt worden und erhielt sehr schnell das Prädikat: „Pascal ohne die Nachteile von Pascal".
Nicht nur der Programmaufbau ist modular, sondern auch die Compilation erfolgt modulweise. Die Sprachdefinitionen sind auf ein Minimum beschränkt, dafür wird der Aufbau von Modulbibliotheken unterstützt. Dies wird erreicht durch eine gesonderte Schnittstellenbeschreibung, welche als gesondertes Modul existiert, je nach Implementation, also Installation auf einem bestimmten Rechnertyp.
Folgende Merkmale von Modula 2 unterscheiden diese wesentlich von anderen höheren Programmiersprachen:

- Namen können beliebig lang sein;
- Groß- und Kleinschreibung können unterschieden werden;
- Konstantenarithmetik;
- Reihenfolge der Deklarationen ist nicht streng vorgeschrieben;
- Open Arrays in Prozedur-Deklarationen;
- ELSE-Variante beim CASE-Statement.

Modula 2 setzt einigen Aufwand an Hardwareeinsatz voraus: mindesten 56-Kbyte-Arbeitsspeicher, zwei Floppy-Laufwerke mit je 350-KByte-Kapazität. (Die minimale Konfiguration der notwendigen Files für den Compiler, den Linker und den Editor ist bereits 226 KByte groß.)
Modula 2 ist ursprünglich für den Z80-Prozessor entworfen worden, unter dem Betriebssystem CP/M80.
Wie bei vielen anderen compilierbaren Programmiersprachen muß für Modula 2 jeweils separat

- der Texteditor – zum Erstellen und Eingeben von Programmfiles;
- der Compiler – zum Übersetzen der einzelnen Module und
- der Linker – zum Binden der einzelnen Module und Bibliotheksunterprogramme

aufgerufen und gestartet werden.
Die Bibliothek von Modula 2 besteht ihrerseits ebenfalls aus Modulen und kann vom Anwender ergänzt werden.
Das hat zur Folge, daß im Laufe der Zeit die einzelnen Modula 2-Programme immer kürzer werden können, da auf eine wachsende Anzahl von fertigen Problemlösungen in Form von Unterprogrammen zurückgegriffen werden kann.
Die Bibliotheksmodule liegen dann bereits in übersetzter Form vor und brauchen nur noch an das neu geschriebene Programm gebunden zu werden (Link File).
Der Compiler besteht aus vier Compiler-Pässen, das heißt, es sind vier Durchläufe zur Übersetzung eines Modula 2-Programmes nötig.
Eine Aufstellung der beim Übersetzen gefundenen Fehler wird in einem gesonderten File (LISTE) abgelegt und kann dort zur Fehlerbeseitigung eingesehen werden.
Hat das Programm die vier Compilerpässe fehlerfrei durchlaufen, können die einzelnen Programm-Module untereinander und natürlich auch mit den Bibliotheksmodulen gebunden werden. Hierzu sind zwei weitere Pässe notwendig, die der Linker steuert.

Programmbeispiel siehe Tabelle 13.

```
;---------------------------------------
;Programmbeispiel zu Modula
;Assembler-Implementation
;---------------------------------------
;****************************************
;*     IMPLEMENTATION MODULE Ports      *
;****************************************
;
        name ('ports')
        title ports
        .z80
;
;***** EXPORT ***************************
;
inp:    pop iy  ; get return address
        pop hl  ; address of 'data' in hl
        pop bc  ; c contains 'portAdr'
        in a, (c)
        ld (hl), a
        jp (iy) ; instead of push, ret
;
outp:   pop iy  ; get return address
        pop de  ; 'data' is in e
        pop bc  ; /portAdr' is in e
        out (c), e
        jp (iy) ; instaed of push, ret
;
;****************************************
;
        end
;---------------------------------------
```

Tabelle 14

Modula 2 bietet auch ein komfortables **Assembler-Interfacing.** Hierzu sind allerdings sehr genaue Kenntnisse über die Stacks und den Aufbau der Variablen notwendig.
Modula ist eine moderne Programmiersprache, die sich besonders zur Konstruktion von großen, modularen Software-Systemen eignet. Der modulare Aufbau der Programme hat sich als sehr vorteilhaft erwiesen und ist gemeinsam mit der separaten Übersetzung der Module als Garantie für effizienten Unterhalt und dauernde Konsistenz der Software verantwortlich.
Obwohl einfach in seiner Konzeption, hat sich das Modul als ein überraschend schwierig zu handhabendes Hilfsmittel herausgestellt.
Module treten in den verschiedensten Erscheinungsformen auf, so zum Beispiel als Package von Prozeduren, als Verwaltungsinstanz von Ressourcen, als Schnittstelle zur Hardware, als „Black Box" zur Verarbeitung von Objekten als Monitor und als Datenbasis.
Das modulare Denken eröffnet neue Horizonte und führt zu weitreichenden Konsequenzen, sogar für Betriebssystem-Designer und Computer-Architekten.

Programmbeispiel siehe Tabelle 14.

42. mu-SIMP

High-Level-Programmiersprache für symbolische und halbnumerische Datenverarbeitung.
mu-SIMP benutzt einen schnellen Interpreter, der nur wenig Speicher-

platz benötigt (7 KByte). mu-SIMP ist eine Spezialentwicklung und daher kaum verbreitet: Das Hauptanwendungsgebiet ist die wissenschaftliche Mathematik.

43. Natural

Strukturierte non-prozedurale Programmiersprache mit Schnittstellen zu Cobol und PL/1. Damit können Natural-Programme von Cobol- oder PL/1-Programmen aufgerufen werden, beziehungsweise umgekehrt.
Natural ist im Vergleich zu den oben genannten eine relativ „junge" Programmiersprache, jedoch keine Weiterentwicklung von Cobol oder PL/1.
Natural ist strukturierend (zum Beispiel kein GOTO-Befehl) und unterstützt die modulare Programmierung.
Nachteile von Natural: es können nur eindimensionale Tabellen verarbeitet werden.
Erstaunlich komfortabel hingegen ist die interaktive Programmierumgebung mit Compiler, Testprogrammen, Maskengenerator, Editor, Bibliotheks- und Datenverwaltung, DATA-Dictionary und Zugriffs-Schutzmöglichkeiten.
Einsatzgebiet für die Formuliersprache Natural sind unter anderem Datenbanken.
Natural ist mehr als eine der „neuen Programmiersprachen" (sie wird zur 4. Generation gezählt).
Natural stellt eine komplette Programmierumgebung bereit. Integriert sind Compiler, Editor, Maskengenerator, Testfunktionen, sowie Bibliotheks- und Datenverwaltung, Data-Dictionary und Zugriffsschutzmöglichkeiten. Teilweise werden also schon Betriebssystem-Aufgaben übernommen. Natural unterstützt durch seine hervorragende Menütechnik das Prototyping von Programmen.
Die Handhabung von Natural ist nicht ganz unproblematisch, daher erscheint diese Programmiersprache nicht unbedingt geeignet für den unerfahrenen Programmierer. Optimal programmiert sind Natural-Applikationen jedoch extrem schnell und benutzerfreundlich.

44. NEAT

NCR Elektronik Autocoding Technique.
Maschinenorientierte Spezialentwicklung der Firma NCR. NEAT ist eine NC-Sprache (Numeric Control) für die Steuerung von Werkzeugmaschinen.

45. NELIAC

Ein Algol-Dialekt (siehe unter Algol).

46. NPL

New Programming Language.
Vorläufer der Programmiersprache PL/1 (siehe dort), und damit heute nicht mehr gebräuchlich.

48. OPS-5

OPS-5 ist eine Eigenentwicklung der Firma DEC (Digital Equipment Corporation).
Sie eignet sich besonders für die Anwendung in Expertensystemen und für Bereiche der kognitiven Wissenschaften im Rahmen der künstlichen Intelligenz. Darüber hinaus unterstützt OPS-5 auch die Lösung komplexer Probleme im kommerziellen Umfeld.
Expertensysteme sind Datenbanken, welche sich aber von den konventionellen Datenbanken durch ihr „intelligentes" Verhalten unterscheiden.
Expertensysteme sollen die fachliche Kompetenz von Experten in Form von Sach- und Erfahrungswissen bereitstellen.
Zu diesem Zweck wird das Wissen nicht nur in Form von Daten abgespeichert, sondern auch durch Regeln verknüpft, klassifiziert und bewertet. Die Systeme sollen darüber hinaus heuristische Methoden und vages Wissen verarbeiten und nach den vorgegebenen Regeln aus dem vorhandenen Datenmaterial selbständig Schlüsse ziehen, das heißt Problemlösungen anbieten.
Die Systeme können so dem Anwender bei der Problemanalyse helfen, sie dokumentieren ihre Vorgehensweise.
Ein professionelles Expertensystem besteht aus fünf Abschnitten:
1. Die „Wissensbasis" ist der Grundstock. Hier ist das Experten-Wissen in Form von Fakten und Regeln abgespeichert.
2. Die Abteilung „Wissenserwerb" beinhaltet Instrumente zur Aufnahme und Integration von neuen Expertenwissen in das System, also hier in die Wissensbasis. Diese Instrumente formulieren die neuen Informationen so um, daß erkannt werden kann, ob dieses Wissen bereits im System vorhanden ist. Lücken in der Wissensbasis können erkundet werden, usw.
3. Ein weiterer Abschnitt des Systems dokumentiert dem Anwender, welcher Lösungsweg verfolgt wird und welche Ergebnisse das System ermittelt hat und mögliche Schlußfolgerungen daraus.
4. Die wichtigste Komponente des Systems ist aber die "Inference Machine", welche die Problemlösung steuert. Sie verarbeitet das Expertenwissen der Wissensbasis und zieht hieraus Schlußfolgerungen und Ableitungen. In bestimmten Situationen kann die "Inference Machine" auch beim Anwender nachfragen, also Zusatzinformation zum erfragten Problem abrufen, um eindeutigere Schlußfolgerungen ziehen zu können.
5. Die Benutzerschnittstelle „Dialog" verbindet das System schließlich mit der Außenwelt, also dem Anwender. Hier können Eingaben in das System erfolgen und hier gibt das System auch Informationen aus.
Expertensysteme haben das Ziel, sowohl objektive Fakten, als auch menschliches Erfahrungswissen zu verarbeiten.
Sie leisten also wesentlich mehr, als einfache Datenbanken, die wie große Nachschlagewerke funktionieren.
Auf diese Weise sollen Probleme gelöst werden, die aufgrund fehlender exakter Verfahren bisher nur durch erfahrungs- und bewertungsabhängige menschliche Entscheidungen gelöst werden können. Doch an dieser Stelle wieder zurück zu einer Programmiersprache, in der solche Systeme erstellt werden können.
Der OPS-5-Compiler wurde in der code-optimierenden Sprache BLISS-32 geschrieben (siehe dort).
Der Anwender erhält damit einen leistungsfähigen kompakten Code.
OPS-5 ist eine typische KI-Sprache (künstliche Intelligenz), die mit Symbolen arbeitet. Sie kann eine Vielzahl von Lösungswegen erzeugen und damit für mehrere Probleme jeweils mehrere Lösungen erarbeiten.
Ein solches KI-Werkzeug bietet sich als computergestützter Lösungsweg an, wenn sich die geplante Anwendung konventionellen, auf Algorithmen beruhenden Lösungen entzieht oder der Problembereich häufigen Veränderungen unterworfen und nur schwer oder gar nicht einzugrenzen ist.
OPS-5 ist auch eine Sprache zur Entwicklung von Regelsystemen. Sie ermöglicht Problemlösungen unter Ausnutzung menschlicher Erfahrungen und menschlichen Wissens.

Fortsetzung im nächsten Heft

BÖRSE

SUCHE 8/9-Nadel-Matrixdrucker m. Centronics bis 200,– DM. T. Seibt, Margaretenanger 3c, 8044 Lohhof

---

SUCHE Turbo-Pascal V. 3.0 m. Handbuch (nur Orig.), Tel. 07171/85226, J. Grandy

---

SUCHE SPIELE (nur 3"-Disk), vor allem Worl Games, Winter Games, California Games, Reisende im Wind I u. II, Miami, Hunter of the Red October. Listen an Thomas Rieger, Fichtenstr. 15, 8261 Engelsberg, Tel. 08634/1402

CPC ZU VERKAUFEN! CPC 464, GT64, Monitor, 3"-Floppy, Lightpen, Data-Becker-Bücher, ca. 20 Zeitschriften, 20 Kass. m. Software, 30 Disks mit Software, 10 Leerdisketten, ca. 25 Orig.-Spiele auf K. u. D. Preis: 800,– DM. Tel. 06224/10866

SUCHE Buchführungs- u. kaufm. Prog., Wirtschaftsanw. Tausche Software. D. Maron, Bahnstr. 26, 2085 Wuickborn

---

Tausche Spiele f. CPC 6128, 3" u. 5.25". Habe viele Spiele! Schickt Eure Listen an: Malte Kroll, Haidweg 17, 2090 Winsen/Luhe. Tel. 04171/72397 od. 04171/75575

*** AN ALLE CPC-USER *** Tausche Spiele aller Art! Schickt Eure Listen an: Marcel Gogolin, Feldstr. 40, 2802 Ottersberg. 100 % Rückantwort!

PUBLIC-DOMAIN-USER-GRUPPE * PD-Programme f. CPC, Joyce u. IBM zu geringem UKB abzugeben. Prog.-Autoren, die Prog. in die Public Domain bzw. Shareware geben wollen, sind willkommen. Info gg. 2 80-Pf.-Briefm., DOS-Kat.-Disk 3 80-Pf.-Briefm. PDUG, Postfach 1118, 6464 Linsengericht

---

SUCHE FÜR CPC 6128 Turbo Pascal 3.0. Tel. 06326/8088

SUCHE f. CPC 6128 Turbo Pascal 3.0. Tel. 06326/8088

TAUSCHE SOFTWARE auf dem CPC. Habe über 300 Games! Schickt Eure Listen an: Markus Kehle, v.-Freybergstr. 61, 8958 Füssen. Disk only

---

SUCHE SPIELE (nru 3"-Disk), vor allem Hanse, Guild of Thieves, Gee Bee, Outrun, Platoon, EYE (tausche D. u. C.). Listen an Manfred Hauser, Waldstr. 2, 8261 Engelsberg. Tel. 08634/5308

VERKAUFE Trojan Lightpen m. Software DM 15,–; Laser-Farbdrucker anschlußfertig an CPC 464/664 DM 180,–; Vortex F1-X f. CPC 464 DM 400,– (LW neu m. 100 Disks); CPC-464/664/6128-Computer Preis VB. Tel. 08334/1513

CPC 6128, Vortex F1-X, Farb- u. Grünmon., Matrixdrucker, Thermotransferdrucker, Kass.-Laufwerk, 30 St. 3"- u. 150 St. 5.25"-Disketten, alle notwendigen Kabel sowie Joystick f. DM 1.800,–. Schubert, Rüsselsheim, Tel. 06142/62146

VERKAUFE: Schneider CPC 6128 u. Farbmon., 1/2 J. alt, m. Zubehör: u.a. Datenrec., Spiele,Utilities, Resettaste, Disks, Kass., 4 Prog.-Sprachen, Lit. VB DM 1.700.- T. Volontieri, Knoppstr. 13a, 6694 Krexweiler

---

VERKAUFE Orig.-Spiele u. Anwend. f. CPC: Kass. ab DM 2,–; 3"-Disk ab DM 10,– (z.B. Elite, Vermeer etc.). Suche Midway u. andere Strategiespiele sowie Anwend. G. Radons, Kolbenzeil 18, 6900 Heidelberg

---

*** TAUSCHE *** Habe viele Spiele! 3" u. 5.25" Bevorzuge 5.25"! Schickt Eure Listen an: Malte Kroll, Haidweg 17, 2090 Winsen Luhe. 100 % Antwort!

---

17 Orig.-Spiele f. Schneider CPC auf Kass. u.a. Mindshadow, Hacker, Avenger, Ballblazer, Trailblazer, the Eidolon, Highway Encounter, zus. 60,– DM. Tel. 0231/178427

ALLES FÜR DEN CPC: Lightpen DM 30,–; 3 Data-Becker-Bü. DM 50,–; 20 Zeitschr. DM 40,–; 20 Kass. m. Software 40,– DM; 28 Disks m. Softw. DM 120,–; 10 Leerdisketten 3" 45,– DM; 25 Orig. auf K. u. D. DM 200,–. Tel. 06224/10866

SUCHE kOntakte zu CPC-Usern. Infos u. Prog.-Austausch sehr willkommen. Andreas Suwala, Hauptstr. 66, 8740 Bad Neustadt

---

SCHNEIDER-CPC-66-Konsole m. defekter Tastatur DM 50,–. Lit., Kass., Disks 3" f. Schneider CPC wg. Systemwechsel zu verkaufen. Tel. 08157/7255

Schneider CPC 664 Color, Comp.-Tisch, 2. LW, Kass.-Rec., Drucker Star SG 10, 6 DB-Bücher, ca. 122 Schneider-Hefte, viel SW. Neu ca. DM 5.400,–, VB DM 2.500,- Jörg Speckbacher, Westermannweg 3a, 3000 Hannover 21. Bitte nur schriftl. Angebote.

BARCODE-DRUCKPROGRAMM f. normale Matrixdrucker! Läuft auf PC u. CPC. Auch andere Programme im Angebot. Info gg. Rückporto v. Holger Pellmann, Klusmoor 3, 4512 Wallenhorst

---

VERKAUFE: CPC 6128 Color m. F1X DMP 2000 3"-lW, 150 St. 3"- u. ca. 200 St. 5.25 Disks. Sehr viele Anl. u. Bücher. Preis VS. Wilhelm Beese, Baarstr. 27, 7218 Trossingen, Tel. n. 18 Uhr 7425/7777

Gebe ab: CPC 6128, Farbmon. DMP 3000, 5.25"-LW kompl., 3"-LW ohne Stromvers., CPC m. A- u. B-Drive umsch. u. Resettaste, ca. 80 Zeitschr., div. Bücher u. jede Menge sonst. Info-Mat. H.-M. Altenberg, 7218 Trossingen, Tel. 07425/4545

CPC 464/664/6128: Verk. div. Spielesoftware auf Kass. Liste gg. 0,80-DM-Breifm. od. Freiumschlag. Bomb Jack 1 auf Disk. DM 20,–. Klaus Lindner, Weilerstr. 7, 5303 Bornheim 4

VERKAUFE Prog. f. CPC 464. Wer kaufen will, soll ab 18 Uhr anrufen: 069/4970692 Preis VS

VERKAUFE Computer CPC 464 m. Software u. Joystick. Preis VB. Tel. 05657/1035

IHR WOLLT TAUSCHEN? Dann schickt Eure bespielten Disks an: Hans-Dieter Hegedüs, Lindenplatz 15/2, 7417 Pfullingen. Ihr kriegt Sie 100%ig m. gutem Prog. zurück. Nur 3"-Disk.

SUCHE SPEICHERERWEITERUNG 64K od. mehr, auf CPC 664 lauffähig. Chr. Dürschlag, Hauptstr. 9, 7208 Spaichingen

---

CPC 6128: Verk. Small-C-Entwicklungssystem: C-Compiler, 8080/Z80-Macorassembler, Linker/Loader, Bibliotheksverwalter, Editor/Texttools. CPCs 464/664 benötigen Speichererw. (Minimum 56 KByte). Orig.-Handbuch u. 3 Disks DM 90,–, bei NN DM 94,–. Klaus Lindner, Weilerstr. 7, 5303 Bornheim 4

Wer kann mir in Deutschland einen Händler od. eine Bezugsquelle nennen, wo ich f. den CPC 464 das Prog. „Zaxxon" auf Kass. od. Disk bekomme. Biete 30,– DM Belohnung, dem, der mir hilft. Bitte an: Wolfram Schröter, Kurt-Schumacher-Str. 245, 7000 Stuttgart-Fasanenhof

---

CPC 464 grün, SP256, DD1 900,– DM; RS232 u. orig. Trafo 120,– DM; Sprachausgabe 50,– DM; Dataphon 321-23D (Btx-fähig 200,– DM; Prog.: orig. GENO 99,– DM; Dirman 30,– DM; Discovery 30,– DM u. andere div. Spiel- u. Anwendersoftw. Tel. 0212/311210

SUCHE Heft 6787 v. Happy Computer. Zahle bis DM 6,–. Angebote an Christian Herzog, Im Wirtental 6, 7890 WT-Tiengen 2

SUCHE für mich und meinen CPC 6128 m. Drucker Star-NL 10 Heimarbeit. Tel. 06185/690864

VERKAUFE wg. Systemwechsel Computeranlage komplett: CPC 6128, NLQ 401, Maus, umfangr. Software, viel Lit. u. kompl. Zubehör. 30 % unter Neupreis! A. Klemm, Lindenstr. 15, 7032 Sindelfingen, Tel. 07031/86235

VERKAUFE Orig.-CPC-Tapes u. 3"-Disks. Tausche Anwend. od. Spiele auf 3"-Disks. Liste gg. 0,80 DM Rückp., bitte angeben, ob Kauf-, Anwend.- od. Spieleliste bei: Rolf Klein, Finsterheckstr. 22, 6580 Idar-Oberstein 1

PUBLIC-DOMAIN-USER-GRUPPE * CP/M- u. DOS-Programme f. CPC, Joyce u. PC. Habt Ihr selbstgeschriebene Prog., schreibt uns! Wir bauen eine Freeware-Bibliothek auf. Cat-Disk PC (4 St.) 5,– DM; 3" 12,– DM, Info 2mal 80 Pf. Briefm. Rückp. PDUG, PF 1118, 6464 Linsengericht

---

SUCHE DRINGEND Ansteuerdaten f. Textprog. Starwriter 1 V 3.0, um meinen Drukker Star SG 10 anzusteuern. Besonders die Grafikdaten, um Briefköpfe auszudrucken. Hilfe an Peter A. Müller, Eschkopfstr. 1, 6800 Mannheim 1

SUCHE AMX Pagemaker, Mailmerge, Staradress u. billigen Drucker. Bruno Metzinger, Allerheiligenstr. 65, 7590 Achern 12, tagsüber Tel. 07841/3050

SUCHE 3"-Disk-Station möglichst günstig. Zuschriften bitte an: Thore Bunge, Weimarer Str. 8, 2300 Kiel 1

Wordstar m. Mailmerge 3.0 f. DM 90,–; Copyshop DM 30,–; Textomat DM 30,–; Stardatei DM 40,–; Startexter DM 40,–; u. viele andere Orig.-Prog. auf 3"-Disk. Tel. 06157/81550

VERKAUFE CPC 6128 grün, Lehrbücher, Programme, Zeitschriften, 17 Disks. Preis DM 900,–. H.-J. Kirsch, Anhagener Str. 27a, 3050 Wunstorf 1, Tel. 05031/14558

VERKAUFE CPC 6128 grün, CPC-Mousepack 2.0 m. dazugehöriger Software, div. andere Programme, Abdeckhaube, Verlängerungskabel u. Joystick im Wert v. 1.800,- DM f. 600,– DM. Anfragen: Tel. 02324/30649

CPC 464 m. Grünmon., Joystick, Software u. Spielen DM 250,–; dK'tronics 64K-Memory (für CP/M-Plus- u. 6128-Programme) DM 50,-; Multiface II Kopiermodul (kopiert alles!) v. Romantic Robot; DOS-COPY (liest und verarbeitet MS-DOS-Disketten auf Vortex-Laufwerken) DM 50,–; neuwert. (6 Wochen alt) Vortex FD-1 Doppelstation f. CPC 464 m. Garantie. Alles auch einzeln abzugeben. Preise VB. Tel. 06051/73745. PS: Suche Amiga-1000-Tastatur m. Kabel in techn. einwandfreiem Zustand.

BIETE: Eprom-ROM-Board, super ROM-Plus, 14 freie Plätze DM 70,–; Maxam Eprom u. Handbuch DM 70,–. Für alle CPCs. Ralf Ulrich, Daimlerstr. 15, 2850 Bremerhaven

---

VERKAUFE Schneider CPC 6128, Grünmon., NLQ 401 Drucker und sehr viel Software (z.B. Wordstar 3.0, DR Graph, Super Calc 2 m. engl. Handbuch u. vieles mehr Anfragen an Reinhard Schwab, Tel. 09435/2608

SUCHE TAUSCHPARTNER (Disk). Tausche Prog. aller Art. Schickt Eure Listen an: Dirk Frankenhäuser, Hansjakobweg 7, 7562 Gernsbach. 100 % Antwort.

SUCHE TAUSCHPARTNER f. 3"-Disk. Habe Super-Software z.B. World Games, Wintr Games, Know, B 727 Flugsimulator, Jagd auf Roter Oktober u.v.m.! Schreibt an: Heinz Bischoff, Berliner Platz 2, 4030 Ratingen 1, Tel. 02102/490349

Laufw. f. CPC 464 zu verk.: DD! m. Contr. u. 10 Disks 350,– DM VB; 5.25"-Zweitlaufw. u. Direktanschluß u. 20 Disks 300,– DM VB. Info bei: Kretzschmar, Postf. 210405, 4100 Duisburg 1, Tel. 0203/337383 nach 17 Uhr od. 0203/2833116 ab 8 Uhr

---

Gemeinsam mit Kranken

Anwalt für das Leben

caritas

Schneider Experten für Sie

7054 Korb

RICHARD HEINRICH & SOHN

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

7700 Singen

Schneider COMPUTER DIVISION elektronik & service

Ihr Fachhändler
Marianne Tröndle
Hauptstr. 11, 7700 Singen
(Hohentwiel), Tel. 07731/6 44 33

7700 Singen

Ihr kompetenter, autorisierter
SCHNEIDER-Fachhändler
Schellhammer
Hard- u. Software, Peripherie, Literatur
Fachkundige Beratung - kompletter Service
7700 Singen, Freibühlstr. 21-25
Tel. 07731/82020

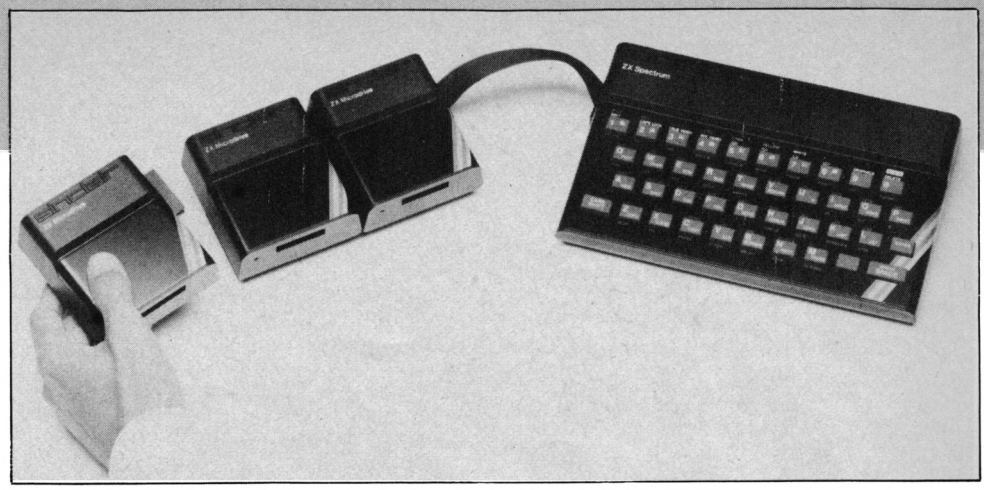
1000 Berlin

Die COMPUTEREI
Wolfgang Hiller
Schneider COMPUTER DIVISION
Hardware
Software
Zubehör
Beratung
Literatur
Tempelhofer Damm 120
1000 Berlin 42
Am U-Bhf. Tempelhof
Tel.: 7 52 20 91

Hier könnte Ihre Anzeige stehen

6090 Rüsselsheim

Computer u. Bürotechnik
Dipl.-Ing. Neuderth
Frankfurter Str. 23/Ecke Friedenspl.
6090 Rüsselsheim
Tel. 06142-68455



Nur die Microdrives bremsten den Erfolg des ZX Spectrum

Fortsetzung von Seite 9

Absatzprobleme bei den klassischen Spielzeugprodukten, die im Kinderzimmer von der Elektronik verdrängt wurden, brachten die Company in finanzielle Schwierigkeiten. Als das Finanzamt zum ersten Mal eine Zahlungsunfähigkeit bescheinigte, mußten die Rechte am Dragon 32 weiterverkauft werden.
Dragon Data erwarb sie. Die Firma war ursprünglich eine Tochtergesellschaft von Mettoy, konnte sich jedoch nach einiger Zeit selbständig machen. Nun war das Kind erwachsen und wollte dem Dragon durch eine Weiterentwicklung zu neuem Leben verhelfen. Aber auch bei Dragon Data reichte das Finanzpolster nicht aus, um die kleine Verkaufsflaute zu überstehen und nebenher noch weiter zu arbeiten. Das bißchen Geld wurde in MSX-Experimente gesteckt. Glücklos, wie man heute weiß.
Dem Dragon 32 wurde keine Systempflege zuteil. Der versprochene Nachfolger Dragon 64 blieb eine Willenserklärung auf dem Papier. Schließlich nahm das bekannte Spiel wieder seinen Lauf: Weiterverkauf der Rechte, nochmaliger Weiterverkauf und so weiter. Zuletzt landeten die Konstruktionspläne in Spanien, wo sogar noch ein 128-KByte-Nachfolger produziert wurde. Aber Spanien war weit; zu weit, um den englischen Computermarkt noch entscheidend zu beeinflussen.

Acorn Proton – Der Fernseh-Computer

Von der britischen Firma Acorn geht das Gerücht, daß sie ein besonders cleveres Management habe. Es heißt sogar, der Firmenname sei nur deswegen so gewählt worden, um vor Apple im Telefonverzeichnis zu stehen. Das muß nicht stimmen, aber wie der Proton unter die Leute gebracht wurde zeugt von Geschäftstüchtigkeit und macht die Telefon-Geschichte glaubwürdig.

An sich handelte es sich beim Proton um das, was man damals als übliche Technik bezeichnen konnte. Niemand regte sich darüber auf. Aber der Rechner war der Behälter eines völlig neuen BASIC. Statt der üblichen Magerversionen wurde hier wirklich so etwas wie eine hochstehende Programmiersprache geboten. Komfortable grafische Befehle, WHILE...WEND-Schleifen und IF...THEN...ELSE-Strukturen ließen strukturiertes Programmieren zu. Außerdem war der Interpreter schnell und absolut fehlerfrei.
Nicht den Meistern des Lötkolbens kam die Ehre für den Proton-Erfolg zu, sondern den Programmierern des ROMs.
Trotzdem verkaufte sich der Homecomputer mehr schlecht als recht. Die Ausschreibung der englischen Fernsehgesellschaft BBC kam den Acorn-Managern gerade im richtigen Augenblick. Die Fernsehanstalt suchte einen Rechner, mit dem ein Schulungskurs für die Zuschauer produziert werden konnte. Da es eine öffentliche Anstalt war, spielte Geld keine Rolle; BBC wollte bezahlen, was verlangt wurde. Entsprechend groß war die Firmenbeteiligung an diesem Wettbewerb. Jeder Hersteller pries seinen Rechner als fleißiges Lieschen und den Interpreter als Wunderwerk der Programmierer. Der Proton aber hatte genau das, was die Fernsehmacher suchten: Ein leistungsfähiges BASIC, mit dem sich alles machen ließ. Ehe sich die BBC die Sache überlegen konnte, hatte sie auch die Rechte am Hals. Acorn machte Nägel mit Köpfen.
Daß Rechte immer mit Pflichten einhergehen, machte dem Fernsehen keine Sorgen. Große Verkaufszahlen waren nicht ihr Problem. Daß der Rechner überteuert war, spielte für die TV-Bosse keine Rolle. Der Teilnehmer am Fernsehkurs mußte ohnehin den BBC Acorn, wie er nun hieß, wählen, wenn er mitmachen wollte. So leicht wurde noch nie ein Computer vertrieben. Die Öffentlichkeitsarbeit war größer als es sich Acorn jemals hätte erlauben können. Die roten Zahlen, die trotzdem geschrieben wurden, blieben dem Hersteller auch erspart, er hatte bereits an den Rechten satt verdient.

Adam Coleco – Amerikaner ertrank im Großen Teich

Es gibt Leute, die behaupten, Adam hätte es nie gegeben. Dabei muß es sich nicht etwa um Ungläubige Heiden handeln, sondern um Computer-Freaks, die vergeblich auf einen Rechner aus den USA warteten. Immerhin gab es dort einige Testgeräte, wohl nicht mehr aus tausend Stück, und niemand wußte, wie wenige Anwender zu der Ehre kamen, ein solches Gerät zu besitzen. Der Wirbel, der um den Adam gemacht wurde, stand freilich in keinem Verhältnis zu seiner tatsächlichen Verbreitung. Die Presse überschlug sich und es gingen weit über hunderttausend Bestellungen ein, noch bevor irgendjemand das Modell im Laden sah. Den Wirbel veranstaltete wieder einmal ein Spielzeughersteller: Coleco.

Die Werbesprüche klangen diesmal aber auch besonders gut. Mit 80 KByte Ram sollte Adam antreten und so ganz nebenbei noch ein eingebautes Textprogramm aufweisen. Verständlich, daß dazu eine vernünftige Tastatur angeboten werden mußte. Den Fehler, ein Gerät ausschließlich für ernsthafte Anwender zu bauen, wiederholte Coleco nicht. Das Angebot an die Spielefreaks waren die beiden Zehnertastaturen – hier allerdings nur als Folie – mit eingebautem Joystick. Einer davon konnte an die rechte Seite montiert werden und ließ das Gerät erscheinen, als stamme es aus einem Raumschiff. Das Design stimmte und kam gut bei den erwartungsvollen Kunden an.
Bei der Grundkonfiguration benahm sich der Hersteller wie ein Weihnachtsmann bei braven Kindern: Er verschenkte die Peripherie geradezu. Da gab es einen Datenrekorder mit neuer Technik. Er arbeitete zwar mit herkömmlichen Kassetten, konnte aber schneller und sicherer auf die Daten zugreifen. Da der Adam als Textsystem konzipiert war, gab es auch einen Typenraddrucker, als wäre das nichts Besonderes. Und damit jeder auch gleich damit spielen konnte, wurden auch noch zwei Spiele mitgeliefert. Der Preis für die ganze Bescherung hätte zweitausend Mark nicht überschritten. Kein Wunder, daß es im Blätterwald der Fachpresse nur so rauschte.
Allein, ein Durchbruch sollte es nicht sein. Obwohl bereits im Herbst 1984 der deutsche Vertrieb bekannt war, schipperten nur zwei oder drei Prototypen für den Fotografen über den Atlantik. Kaufen konnte das Gerät niemand, weil Coleco gar nicht erst mit einer Produktion anfing. Der Spielwarenhersteller mit dem Schwerpunkt Automatenspiele engagierte sich in anderen Produkten. Einfach so.
Adam, so genannt, weil er der erste von vielen Coleco-Computern werden sollte, machte seinem Namen keine Ehre. Er war auch der letzte. Vielleicht, weil ihm die „Eva" fehlte?
Die Liste der Flops ist in Wirklichkeit länger. So zahlreich und gleichzeitig kurzlebig waren die Neuvorstellungen in den ersten Homecomputer-Jahren. Aber nicht nur die kleinen Firmen leisteten sich Fehltritte. Auch große Firmen, etwa IBM mit dem PC Junior, hatten ihren Mißerfolg. Die eingeführten Hersteller konnten sich derlei Fehlinvestitionen allerdings erlauben; andere Unternehmen gingen daran zugrunde. Aber in der Schlacht um den Kunden gab es nicht nur Verluste. Viele Rechner erzielten einen Achtungserfolg und konnten sich dank guter Systempflege behaupten. Wiederum andere Hersteller besetzten eine Marktnische und waren mit wenig Gewinn und der Tatsache, daß die Konkurrenz sie in Ruhe ließ, sehr zufrieden. Hauptsache, man schrieb keine roten Zahlen.
Einige Rechner wurden damals zu Bestsellern. Heute ist ihr Stern im Sinken. Schließlich hält der Verkaufserfolg nicht alleine dann an, weil einige Hunderttausend Stück eines Gerätes verkauft sind.
In der jungen Computergeschichte folgt also die Abteilung jener Computer, die auch heute noch sehr beliebt sind, deren Besitzer freilich schon mit einem neuen Modell liebäugeln.



Der erste „Volks-68000er" stammte aus Sir Clive Sinclairs Werkstatt

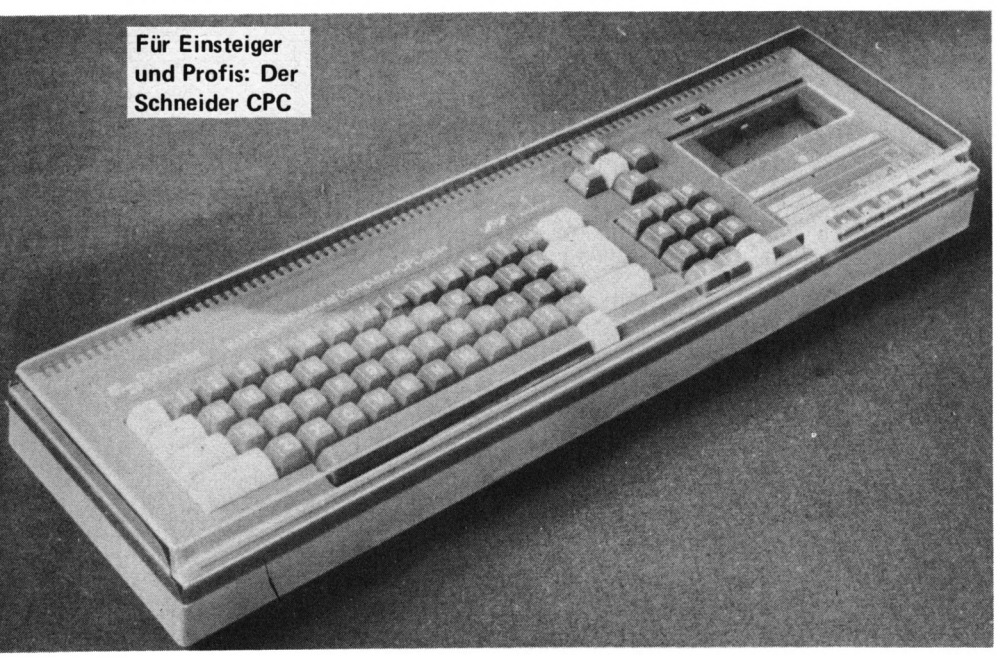
Sharp MZ 800 – Der halbe Schritt zum PC

Die Sharp-MZ-Serie, angeführt vom MZ 800, ist sicherlich einer der Marktnischen-Computer. In den ersten Werbungen sprach der japanische Multi von einem Clear-System, womit die Homecomputer-Kunden zuerst einmal nichts anfangen konnten. Egal, wichtig war nur, daß die 64 KByte Arbeitsspeicher nicht wieder zum Teil von einem BASIC-Interpreter beansprucht wurden. Wer Pascal bevorzugte, der sollte sich seinen Interpreter in einen freien Speicher einlesen können. Und wer bei seinen Programmen mit einem kleinen Runtime-Interpreter auskam, der hatte um so mehr Platz für seine Daten. Bei jedem PC, den sich Abiturienten und Studenten aber nicht leisten konnten, funktionierte das System prächtig. Genau dies meinte Sharp, als die Werbung den Clear-Computer anpries.
Der Sharp gab sich professionell. Die Tasten hatten ein angenehmes Anschlaggefühl und waren ordentlich in Funktions- und Cursorblock geordnet. Es gab sogar einen eingebauten Datenrekorder, der den MZ 800 unabhängig von jedem, damals noch recht teuren, Laufwerk machte. Dadurch verlor er etwas von seinem professio-

nellen Touch, aber der Preis blieb erschwinglich. Für zwölfhundert Mark war der Kunde dabei. Das Gerät wurde schnell zum Geheimtip unter Informatikstudenten, die ja selten etwas mit BASIC anzufangen wußten, aber dennoch einen Rechner mit großem Speichervolumen haben wollten. In dieser Marktnische behauptete sich der Computer und bekam sogar noch Unterstützung durch einen programmierbaren Taschenrechner, der in jedem Hörsaal zu finden war. Sharp war – und ist es noch – **die** Marke in den Universitäten.
Wie lange der MZ 800 noch im Gespräch bleiben wird, weiß niemand. Zwar ist das Bessere des Guten Feind – und es gibt heute gewiß Besseres – doch gerade bei diesem Benutzerkreis gilt die Markentreue noch etwas. Sharp bleibt Sharp, auch wenn man den MZ 800 nur noch auf dem Flohmarkt erwerben kann. Der Hersteller ist inzwischen ja auch auf dem PC-Sektor aktiv. Und da ist es schön, wenn er auf seine Stammkunden zurückgreifen kann.

TI 99/4A – Ungeliebter Bestseller

Mit neuem, von der amerikanischen Firma Texas Instruments selbstentwickelten Prozessor trat der TI/99A im Jahr 1979 seinen Siegeszug an. Damit hatte sich erstmals ein PC-Hersteller im Homecomputerbereich engagiert. Die Leistungen, die der „Kleine" aufzeigte, machten sich besonders bei der Anwendung mit Maschinensprache bemerkbar. Damit zu programmieren, machte mit dem TI Spaß.
Nachteilig waren manche Besonderheiten der Hardware, etwa Expansion-Ports, die Texas Instruments aber durch konsequente Systempflege wettmachte. Auch die Preissenkungen der elektronischen Einzelteile wurden an die Kunden weitergegeben, so daß eigentlich alle zufrieden sein konnten.
Einer war es nicht: Hersteller Texas Instruments. Der Homecomputer verursachte zuviel Aufwand und brachte nicht das vergleichbare Image eines PC ein, den man ja – ganz nebenbei – auch herstellte. Der Wettkampf auf diesem Sektor gegen IBM kostete alle Kraft. Die Marktführung im Homecomputer-Bereich nutzte werblich wenig, kostete nur Kapazitäten. Also beschlossen die Manager, die Produktion einzustellen und die Lagerbestände in einer Großaktion zu verramschen. Jetzt erst zeigte sich, wie gut der Vertrieb funktionierte. Kleine Firmen übernahmen die Versorgung mit Peripheriegeräten, und Userclubs hielten den TI am Leben. Der Rechner konnte einfach nicht sang- und klanglos vom Markt verschwinden; er galt bei den Besitzern – die ohnehin als fanatisch berüchtigt waren – als der beste Homecomputer aller Zeiten. Der TI 99/4A ist deshalb nicht tot. Er wird nur seit 1983 nicht mehr hergestellt – wem macht dies schon etwas aus?



Für Einsteiger und Profis: Der Schneider CPC

MSX – Japans mißglückter Angriff

Die vielen Homecomputer, die um die Gunst des Käufers buhlten, hatten nichts, kein Detail, gemeinsam. Keiner war kompatibel zu irgendeinem anderen, jeder Hersteller benutzte ein eigenes Betriebssystem und ein eigenes BASIC. Softwareverlage mußten eine gute Nase haben, um zu erkennen, welche Maschine so gut verkaufen würde, daß sich das Erstellen von speziellen Programmen lohnte. Dementsprechend hoch waren die Preise für Programme oder Peripheriegeräte. In diesem unübersichtlichen Markt konnte ja noch keine Massenfabrikation bei der Software anlaufen. Was lag näher, als endlich einen Standard zu schaffen, der mit diesem Durcheinander Schluß machte? Die Japaner griffen die Idee begeistert auf und formierten sich zum Angriff auf den Homecomputermarkt der USA und Europas. Mehrere Elektronik-Hersteller schlossen sich zusammen und wählten den amerikanischen Systementwickler Microsoft – daher das MS bei MSX – zum „Guru". Was bei der gemeinsamen Arbeit herauskam, lehrte die Konkurrenz das Fürchten, nicht nur wegen der zu erwartenden Handbücher. Jeder hatte ja noch das Beispiel der Video-Industrie vor Augen, bei der eine Standardisierung weltweit gelungen war. Betriebssysteme, Schnittstellen



Sein Markterfolg ist unbestritten: Commodores C64

und BASIC-Interpreter wurden vereinheitlicht, so daß jeder Hardwareteil und jedes Programm auf einem MSX-Rechner lief, egal welcher Hersteller den Aufkleber an der Tastatur anbrachte. Ziel war es, daß der Besitzer eines MSX-Gerätes sich schnell die Software im Kiosk nebenan kaufen konnte, ohne dabei nach der Kompatibilität zu fragen.
Der erwartete Erfolg stellte sich trotzdem nicht ein. MSX geriet fast zum japanischen Heimspiel, denn in Europa zog nur Philips richtig mit. Die Engländer aber fragte wieder einmal keiner. Und ausgerechnet die „machten" die Homecomputer und bestimmten, wo es lang ging. Jetzt waren sie auch noch beleidigt, weil sich jemand einmischte.
Als Verkaufsbremse für MSX erwiesen sich die hohen Einführungspreise. Es war schon paradox, daß Geräte, die das EDV-Hobby durch Massenherstellung billig machen sollten, teurer waren, als jene, die sich den Luxus eines eigenen Systemes erlaubten und nur in kleinen Stückzahlen gefertigt wurden. Während die Computer-Magazine noch von der großen MSX-Zukunft sprachen, war für die User der Fall schon klar. MSX war nicht besser, sondern nur so ähnlich wie das, was sie schon immer hatten. Also blieben sie ihrer Marke treu. Die Rechner wurden fast ausschließlich an Einsteiger verkauft, selten an erfahrene User, die auf einen anderen Computer umsteigen wollten.
Daß ausgerechnet die Japaner den Homecomputermarkt falsch einschätzten, ist eigentlich unverständlich. Die Manager des fernen Ostens übersahen dabei eine wichtige Tatsache: Die Freaks waren gar nicht so sehr an einem Standard interessiert, sondern schätzten die herausragenden Eigenschaften ihrer eigenen Geräte und nahmen kleine Nachteile in Kauf.
Heute gilt MSX nur als ein Computersystem unter vielen. Die Verkaufszahlen beweisen, daß kein verbindlicher Standard gesetzt werden konnte.

Spectrum – Ein Mirakel

Eigentlich dürfte es ihn gar nicht mehr geben. Dabei steht ihm vielleicht sogar eine große Zukunft bevor. Er ist eigentlich der legitime Nachfolger des ZX 81.
Sir Clive Sinclair erkannte rechtzeitig, daß sein Geniestreich nicht ewig ein Verkaufsschlager bleiben würde. Schon längst beherrschten leistungsfähigere Homecomputer die Szene. Also entwickelte er seinen Nachfolger in dieser Szene, den Spectrum.
Endlich gab es Farbe auf dem Bildschirm, ein wirklich gutes BASIC und dazu 48 KByte Arbeitsspeicher. Der Spectrum entpuppte sich als kleines Kerlchen mit ansprechender Leistung, die man ihm auf den ersten Blick gar nicht ansah. Die Gummi-Tastatur mußte der Käufer ertragen. Aus heutiger Sicht waren schlechte Tastaturen das Markenzeichen von Sir Clive Sinclair. Ihm war das Keyboard einfach nebensächlich.
Zumindest die Engländer akzeptierten den Spleen des großen Meisters und kaufen seine Produkte. Der Erfolg stellte sich schnell ein.
Der Computer, der so ganz nebenbei auf den Markt kam, entwickelte sich wider alle Prognosen zum Hit und verkaufte sich ohne große Werbung. Aber Geldverdienen alleine reichte Clive Sinclair nicht. Das Kleingeld wurde in den QL investiert, mit dem die Firma schließlich in die roten Zahlen kam. Sinclair war pleite und mußte an Amstrad verkaufen. Die hatte aber mit dem CPC schon einen Homecomputer im Rennen. Aus für den Spectrum?
Nicht doch. Der neue Herr, Alan Sugar, wußte, was Geld brachte und vertrieb den „schwarzen Zwerg" weiterhin. Im Laufe der Zeit verpaßte man ihm ein neues Design, eine bessere Tastatur, einen eingebauten Datenrekorder, mehr Speicherplatz und schließlich sogar ein Diskettenlaufwerk. Eben, was so nebenbei anfiel, wenn man sich mit der Entwicklung anderer Geräte befaßte.
Es scheint, als seien die Spectrum-Zeiten noch längst nicht vorüber. In Großbritannien gilt er als der Spielcomputer schlechthin und ersetzt die üblichen Spielkonsolen. Jedes Jahr um die Weihnachtszeit füllen sich die Regale der Kaufhäuser, als sei der Weihnachtsmann für die Produktion dieses Rechners verantwortlich.
Amstrad hat sich mittlerweile mit eigenem Vertrieb in Deutschland etabliert. Hier soll sich das Erfolgs-Mirakel wiederholen. Man darf gespannt sein, welcher Rechner künftig zum Einstiegscomputer wird: der CPC oder der Spectrum.

Sinclair QL – Superrechner mit vielen Abstrichen

Auch der QL, abgeleitet von Quantensprung (Quantum Leap), war ein Geisteskind von Sir Clive Sinclair. Bedauerlicherweise trieb



Der Atari ST leitete den Trend zum Profi-Computer ein

er damit seine Firma in die große Pleite. Dabei hatte alles so gut angefangen. Nur die Firma Apple konnte einen Rechner vorweisen, der auf der Basis des Motorola 68000 funktionierte. Und nur Apple – der Mercedes unter den Computern – konnte vergleichbare Verkaufs-Preise festschreiben. Der QL sollte ein „Volks-68000-er" werden, den sich jeder leisten konnte. Trotz mitgelieferter Software sollte der Preis dieses Wunderrechners unter zweitausend Mark liegen und dies einschließlich der zwei Speichermedien, die mitgeliefert wurden.
Gerade jene Microdrives, wie sie Sinclair nannte, waren ein Schwachpunkt. Er hätte auf sie besser verzichten sollen. Sie basierten auf Endlos-Kassetten, denen weder die Fachpresse, noch der angesprochene Käufer eine hohe Lebenserwartung zutraute. Mittlerweile mag das Gegenteil bewiesen sein, aber das Speichersystem bleibt trotzdem auf Sinclair abgestimmt und die Preise liegen wegen fehlender Massenproduktion sehr hoch.
Nächster Schwachpunkt war die Tastatur – eben wieder einmal typisch für Clive Sinclair. Er vermochte einfach nicht einzusehen, warum ein professioneller Computer auch eine professionelle Tastatur haben sollte. Wer tippen will, der muß auch leiden, war wohl seine Devise. Die Tasten lagen zu eng aneinander, waren chaotisch angeordnet und das Keyboard insgesamt war zu hoch. Ein schwarzer Balken auf dem Schreibtisch, der aussah, als hätte man zwei Briketts nebeneinander gelegt. Als sich bei der Premiere – ausgerechnet in Cambridge, dem Silicon Valley der Engländer – das ROM auch noch als zu klein und sparsam erwies, war der Flop perfekt.
Mittlerweile sind die größten Fehler behoben. Es gibt ein Laufwerk, welches die Microdrives ersetzt, das ROM ist größer, die Bugs sind beseitigt und die Software wurde verbessert. Zu guter Letzt rutschte der Preis sogar unter vierhundert Mark. Die wenigen Besitzer solcher Geräte schwärmen vom QL und möchten ihn nicht mehr hergeben, für den Massenverkauf kamen die verkaufsfördernden Maßnahmen aber zu spät.
Die nachhaltige Begeisterung der Benutzer am „Volks-68000er" wird von Amstrad, die ja das Sinclair-Komplett-Paket gekauft haben, nicht wahrgenommen. Zumindest zur Zeit noch nicht. Der QL verstaubt in englischen Lagerhäusern und wartet auf eine zweite Chance.
Es gibt bemerkenswerte Rechner, die sich selbst noch gegen die heutigen Supermaschinen behaupten. Sie wurden so oft verkauft, daß die Mundpropaganda die Werbung ersetzte, die die Firmen heute schon nicht mehr investieren. Das Software-Angebot ist so groß, daß Einsteiger geradezu dazu gezwungen werden, zu einem bestimmten Rechner zu greifen. Uneingeschränkt treffen all diese Faktoren bislang allerdings nur für den Commodore C64 zu. Der CPC von Amstrad zählt zu den ewigen Zweiten. Aber es könnte mal ein Bestseller aus ihm werden, wenn die Firma ihn lange genug auf dem Markt hält.
Der CPC kam 1984 auf den Markt und wurde auf Anhieb ein Verkaufserfolg. Das Komplettsystem mit Speichermedium und Monitor verkaufte sich, wenn man die drei angebotenen und verkauften Varianten zusammenzählt, über dreihunderttausend Mal in Deutschland. Die Pluspunkte sind schnell aufgezählt und erklären diesen Erfolg: ein gutes BASIC, CP/M-Fähigkeit und, damit verbunden, die 80-Zeichen-Darstellung. Der Niedrigpreis sprach für sich selbst.

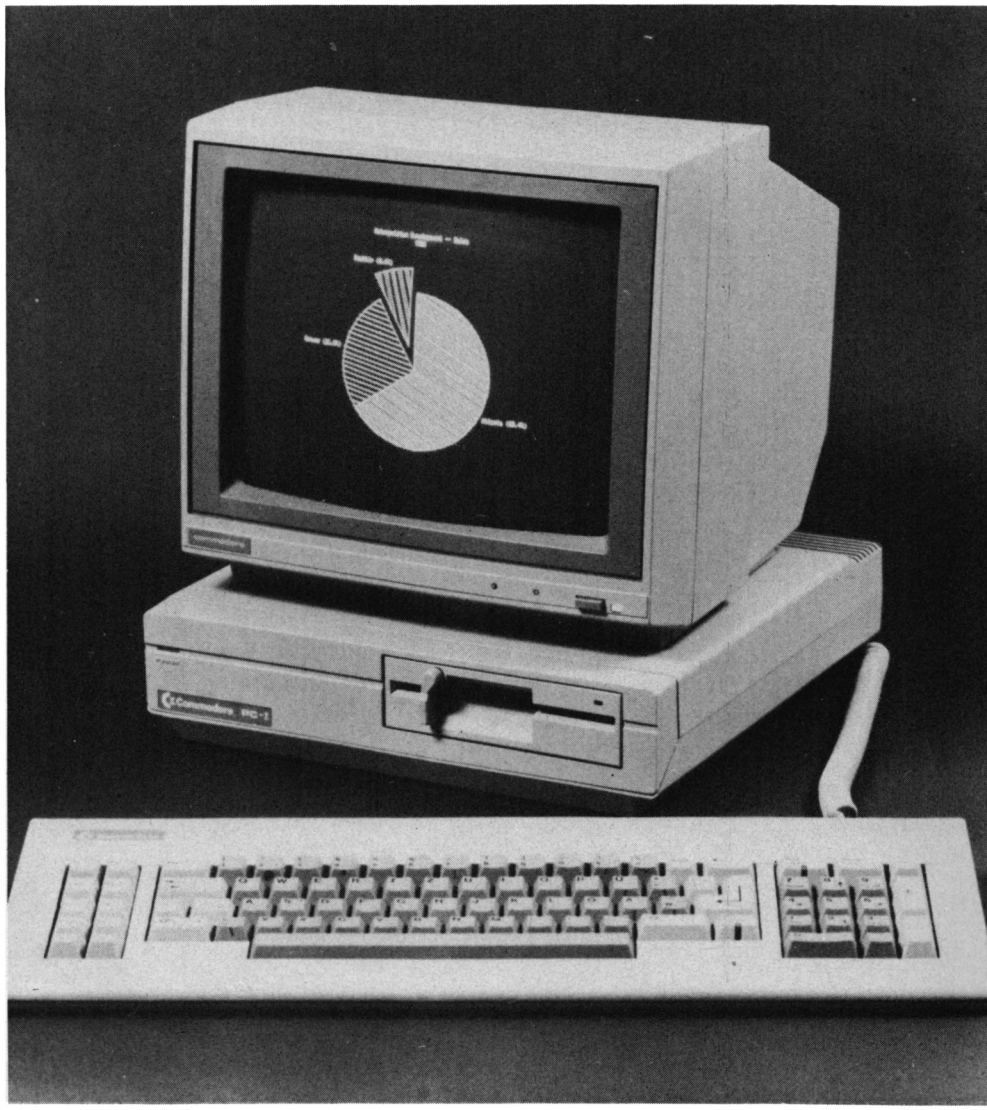
Schneider/Amstrad CPC – Ein Streitfall

Der stärkste, nein, der einzige Konkurrent war der Commodore C64. Und dessen Vorsprung war auf allen Gebieten groß. Die vielen Geräte, die bereits verkauft waren, hatten den Softwaremarkt erobert und sorgten für immer bessere Neuheiten. Dagegen galt es anzukämpfen.
Mit dem CPC hätte vielleicht einiges getan werden können, aber die Vorstellungen, die man bei Amstrad und Schneider von einer Produktpflege hatte, waren – gelinde gesagt – eigenartig. Statt einen Rechner mit Zubehör zu vervollkommnen, brachte man lieber die nächste Variante heraus. Und dann wieder eine. In England verließ sich Amstrad voll auf Drittthersteller, die aber nicht immer Lust hatten, das Spielchen mitzumachen.
Als Firmenchef Alan Sugar nichts mehr einfiel, verließ er die Produktreihe und ließ den Joyce bau-

en. Software- und Hardware-Hersteller waren irritiert und brachten ihre Produkte nur zögernd auf den Markt. Vor allem immer erst als Nachlieferung zu den neuen Versionen für den C64. Der CPC geriet zum ewigen Zweiten. Aber auch auf diese Weise läßt sich gut leben, wie der Erfolg von Amstrad zeigt. Die oben erwähnte Verkaufsstatistik gilt ja nur für Deutschland. Insgesamt kommt mit England und Frankreich ein erkleckliches Sümmchen zusammen. Die Herstellung des CPC wird deshalb noch lange nicht eingestellt, wie es auch die neue Vertriebspolitik der Firma Amstrad offenlegt. Der Rechner wird noch lange Zeit in den Geschäften zu finden sein. Und wenn es ans Sterben geht, ist es gegenüber dem C64 vielleicht sogar ganz gut, der Zweite zu sein.

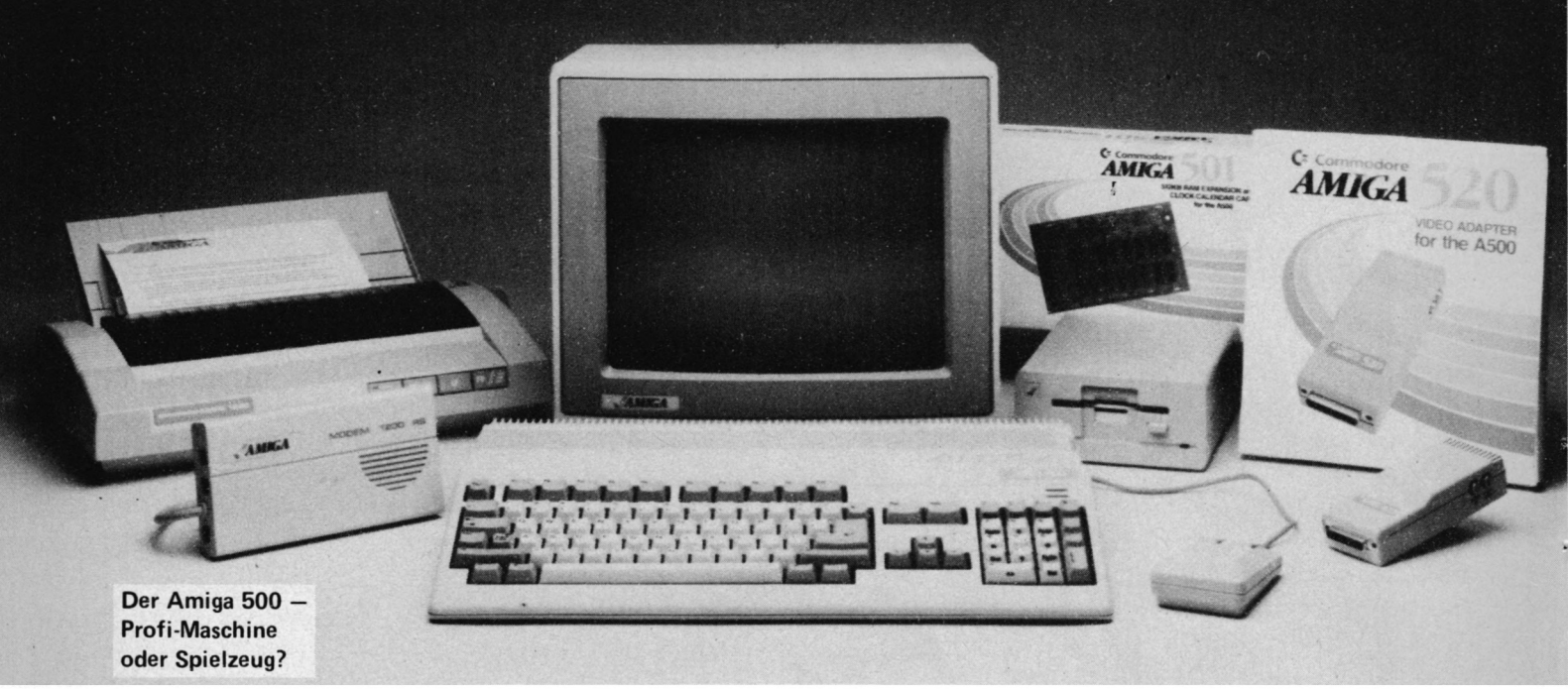
Commodore C64 – Rechner mit Eigendynamik

Es ist schon unglaublich, was diese „Kiste" so alles schafft. Der C64 war der erste Homecomputer, der über 1,5 Millionen Mal verkauft wurde. Dem „Kopfkissen mit Tastatur" wird sogar nachgesagt, daß es seinen Hersteller vor der Pleite rettete. Selbst, als ihm seine geistigen Eltern mit dem Plus/4 den Garaus machen wollten, überlebte der C64 den versuchten Kindesmord. Gerade, als man dabei war, ihn auszuzählen, stand er bei Neun wieder auf, hatte sich durch das neue GEOS-Betriebssystem erholt und kämpfte so gut wie eh und je um seine Marktposition. Dem Plus/4 ging daraufhin die Luft aus, er rechnete wohl mit einer schnellen Niederlage seines großen Bruders.
Mittlerweile wiederholt sich das Spielchen. Der Amiga wurde zwar von seinen Schöpfern als C64-Nachfolger auserkoren, aber das Publikum hat bei diesem ungleichen Kampf den Oldie schon längst in den Ring zurückgerufen. Der Champion hat ein neues Platinen-Layout – eigentlich nur, weil es billiger ist – und präsentiert sich in alter Frische, umgeben von unzähliger Software. Der Amiga 500 muß aufpassen, daß er nicht auf der Strecke bleibt. Zum Glück hat er mit seinem großen Bruder, dem Amiga 2000, einen guten Helfer im Kampf der Konkurrenten.



Die MS-DOS-Rechner sollen den Markt der privaten Anwender erobern

Woran es liegt, daß der C64 seit 1982 unangefochten das Feld der Homecomputer anführt und sich gegen jede Konkurrenz, sogar die aus dem eigenen Haus, behaupten kann?
Mit ihm hat Chuck Peddle eine Lawine in Bewegung gesetzt, die sich nicht mehr aufhalten läßt. Die Markentreue der ersten Käufer sorgte dafür, daß andere Kunden folgten. Die Schüler kauften sich logischerweise jenen Computer, den ihre Mitschüler hatten, um sich die notwendige Software nicht kaufen zu müssen. Gleichzeitig produzierten Softwareverlage für den Rechner, der die größten Absatzzahlen zu melden hatte und damit den größten Verkauf auch bei der Software garantierte.
Händler stellten sich jenen Rechner in die Regale, der sich fast von alleine verkaufte. Es war eine Kette ohne Ende, die noch nicht einmal von Commodores merkwürdiger Preispolitik unterbrochen werden konnte.
Der C64 ist in der kurzen, erst acht Jahre jungen Homecomputer-Geschichte der eigentliche Sieger. Es scheint, als müsse man der Sa-



Der Amiga 500 – Profi-Maschine oder Spielzeug?

che ein gewaltsames Ende setzen, damit es keine Never-Ending-Story gibt. Aber warum, um alles in der Welt, sollte jemand so etwas tun?

Neuer Standard

Commodores C64 war der letzte Homecomputer im klassischen Sinne, der letzte der Acht-Bit-Krieger. Viele Spiele, etwas BASIC und ein paar Anwendungen, damit es nur eben zum täglichen Gebrauch reicht, dies waren damals die Anforderungen, die man an jeden Neuling auf dem Computermarkt stellte. Im wesentlichen gilt dies heute noch, aber es müssen bessere Spiele, mehr Programmiersprachen und professionellere Anwendungen sein, sonst ist heute kein Käufer mehr zufriedenzustellen.
Dem Wunsch sind einige Hersteller nachgekommen, und heute ist es soweit: Niedrige Herstellungskosten und billige Speicherbausteine auf der einen, die Bereitschaft, auch mal mehr Geld auszugeben, auf der anderen Seite: Beide führten dazu, daß sich die 68000er-Prozessoren durchsetzten.
Dabei schämen sich manche Hersteller ihrer Homecomputer-Vergangenheit und sprechen lieber von einem Rechner für den **privaten Anwender** – als sei damit etwas anderes gemeint.
Was der C64 als Computer ist, ist Jack Tramiel in persona: ein unverwüstlicher Champion, der immer wieder auf die Beine kommt. Kaum wurde er bei Commodore „gegangen", machte sich der Manager seine eigenen Gedanken über die Zukunft des Homecomputermarktes. Der Atari ST ist seine Geschichte: die eines Mannes, der mit Dingen Erfolg hat und Geld verdient, bei denen sich andere das Genick brechen würden. Einstmals gründete er Commodore, verließ mehr oder weniger freiwillig die Firma, kaufte Atari, ohne dafür einen Pfennig zu zahlen und ist heute Chef einer der innovativsten und finanziell stärksten Hersteller-Firmen im Computermarkt. Der Erfolg kam mit dem ST.

Atari ST – Was lange währte, wurde endlich gut

Als er 1984 einen Computer ankündigte, der trotz 68000er-Prozessors, Laufwerks, Desktop-Systems, trotz Maus und Monitor weniger als dreitausend Mark kosten sollte, wußten die Fachleute nicht, ob sie weinen oder lachen sollten. Machte sich Jack Tramiel lächerlich, oder wollte er sich mit aller Gewalt ruinieren?
Weder das eine, noch das andere traf zu, wie sich spätestens 1985 zeigte, als der Atari 520 ST auf den Markt kam. Aus den Gratisbeigaben GEM-Draw und GEM-Write wurde zwar nichts, auch konnte der ursprünglich angepeilte Liefertermin nicht eingehalten werden, aber sonst bewahrheiteten sich Tramiels Vorgaben.
Der ST begann sein wildes Leben und hatte von Anfang an die Freaks auf seiner Seite. Das Charisma von Jack Tramiel sorgte dafür, daß die kleinen Fehler – kein ROM, miserables BASIC – erst einmal hingenommen wurde.
Statt dessen folgte Modell auf Modell und Atari startete die Inflation der KByte. Ein MByte RAM war schon bald nichts Ungewöhnliches mehr. Die Preise purzelten, während die Rechner immer leistungsfähiger wurden.
Was Sir Clive Sinclair erfolglos versuchte, war Atari gelungen: Der „Volks-68000er" zum Spottpreis stand im Laden und kurz darauf auf den Schreibtischen. Der Umsatz stimmte. Und daran wird sich wohl auch so schnell nichts ändern.

Amiga – Was länger, währt wird besser

1985 wurde ein neues Gerät in den USA vorgestellt, im Frühjahr 1986 fand es seinen Weg nach Deutschland: der Amiga 1000. Commodores Vorschlag für den „Volks-68000er", dem die Zukunft gehören soll, ist keine Eigenentwicklung. Man kaufte gleich die ganze Firma, die den Wunderrechner entworfen hatte. Die Erwartungen, die deutsche User dem Computer entgegenbrachten, waren sehr hoch und wurden in zweierlei Hinsicht ent-

täuscht. Erstens hatte der Rechner nur 256 KByte Arbeitsspeicher, was sich bei den enormen Speicherbedarf für Grafikbildschirme als lächerlich gering erwies. Zweitens – und dies war unangenehmer – lagen die fünftausend Mark weit über der Schmerzgrenze der angepeilten Käuferschicht: den privaten Anwendern. Commodores Preispolitik entbehrte schon wieder einmal jeder Logik. Gerade zur Einführung eines neuen Rechners wurde der Preis in die Höhe geschraubt. Jack Tramiel konnte sich freuen, weil sich seine Konkurrenz im Preiswettbewerb selbst disqualifzierte.
Dabei hatte der Luxusrechner einiges zu bieten: Er wurde mit Farbmonitor ausgeliefert, hatte Custom-Chips, die den Prozessor entlasteten und für guten Sound sowie bunte, hochauflösende Grafik sorgten. Er konnte mehrere Programme gleichzeitig abarbeiten und für die Büroarbeit gab es schon einen Hardware-MS-DOS-Emulator namens Sidecar. Dieser stammte aus Braunschweig und galt als Bonbon für alle, die nicht vom IBM-Standard lassen konnten.
Wegen des schleppenden Verkaufes besann man sich bei Commodore auf eine Preissenkung. Kleinvieh macht schließlich auch Mist, so lautete plötzlich die Marketing-Strategie. Sobald die Speichererweiterung im Handel war, stand dem Erfolg auch nichts mehr im Weg. Die ersten Käufer stellten sich ein, die Softwarewelle rollte an und rief die nächste hervor. Den letzten Kick zum Bestseller erhielt der Amiga schließlich mit dem 500er-Modell, welches den Home-Bereich abdecken soll. Der Amiga 500 konnte wesentlich billiger als sein Vorgänger angeboten werden, ließ dem Käufer die freie Wahl des Monitors und war kompatibel zu seinem Vorgänger. Hinzu kam der Amiga 2000, der als offenes System alle Möglichkeiten aufzeigt, die dem Rechner noch bevorstehen.

War es das schon, ist die Zukunft des Homecomputers mit dem Motorola 68000 festgeschrieben? Sicher nicht, die ersten Alternativen sind schon erhältlich und sie übertreffen – zumindest bei der letzten Neuheit – wieder einmal alles, was schon da war.

MS-DOS – Der Profi kommt

Der MS-DOS-Rechner war bislang kein Homecomputer und er ließ sich auch nicht in der Rubrik „für den privaten Anwender" einordnen. Die Standardkonfigurationen, die man sich leisten konnte, waren nicht grafikfähig genug und das Spektrum der Software war nicht breit genug gefächert. Die Erweiterungen, wie sie etwa für eine bessere Grafik notwendig waren, kosteten viel Geld.
Der Preisrutsch in der Elektronik-Industrie wirkte sich aber auch auf diese Geräte aus. Die ersten Home-PCs sind da, Amstrad/Schneider, Atari und Commodore machten den Anfang. Trotz niedriger Preise sind die neuen PCs oft besser ausgestattet als die Urväter von IBM. Eine leistungsfähige Grafikkarte ist auf der Hauptplatine untergebracht, was schließlich auch die Softwareverlage dazu brachte, ein paar Spielprogramme anzubieten. Aus der Sicht der Hersteller sind damit die größten Probleme beseitigt.
Größtes Argument beim Verkauf des PC ist jedoch der leichte Umstieg auf die professionellen Rechner, die den Kunden im Berufsleben erwarten. Da es noch lange dauert, bis sich in den Büros ein anderes System durchsetzt, wird der MS-DOS-Rechner noch einige Zeit zu den Homecomputern gehören.

RISC – Schneller, bunter, leistungsfähiger

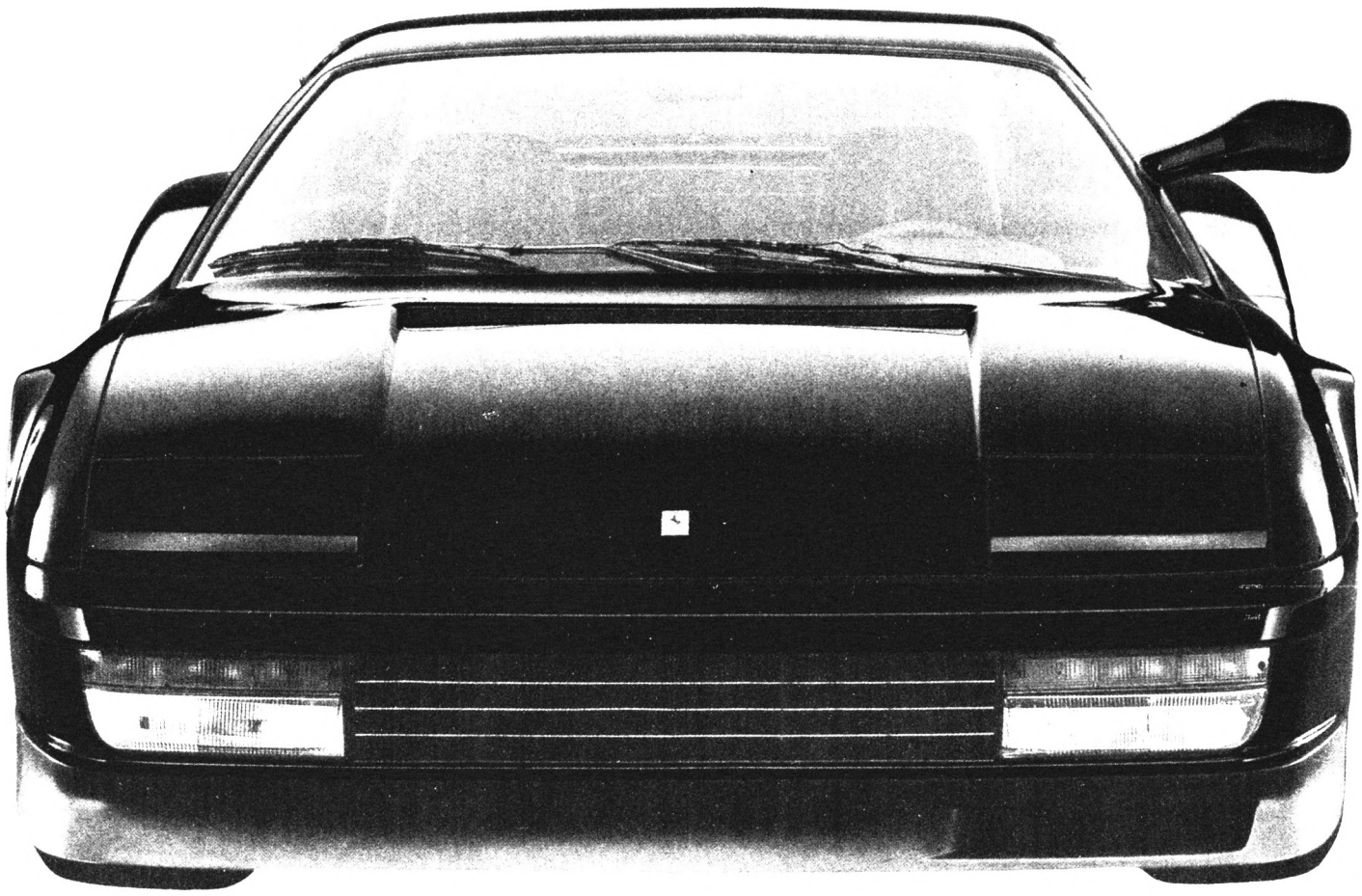
Es ist noch zu früh, um etwas Konkretes über RISC zu sagen. In England ist seit Anfang dieses Jahres der Acorn Archimedes erfolgreich, der erste RISC-Computer, den sich ein Privatmann leisten kann. RISC bedeutet **R**educed **I**nstruction **S**et **C**omputer, ein Computer, der mit einem stark eingeschränktem Befehlssatz auskommt. Dies führt, vereinfacht gesagt, dazu, daß weniger Befehle schneller im Prozessor abgearbeitet werden können. 32-Bit-Register sind dazu die logische Voraussetzung.
Ob sich RISC-Rechner im Homebereich durchsetzen konnen, bleibt abzuwarten. Wie immer entscheidet der Preis. Aber soweit es den Archimedes angeht, ist mit etwa dreitausend Mark das Angebot überzeugend. Die erste Software ist auch schon da. Wenn alle Faktoren zusammenpassen, also Preis, Leistung, Software- und Peripherieangebot im Sinne des Käufers sind, dann ist es unerheblich, ob solch ein Superrechner wirklich zu Hause benötigt wird. Leistungsfähigkeit überzeugt immer, auch wenn sie nicht ganz genutzt werden kann. Schließlich muß ein Porsche auch nicht immer mit Höchstgeschwindigkeit ausgefahren werden.

Zwischenbilanz

Haben Sie Ihren Rechner in der Aufzählung vermißt? Hoffentlich nicht bei den Flops. Aber bei der Fülle des Angebotes lassen sich gar nicht alle Marken erwähnen, die Computer-Geschichte gemacht haben. Es fehlt nicht nur der IBM Junior, sondern auch der Mephisto, der Alphatronic von Triumph Adler und viele andere. Es allen recht zu machen, ist eben unmöglich.
Vielleicht sind Sie auch verärgert, weil ich Ihren Rechner falsch eingeordnet habe und Sie fragen sich, welcher mir denn am liebsten war. Ganz ehrlich: Der ZX 81, ihm verdanke ich mein Hobby. Vielleicht denken Sie auch manchmal an den Oldie, der Ihnen die EDV näherbrachte? Schreiben Sie uns darüber. *GS*□

Das Renn-Tier

IN CPC-SPECIAL HEFT 5/88
stellten wir Ihnen den Archimedes von Acorn vor
und versprachen, am Ball zu bleiben.
Dieses Versprechen wollen wir hiermit einlösen
und bringen einen kompletten Testbericht des neuen
Computerwunders.





Jetzt ist es offiziell: Der Archimedes ist da. Pünktlich zur Hannovermesse CeBIT flatterten uns die Pressemitteilungen des holländischen Importeurs Eeckhorn ins Haus, welche die ersten offiziellen Verkaufspreise für Deutschland enthalten. Gleichzeitig erhielten auch dreihundert Fachhändler in Deutschland ein ähnliches Schreiben, um auf Archie – so der Kosename – aufmerksam zu machen. Dies alles macht deutlich, daß entgegen mancher Unkenrufe die Auslieferung des Archimedes im deutschen Markt kurz bevorsteht. Bei uns steht er in Form eines Modelles 310 bereits etwas länger, wir hatten in der Zwischenzeit Gelegenheit, uns dem Flitzer aus Großbritannien zu widmen.

Computer wunder

Wie immer beginnt auch beim Archimedes alles beim Auspacken. Was aus dem kleinen Karton zum Vorschein kommt – CPU, Tastatur und Maus sowie die Handbücher – sieht so gar nicht nach Computerwunder aus. Das angenehm gestylte Gehäuse der Zentraleinheit ist sehr zierlich und erweckt eher den Anschein eines größeren Homecomputers. Weitaus professioneller ist da schon die Tastatur anzusehen, die mit 103 Tasten jedem PC Konkurrenz machen kann. Zu diesem guten Eindruck passen auch der separate Cursorblock und die abgesetzte Zahlentastatur.

Das Schreibgefühl der Tastatur ist hervorragend und angenehm, zwei ausklappbare Stützen an der Unterseite sorgen für die richtige Stellung beim Schreiben. Positiv zu vermerken ist auch ein ausklappbarer Konzepthalter über den Funktionstasten, in den vorgefertigte Tastaturschablonen eingesteckt werden können.

Das zweite Eingabegerät, die Maus, darf bei einem fortschrittlichen Computer natürlich auch nicht fehlen. Die Archimedes-Maus überrascht dabei dadurch, daß sie über drei Tasten verfügt. Auf den ersten Blick mag das zwar etwas umständlich erscheinen, doch die Praxis zeigt, daß die drei Knöpfe durchaus sinnvoll sind. Bei herkömmlichen Mäusen ist eine Taste für Menü-Auswahlen und die andere zum Aktivieren bestimmter Punkte vorhanden. Diesen Zweck erfüllen die linke und mittlere Taste der Archimedes-Maus ebenso.

Zusätzlich aber gesellt sich noch die rechte Maustaste hinzu, die für eine direkte Anwahl zuständig ist. Darunter ist zu verstehen, daß mit der rechten Taste bei einmaliger Betätigung der von Amiga oder Atari ST bekannte Doppel-Klick erreicht wird.

Vielfältige Anschlußmöglichkeiten

Nach dem Auspacken und Ansehen folgt logischerweise das Anschließen des Computers. Auch hier zeigt sich der Archimedes von seiner besten Seite. Da das Netzteil eingebaut ist, muß lediglich ein Stromkabel verwendet werden, um dem Computer seinen Lebenssaft zu verpassen.

Beim Monitoranschluß hat der Käufer die Wahl zwischen Monochrom



Generalimporteur André van den Berg

und RGB (Analog-)-Monitor, für beide befinden sich Buchsen am Gerät. Der Monitor kann dabei getrost auf der Zentraleinheit Platz nehmen, denn der Archimedes ist aus solidem Blech gearbeitet.

Auch für den guten Ton ist gesorgt. Der Archimedes beherrscht unglaubliche acht Tonkanäle in Stereo, daher muß ein entsprechender Stekker vorhanden sein. Dieser findet sich ebenfalls an der Gehäuserückseite in Form einer Buchse für einen Klinkenstecker. Hiermit kann der Computer etwa an Verstärkerboxen oder die heimische Stereoanlage angeschlossen werden.

Wer nichts dergleichen zur Verfügung hat, muß nicht gleich verzweifeln: Im Computer eingebaut befindet sich ein kleiner Lautsprecher, der ebenfalls für den nötigen Ton sorgen kann und bei Bedarf auch abschaltbar ist.

Auch für Drucker ist gesorgt, eine parallele Schnittstelle nach PC-Norm sorgt für den richtigen Anschluß an Centronics-Drucker. Lediglich im Bereich der seriellen Schnittstelle war Acorn für den jetzigen Zeitpunkt ein wenig zu zukunftsorientiert. Es

Druckerschnittstelle

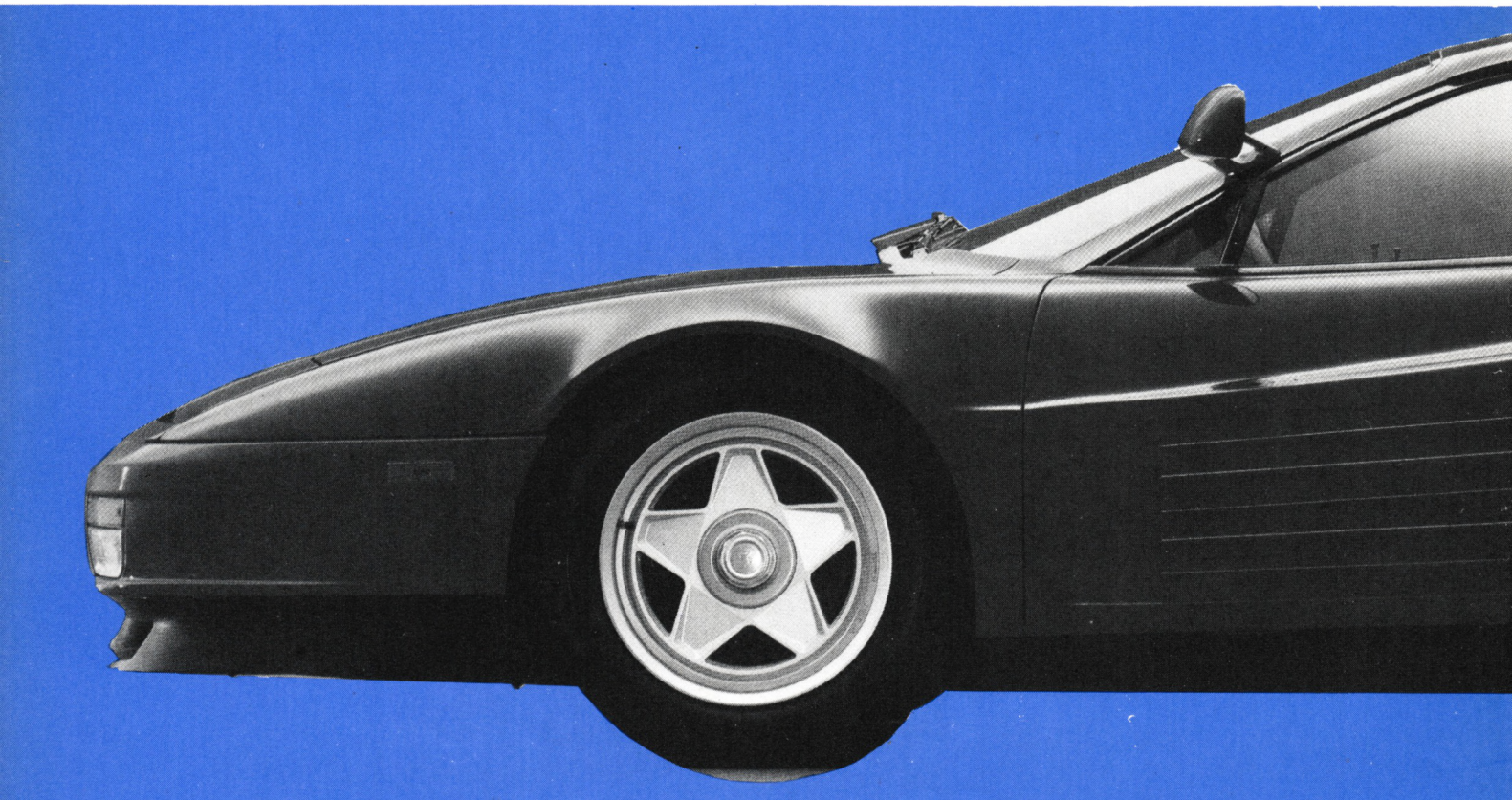
wurde eine neue RS 432 installiert. Diese kann zwar mit einem selbstgebastelten Interface auch für RS 232-Peripherie verwendet werden, doch dürften sich technisch weniger versierte Benutzer hiermit etwas schwer tun. Es ist allerdings zu erwarten, daß ein entsprechendes Anschlußkabel angeboten wird.

Bereits in CPC-WELT 5/88 lasen Sie von Archimedes, einem neuen Computer aus England. Da wir uns in der Zwischenzeit sehr ausführlich mit diesem Rechner befassen und uns von seiner Leistungsfähigkeit überzeugen konnten, kamen wir zu dem Entschluß, dem Archimedes ab sofort hier in der CPC-WELT eine feste Rubrik einzurichten. Diese wird in jeder Ausgabe enthalten sein, der Umfang hängt vom behandelten Thema ab. Damit wollen wir Sie bereits frühzeitig mit diesem Computer vertraut machen, da er unserer Meinung nach neue Maßstäbe setzen wird. Wichtig ist dabei, daß der Archimedes weniger als Konkurrenz zu bewährten Computern der MS-DOS-Welt gesehen werden soll, sondern als Erweiterung der kompletten Computerwelt.

Die Entscheidung zur festen Archimedes-Rubrik fiel uns um so leichter, da wir auf unserem Verlagsstand zur CeBIT in Hannover eine überwältigende Resonanz der Leser zum Archimedes-Bericht feststellen konnten. Wir hoffen, mit dieser Entscheidung auch in Ihrem Sinne gehandelt zu haben. *ts/gs*

In unserer letzten Ausgabe lasen Sie, daß der Archimedes sich nach dem Einschalten mit seinem BASIC-Interpreter meldet. Zum damaligen Zeitpunkt war dies zwar richtig, inzwischen hat sich hier aber etwas getan: In der Grundeinstellung wird nun sofort nach dem Einschalten das Desktop eröffnet, welches bisher noch auf Diskette vorlag.

Dieses Desktop, WIMPS (Windows Icons Mouse Pointer System) genannt, kann als Mischung zwischen GEM und der Amiga-Workbench angesehen werden. Für Diskettenstationen und andere Menübefehle – beispielsweise die Systemuhr – sind in einer Leiste am unteren Bildrand Icons angebracht, die mit dem Mauszeiger angewählt werden können.



Überraschend ist dabei die Geschwindigkeit, denn praktisch ohne jede Verzögerung erscheinen die angewählten Menüs auf dem Bildschirm. Noch überraschender aber: Das Desktop ist in reinstem BASIC geschrieben. So wird gleich bei der ersten Berührungsaufnahme mit Archimedes deutlich, wie schnell dieser Rechner ist.

Denn das BASIC-Desktop schlägt beispielsweise GEM, egal ob auf Atari ST oder einem MS-DOS-Rechner, in der Geschwindigkeit um Längen. Und das, obwohl GEM in C geschrieben ist. Das BASIC bleibt jedoch ebenfalls eingebaut, denn auch diese Benutzeroberfläche kann sofort angewählt werden. Um das Trio noch perfekt zu machen, gesellt sich schließlich auch noch das DOS (Disk Operating System) des Archimedes zu den Eingabemöglichkeiten.

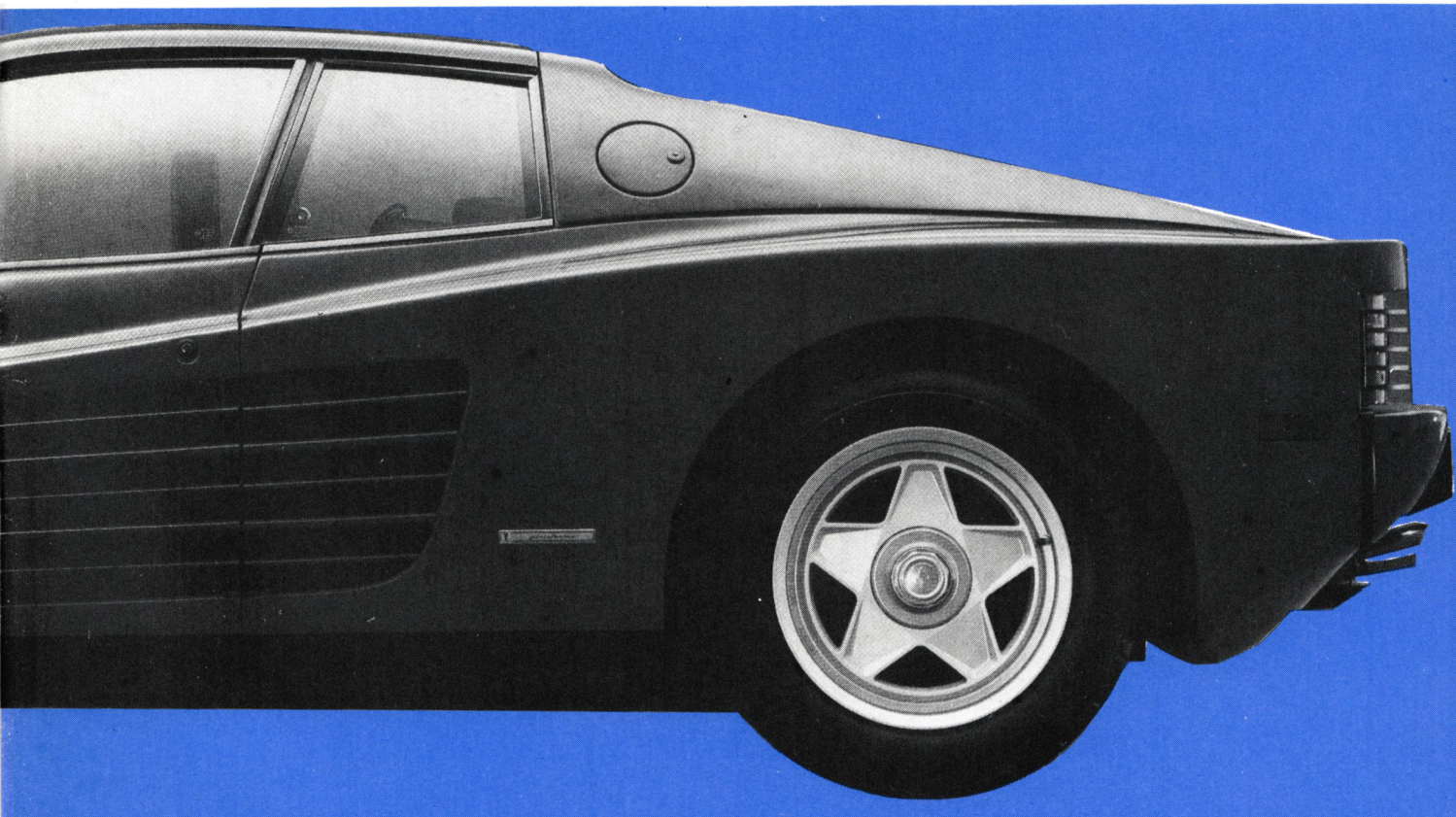
Schnelles Desktop

Das DOS orientiert sich mit andersnamigen Befehlen, aber ähnlicher Struktur sehr stark an MS-DOS und kann daher von vorbelasteten Computereignern sehr schnell begriffen werden. Dabei ist sehr wichtig, daß alle drei Benutzeroberflächen (WIMPS, BASIC und DOS) bereits fest im Rechner eingebaut und sofort nach dem Einschalten verfügbar sind.

Um gleich beim Thema MS-DOS und Geschwindigkeit zu bleiben: Der Archimedes verfügt sowohl über einen Software- als auch einen Hardware-Emulator für den PC-Standard. Uns lag zum Zeitpunkt unseres Tests nur der Software-Emulator vor, welcher trotzdem alles bisher dagewesene schlug: Mit den bekannten Norton-Utilities ermittelten wir die Geschwindigkeit des MS-DOS-Emulators und stellten fest, daß dieser fast exakt der Geschwindigkeit eines herkömmlichen IBM-PC gleichkommt (Faktor 0,98). Um einen kleinen Vergleich zu nennen: Der Software-Emulator des nicht gerade langsamen Amiga bringt es auf einen Faktor von gerade 0,2.

Soft- und Hardware-Emulator für MS-DOS

Auch in Sachen Kompatibilität konnte uns der Emulator überraschen. In unserem Test fand sich keinerlei Software, die nicht einwandfrei funktionierte, egal ob Word, Wordstar oder Grafikprogramme. Wer schon einmal mit einem Software-Emulator gearbeitet hat, weiß, was diese Leistung bedeutet. Der Preis für den Software-Emulator ist allerdings mit knapp 500 Mark etwas hoch ausgefallen. Der hardwareseitige Emulator, welcher auf Steckmodul im Rechner untergebracht wird, soll laut Hersteller um etwa das Dreifache schneller sein als ein Original-IBM-PC.



Die Erweiterungsfähigkeit des Archimedes ist enorm. In den 300er-Modellen befindet sich Platz für zwei Erweiterungskarten (Podule genannt), die 400er-Serie bietet gar vier Steckplätze an. Verfügbar sind an Erweiterungskarten beispielsweise Festplattencontroller (in der 400er-Serie bereits eingebaut), MIDI-Interfaces oder Netzwerkkarten. Auch externe Peripherie wie Genlocks oder Digitizer sind erhältlich.

Entscheidend ist auch die Grafikfähigkeit von Archimedes. Zwanzig verschiedene Auflösungsarten stehen dabei zur Verfügung, maximal beherrscht der Archimedes eine Darstellung von 640 mal 512 Bildpunkten. Der Benutzer kann aus einer möglichen Palette von 4096 Farben schöpfen und davon maximal 256 Farben gleichzeitig darstellen. Das dürfte auch anspruchsvollen Aufgaben genügen.

Bei all diesen überraschenden Fähigkeiten darf eines nicht vergessen werden: Die komplette Leistungsfähigkeit des Archimedes in Sachen Sound, Grafik und Geschwindigkeit läßt sich mit dem eingebauten BASIC realisieren. Keine komplizierten Hochsprachen wie Assembler stellen Einsteiger vor unlösbare Probleme.

Das Archimedes-BASIC kann überdies mit seinen Möglichkeiten als das wohl leistungsfähigste BASIC bezeichnet werden, welches derzeit für einen Computer verfügbar ist. Da es derart umfangreich und leistungsstark ist, werden wir in diesem Bericht nicht näher darauf einghen. Wir werden uns damit in einem eigenen Bericht befassen.

Bei der verfügbaren Software handelt es sich meist um das erste Argument aller Skeptiker, wenn es darum geht, einem neuen Computer düstere Zukunft zu bescheinigen. Im Falle des Archimedes fällt das Argument allerdings für den deutschen Anwender aus. Da Acorn in Großbritannien ein sehr bekannter Hersteller ist, wurden bereits vor der Markteinführung des Archimedes in Deutschland sehr viele Programme entwickelt.



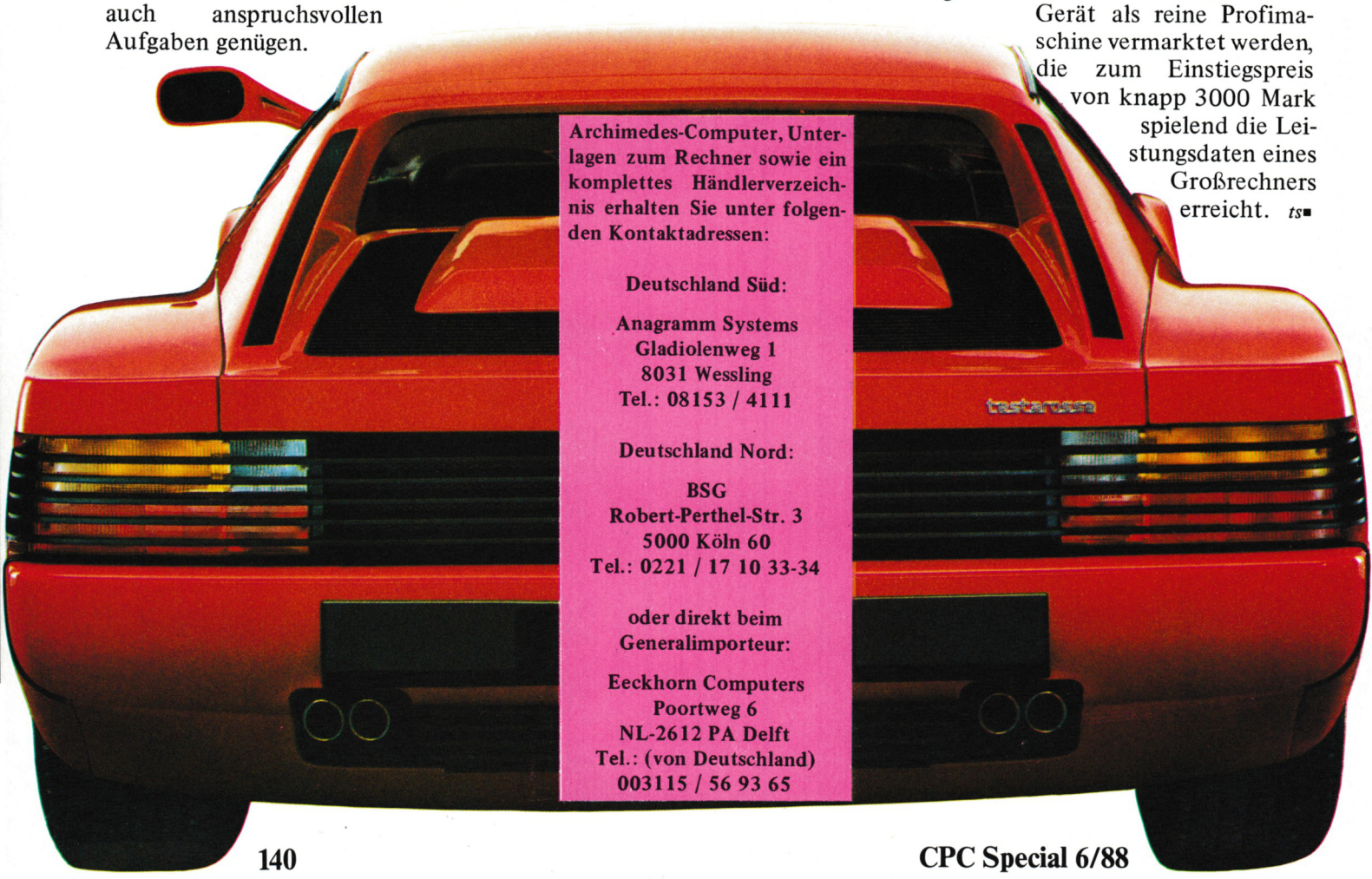
WIMPS, das Desktop

Uns lag zum Zeitpunkt des Tests ein etwa 80seitiger Katalog über die derzeit verfügbare Software vor, von Programmiersprachen über Anwenderprogramme wie Grafik- oder Textverarbeitung bis hin zu Spielen ist alles dabei, was der Käufer braucht.

Um nur zwei Beispiele zu nennen: Im Anwenderbereich sind Logistix (eine Tabellenkalkulation) und FirstWord für Textverarbeitung verfügbar. Beides sehr illustre Produkte, die sich schon auf anderen Computern einen guten Namen geschaffen haben.

Fazit

Der Archimedes setzt zweifellos neue Maßstäbe im Computermarkt. Eines soll dabei allerdings nicht vermutet werden: Der Archimedes ist nicht gedacht, den beliebten Computern vom Schlage eines Atari ST oder Amiga das Wasser abzugraben. Stattdessen soll dieses Gerät als reine Profimaschine vermarktet werden, die zum Einstiegspreis von knapp 3000 Mark spielend die Leistungsdaten eines Großrechners erreicht. *ts*



Archimedes-Computer, Unterlagen zum Rechner sowie ein komplettes Händlerverzeichnis erhalten Sie unter folgenden Kontaktadressen:

Deutschland Süd:

Anagramm Systems
Gladiolenweg 1
8031 Wessling
Tel.: 08153 / 4111

Deutschland Nord:

BSG
Robert-Perthel-Str. 3
5000 Köln 60
Tel.: 0221 / 17 10 33-34

oder direkt beim Generalimporteur:

Eeckhorn Computers
Poortweg 6
NL-2612 PA Delft
Tel.: (von Deutschland)
003115 / 56 93 65

dk'tronics Silicon Disc

Das dritte Laufwerk

Eine RAM-Disk für die CPCs

Die englische Firma dk'tronics bietet diverse Erweiterungen für die CPCs an. Wir haben für Sie eine RAM-Disk unter die Lupe genommen, die 256 KByte Speicher bietet.



Ein zusätzlicher Speicherplatz von 256 KByte ist genau das, was sich mancher CPC-Anwender schon lange gewünscht hat.
Eine RAM-Disk ist nichts anderes als eine Speicher-Erweiterung, die Sie mit Hilfe der bekannten Dateiverwaltungs-Befehle ansteuern können. Die Erweiterung verhält sich also so, als wäre ein zusätzliches Laufwerk an Ihren CPC angeschlossen.
Dieses virtuelle Laufwerk kam bei uns in einer unauffälligen Verpakkung an, die zwei Steckmodule und ein recht umfangreiches Handbuch beherbergte. Wie bei allen Produkten von dk'tronics ist auch bei diesen Modulen der Erweiterungs-Bus durchgeschleift. Beim Anschluß des Floppy-Laufwerks oder von Erweiterungen treten daher keine Probleme auf.

Wir haben versucht, die RAM-Disk durch Anschluß zusätzlicher Erweiterungen zum Abstürzen zu bringen. Es ist uns nicht gelungen. Der MAXAM-Assembler half uns sogar, die neuen RSX-Befehle herauszufinden, ohne einen Blick ins Handbuch zu werfen.

Anschluß ohne Probleme

Da die Silicon Disc (so die Bezeichnung, die dk'tronics diesem Produkt gegeben hat) als zusätzliches Laufwerk fungiert, muß bereits mindestens ein Floppy-Drive angeschlossen sein. Der Anschluß einer Vortex-Floppy war leider nicht möglich. Dies ist verständlich, da es sich beim Hersteller der Silicon Disc um eine englische Firma handelt. Vortex ist jenseits des Kanals schließlich kaum bekannt.

Nach den ersten Experimenten wollten wir endlich „richtig" mit der Silicon Disc arbeiten. Also: Handbuch genommen, aufgeschlagen – großes Staunen. Keine Zeile über die RAM-Disk. Statt dessen wird der Lichtgriffel von dk'tronics beschrieben.

Ein Blick ins Inhaltsverzeichnis klärt das Mißverständnis. Wohl, um Herstellungs-Kosten zu sparen, sind die Beschreibungen aller CPC-Erweiterungen dieser Firma in einem Heft zusammengefaßt. Doch Sparmaßnahmen können auch übertrieben werden. Matrixdrucker-Schrift von miserabler Qualität macht einen benso schlechten Eindruck wie das DIN-A6-Format des Heftchens. Englische Sprachkenntnisse sind ebenfalls nötig; eine deutsche Übersetzung lag zumindest uns nicht vor.

Software im ROM

Insgesamt jedoch ist die Beschreibung knapp, aber vollständig. Wenn es nicht schon durch einen Blick auf die Module selbst klar wurde, so ist es nach der Lektüre des Handbuches offensichtlich: Eines der Steckmodule enthält die RAM-Erweiterung, das andere das Betriebssystem für die Silicon Disc.

Dadurch wird das umständliche Laden der Software von Diskette überflüssig. Außerdem sparen Sie einiges an RAM-Speicher. Für BASIC stehen nach dem Anschluß der Silicon Disc nur 450 Byte weniger zur Verfügung als vorher. Weshalb jedoch nicht alle Chips in einem einzigen Gehäuse untergebracht wurden, bleibt unklar. Beim Anschluß zusätzlicher Erweiterungen wächst der Expansion Port doch um etliche Zentimeter nach hinten. Das kann zu Schwierigkeiten mit den ohnehin viel zu kurzen Kabeln für Monitor und Stromversorgung führen. Silicon Disc, Assembler-Modul und Floppy-Controller sind zusammen gerade noch erträglich.

Die ersten Versuche

:SDISC
initialisiert die RAM-Disk.

Haben Sie zwei Laufwerke angeschlossen, so wird dies automatisch erkannt. Die Silicon Disc wird in diesem Fall mit C: angesprochen.
Der Befehl

:LOADDISC

kopiert den Inhalt einer Drei-Zoll-Diskette in die RAM-Disk. Bei normalen Disketten gab es keine Probleme; die Silicon Disc streikte nur dann, wenn der CPC 464 Disketten im CP/M-Plus-Format lesen sollte.

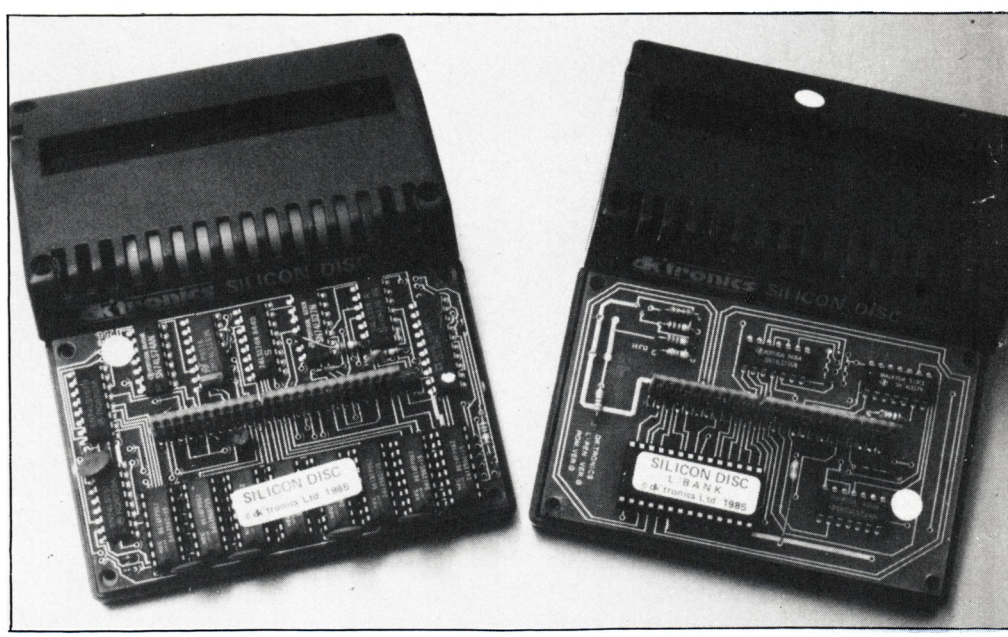
Nachdem der Inhalt einer Diskette übertragen wurde, muß die RAM-Disk mit dem SDISC-Befehl neu initialisiert werden. Dieser Hinweis im Handbuch ist uns beim ersten Durchlesen entgangen. Die Meldung "Drive B: disc missing" verursachte uns daher einiges Kopfzerbrechen. Das hätte sich wohl eleganter lösen lassen können.

Die Arbeitsweisen des LOADDISC-Befehls und seiner Umkehrung SAVEDISC stellten ebenfalls nicht die günstigste Lösung dar. Der Inhalt der Zieldiskette, egal, ob RAM- oder normale Disk, wird dabei in jedem Fall gelöscht. Da die Silicon Disc genügend Kapazität bietet, um den Inhalt von zwei nicht vollständig genutzten Disketten aufzunehmen, sind für eine Kopie oft Umwege nötig.

CP/M muß gepatcht werden

Das Betriebssystem CP/M bietet eine Lösung an. Wenn Sie die Silicon Disc damit betreiben wollen, müssen Sie zunächst Ihre CP/M-Version patchen. Nach der Anleitung im Handbuch ist dies innerhalb von zehn Minuten geschehen. Danach steht für CP/M eine um wenige Byte geringere TPA zur

Wenige Bauteile, große Wirkung



Verfügung als in der Original-Version. Dies beeinträchtigt die Funktion der Programme DISKCOPY und COPYDISK: Sie benötigen den vollen Speicherplatz und funktionieren nicht mehr. Dafür können Sie jetzt PIP anwenden, auch wenn Sie nur ein Laufwerk angeschlossen haben. Die RAM-Disk fungiert als Laufwerk B.
Wir haben mit Hilfe der RAM-Disk Dateien jeder Art kopiert. Auch Standard-Software lief reibungslos. Das einzige Problem stellten Programme dar, die mit Turbo Pascal compiliert wurden. Dieser Compiler benutzt die volle TPA des CP/M-Systems. Derart erstellte COM-Files sind mit der RAM-Disk nicht mehr lauffähig. Nachdem wir den Quelltext auf dem virtuellen Laufwerk B erneut compiliert hatten, funktionierten die Programme wieder.

Reset-sicher

Nach diesen Experimenten wollten wir CP/M wieder verlassen. Wie immer wählten wir dafür die bequeme Methode: CTRL-SHIFT-ESC. Erst nach diesem Reset fiel uns ein, daß wir den Inhalt der RAM-Disk noch auf Diskette sichern mußten.
Wir glaubten schon alles verloren. Doch dem war nicht so: Nach einer Neu-Initialisierung der Silicon Disc erschien die Meldung

Silicon Disc is Drive B:
124K from 256K available

Schnell nach B gewechselt, CAT ausgeführt: Alle Dateien waren noch da. Solange ein Programm mit einem Software-Reset unterbrochen werden kann, können Sie also getrost mit der RAM-Disk arbeiten. Nur ein totaler System-Absturz kann ihren Inhalt zerstören.

Beeindruckende Zugriffszeiten

Auch die Leistungen, die die Silicon Disc in BASIC bringt, sind beeindruckend. Das Laden und Speichern von Programmen geschieht blitzschnell.
Ein erster Versuch, die Geschwindigkeit zu testen, brachte nicht gerade beeindruckende Ergebnisse. Die Aufgabe war, eine sequentielle Datei mit 3.000 Zahlen zu beschreiben und diese anschließend wieder zu lesen. Die DDI-1-Floppy war nur um eine Sekunde langsamer als die Silicon Disc. Diese Aufgabe verlangte allerdings nicht allzuviele Kopf-Bewegungen von der mechanischen Floppy. Deshalb ließen wir 20mal hintereinander dieselbe Grafik als Binärdatei einlesen und auf dem Bildschirm anzeigen.
Dabei zeigte sich, wie schnell die Silicon Disc wirklich ist. Sie benötigte nur 14,5 Sekunden, während die normale Floppy eine Minute und 28 Sekunden zu tun hatte.
Der letzte Test, den wir durchführten, war der Kompatibilitäts-Test. Schließlich sollen bereits vorhandene Programme auch mit der RAM-Disk funktionieren, ohne daß große Änderungen nötig sind.

Kompatibilität

In BASIC gelang dies einwandfrei. Die bekannten Befehle für die Dateiverwaltung gelten weiterhin. Auch CP/M-Software funktionierte.
Besonders kritisch sind bekanntlich Assembler-Programme. Auch damit versuchten wir, einen Fehler in der RAM-Disk zu finden. Doch es wollte nicht gelingen. Jede Software, die nur Einsprung-Vektoren der Firmware benutzt, funktionierte zu unserer Zufriedenheit. So kann beispielsweise das Directory der Diskette auf die gewohnte Weise ausgelesen werden.
Die Software der Silicon Disc ist also wohldurchdacht. Doch die Vorgehensweise, sich streng an das Konzept der Amstrad-Computer zu halten, birgt auch Nachteile.
Am schwersten fällt ins Gewicht, daß das leidige Problem der Garbage Collection weiterhin besteht. Es werden auch diejenigen enttäuscht, die gehofft haben, eine Diskettenverwaltung mit Random-Access-Dateien zu erhalten.
Die RAM-Disk von dk'tronics ist dennoch eine lohnenswerte Anschaffung. Wer viel programmiert, weiß schnelle Zugriffszeiten zu schätzen. Außerdem wird das normale Laufwerk geschont, das gerade bei Compiler-Sprachen oft stundenlang am Rotieren ist.
Gerade für die Entwicklung von Programmen ist auch die hohe Speicher-Kapazität von 256 K-Byte von Bedeutung. Der Platz auf einer Drei-Zoll-Diskette ist etwa durch einen Compiler, einige Modul-Bibliotheken und Backup-Kopien schnell erschöpft. Eine Kapazität von 256 KByte stellt in Verbindung mit den kurzen Zugriffszeiten der Silicon Disc eine große Erleichterung dar.
Doch auch für BASIC-Programmierer und -Anwender lohnt sich die RAM-Disk. Bilder werden schneller geladen, Dateien schneller bearbeitet.

Den Preis wert

Ein kleiner Wermutstropfen jedoch bleibt: der Preis. Während die 64-KByte-Version der RAM-Disk bereits für 148 Mark zu haben ist, kostet das 256-KByte-Modell die stolze Summe von 378 Mark.
Nach Angaben der Firma Weeske, die uns das Modul zur Verfügung stellte, hat diese Preiserhöhung erst vor kurzer Zeit stattgefunden. Der Grund dafür liegt in der hohen Nachfrage nach RAM-Chips, die die Industrie nicht in der benötigten Menge liefern kann. Der Preis der Silicon Disc soll jedoch in absehbarer Zukunft nicht weiter steigen.
Als echtes zweites oder drittes Laufwerk ist die RAM-Disk jedoch auch diesen hohen Preis wert. Wir möchten sie bei der Arbeit mit dem CPC nicht mehr missen. *AE*□

COMPUTERN LEICHT GEMACHT

Das PC-Magazin



NEU

Jetzt an ausgewählten Kiosken und im Bahnhofs-Buchhandel