CPC-WELT

CPC SPECIAL

DM 14,80/ÖS 124/SFR 14,80 **Nr. 1/89**

SOFT-WARE JAHR-BUCH 1989

Die besten Anwender-Programme

Die schönsten Spiele

Hilfreiche Utilities

Das Super-Sonderheft für Ihren CPC

Sonderheft Nr. 1/89 — DM 14,80/ÖS 124/SFR 14,80 CA-Special

COMPUTER-TEST-JAHRBUCH

COMPUTER TEST JAHR BUCH

Rund 150 Seiten Kaufberatung

Computer von A bis Z

Was sie können
Was sie kosten
Was sie leisten

Jetzt an Ihrem Kiosk

Der Computer-Einkaufsführer

Für Sie getestet:
Computer
Programme
Zubehör

Guten Tag

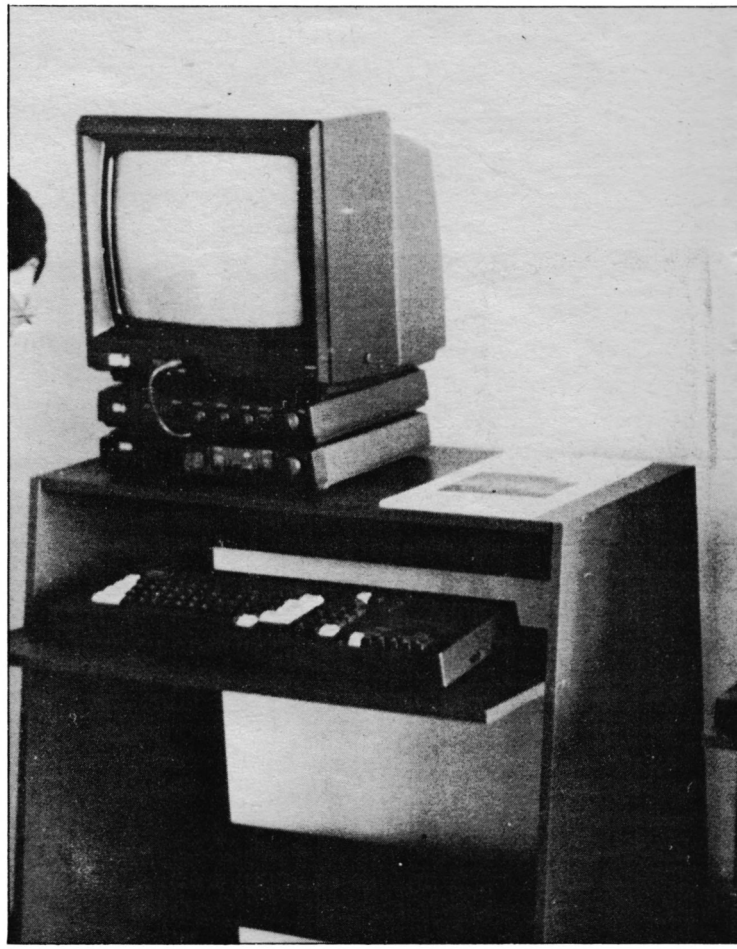
Das Jahr 1988 brachte allerlei Trubel rund um den CPC. Das begann mit der Übernahme des Vertriebs durch den Hersteller Amstrad, der nicht so richtig anlaufen wollte. Plötzlich waren die CPCs, vormals von Schneider vertrieben, aus den Geschäften verschwunden. Wie es mit dem ,,kleinen CPC" weitergehen sollte, wußte anscheinend niemand so recht.
Wen wundert's, daß sich unter diesen Voraussetzungen auch manche Software-Firma aus dem CPC- beziehungsweise CP/M-Markt zurückzog? Ein Beispiel hierfür ist die Firma Heimsoeth, bekannt als Hersteller des Turbo-Pascal-Compilers.
Hatte man wenige Monate zuvor noch eine CP/M-Version der Fortgeschrittenen-Sprache Modula-2 auf den Markt gebracht, so suchte man in den Läden nunmehr vergeblich danach. Direkt beim Hersteller war dieser Compiler zwar noch zu beziehen; doch wer wußte schon davon?
Es scheint, als hätte man den CPC vielerorts unterschätzt. Auch wenn der Trend zum PC nicht zu verkennen ist, wächst die Zahl der CPC-Anwender immer noch. Die Leistungen dieses Geräts können sich wirklich sehen lassen. Schade nur, daß er mittlerweile zum Billig-Rechner degradiert wurde.
Software für den CPC wird also immer noch gebraucht. Die CPC WELT hat in den letztjährigen Ausgaben zahlreiche leistungsfähige Programme vorgestellt, deren Highlights wir für Sie in diesem Sammelband zusammengefaßt haben. Damit wollen wir allen CPC-Einsteigern und -Anwendern ,,Software satt" anbieten, die Sie sonst nirgendwo finden werden. Vom Spiel bis zum Anwender-Programm finden Sie in diesem Heft zahlreiche vergnügliche und hilfreiche Programme, die Ihnen — da sind wir ganz sicher — viele amüsante Spiel-Stunden und wirksame Arbeitshilfen am CPC bieten werden. Einige Programme sollen hier herausgegriffen werden.
Der CPC-Graphic-Designer etwa ist ein Grafik-Editor der Spitzenklasse, der allen 464-Besitzern sicher große Freude machen wird. Das Programm ist so umfangreich, daß wir es nicht in vollem Umfang abdrucken konnten. Die Version, die Sie in diesem Heft finden, ist zwar lauffähig; sämtliche Zeichensätze für das Programm konnten wir jedoch nur auf der Softbox-Diskette unterbringen.
Unser Spiel ,,Vollgas" wiederum ist ein Leckerbissen für alle 6128-Besitzer. Geschwindigkeit und Grafik sind für ein BASIC-Programm wirklich außergewöhnlich und brauchen den Vergleich mit den sogenannten ,,professionellen" Spielen nicht zu scheuen. Die Mühe des Abtippens lohnt sich.
Wieder für den 464 ist unsere Befehlserweiterung RSX-Extended-BASIC. Wer nicht die Zeit und Geduld aufbringen kann, Assembler zu lernen, der findet hier hilfreiche Routinen, die mit einem einfachen BASIC-Befehl aufgerufen werden können. Damit können Sie Ihre Programme noch leistungsfähiger gestalten.
Gleich zwei Programme bieten wir den Disketten-Anwendern. DOS 208 formatiert eine Diskette so, daß Sie 208 KByte darauf unterbringen können. Damit können Sie über die Meldung ,,Disc full", die andere CPC-Anwender so oft zu sehen bekommen, nur noch lächeln. Mit unserem Disk-Monitor schließlich können Sie Ihren Disketten auf die Bit und Byte schauen.
Leistungsfähige Programme sind in der Regel sehr umfangreich. Wer nicht tippen will, erhält wie immer die Möglichkeit, alle Programme dieses Heftes auf Diskette zu bestellen. Den Coupon finden Sie auf Seite 64.
Doch auch kommerzielle Software und Zubehör für den CPC sind ein wichtiges Hilfsmittel bei der Arbeit mit diesem Computer. Unsere Testberichte, die Sie in dieser Ausgabe finden, wollen einen kleinen Überblick über die noch verfügbaren Programme und Geräte geben und Ihnen Kaufhilfen bieten. Vom BASIC-Compiler bis zur Sprache C ist alles vertreten.
Besonders hervorzuheben ist dabei die RAM-Disk der Firma dk'tronics. Dieser britische Hersteller war es, der von Anfang an Zubehör für den CPC lieferte. Unsere Erfahrungen mit der Silicon Disc finden Sie ab Seite 17. Und sie sind durchweg positiv.
Für die zahlreichen Freunde von Turbo-Pascal haben wir auch etwas zu bieten. Wenn Sie schon immer wissen wollten, wie Sie in einer Grafik Flächen ausfüllen können: Ab Seite 20 finden Sie ein Programm mit einer ausführlichen Erläuterung. Wer nur in BASIC programmiert, kann mit diesen Hinweisen eine entsprechende Routine selbst entwickeln.
Natürlich dürfen auch die Tips und Hilfen für CPC-Einsteiger nicht fehlen. Unser Checksummer auf Seite 11 sorgt für fehlerfreies Abtippen. Und wenn Sie Probleme mit mehrteiligen Programmen haben, so lesen Sie unseren Beitrag auf Seite 6. Begriffe wie Binärlader und Maschinencode stellen für Sie dann sicherlich keine Geheimnisse mehr dar.
Wer seinen CPC schon seit längerer Zeit besitzt, hat ihn wahrscheinlich bereits um ein Zusatz-Laufwerk erweitert. Nicht selten dürfte dies das Laufwerk der Firma Vortex sein.
Assembler-Programmierer aber haben gerne Zugriff auf die internen Routinen des Floppy-ROMs. Vortex hat dem aber einen Riegel vorgeschoben, indem die entsprechenden Bausteine durch einen trickreichen Kopierschutz abgesichert wurde. Wie Sie trotzdem ans ROM kommen und nachsehen können, was dort abläuft, finden Sie in unserem Beitrag ab Seite 128. Kopieren dürfen Sie es selbstverständlich trotzdem nicht.
Auch wenn es Soft- und Hardware-Hersteller gleichermaßen behaupten: Ein Spielzeug ist der CPC nicht. Vielmehr ist er ein vorzüglicher Spiele-Computer, dessen Qualitäten als Arbeitstier man aber nicht unterschätzen sollte. Und ein besseres Gerät, mit dem Sie auf einfache Weise den Einstieg in die Welt des Programmierens schaffen können, werden Sie kaum finden.
Auch in Zukunft viel Spaß und Erfolg mit Ihrem CPC wünscht Ihnen Ihr
Alwin Ertl

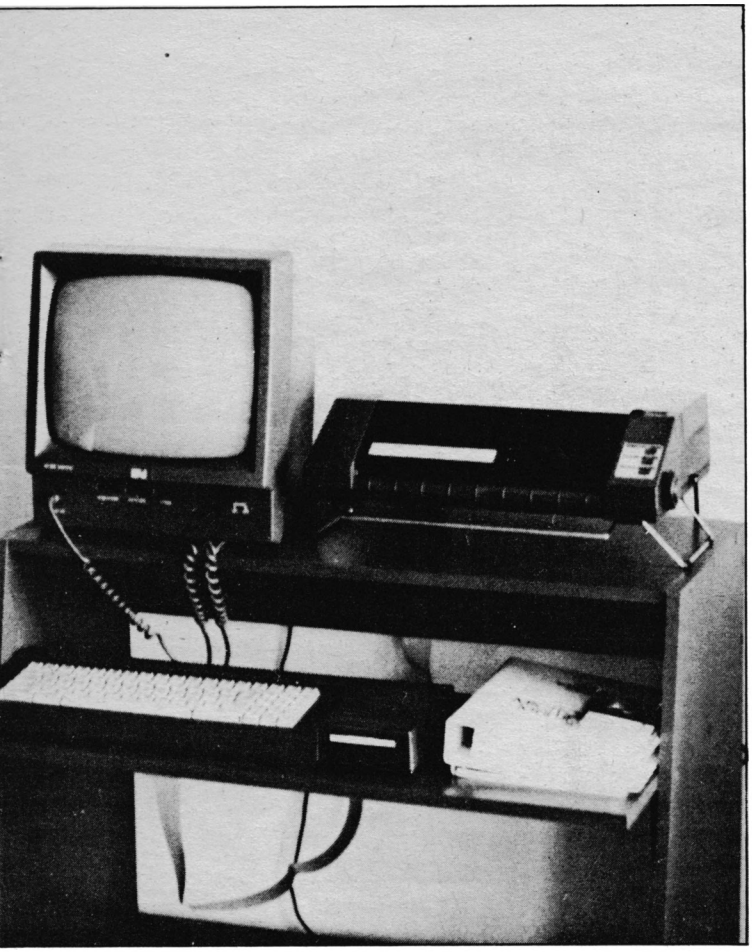
Inhalt

GRUNDLAGEN

TEST

LISTINGS





TIPS & TRICKS

RUBRIKEN

Die leidigen Mehrteiler

Starthilfe

Mehrteilige Programme können noch so exakt abgetippt sein: Wenn die Reihenfolge beim Start nicht stimmt, läuft nichts. Auch ein falscher Speichername führt zu Problemen. Aber "Mehrteiler" müssen nun einmal sein. Sie sind leistungsstark und holen mehr aus dem relativ kleinen CPC-Speicher. Dieser Artikel klärt, wie man mit Ladeprogrammen besser zurechtkommt.

Immer wieder erreichen uns Nachfragen, warum denn Programme in mehrere Teile zerlegt werden. Unsere Hotline oder unsere Programmbearbeiter wissen ein traurig Lied darüber zu singen, wie schwer es ist, die richtige Benutzungsanleitung zu schreiben. Immer fehlt ein wichtiger Punkt, immer bleibt etwas ungeklärt. Dadurch kommt es auch zu Fehlern, weil die Speichernamen nicht korrekt sind, weil in der falschen Reihenfolge gestartet wird oder weil bestimmte Files nicht vorhanden sind. Wir wollen deshalb einmal klären, wieso Programmautoren überhaupt ihr Listing zerpflücken und wo die Fehlermöglichkeiten sein könnten, wenn es am Ende aller Abtipperei doch nicht läuft.

SPEICHERNAME IST WICHTIG

Schneider CPC-Welt kann (und will) seinen Lesern nicht vorschreiben, wie sie ihr Programm zu nennen haben. Es ist völlig egal, ob ein Kontoführungsprogramm unter 'Kasse', 'Konto' oder gar nur einer Nummer abgespeichert wird, solange es aus einem einzigen Listing besteht. In dem Moment, wo ein Programm in zwei Teilen gedruckt ist, sollten Sie jedoch mißtrauisch werden und sich an unsere Namensvorschläge halten. Es sei denn, Sie haben diesen Artikel gelesen, kennen sich etwas aus und ändern das Programm nach Ihren Bedürfnissen um. Wie das geht, wird am Schluß ebenfalls verraten.

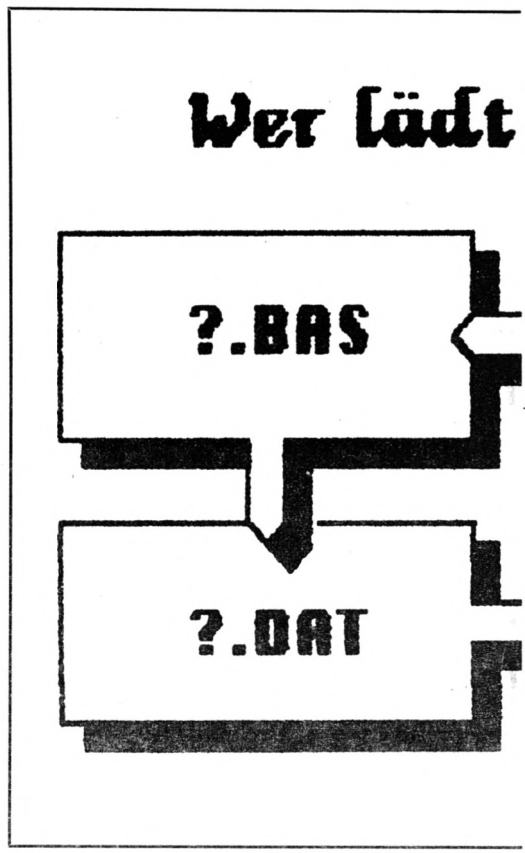
WARUM ÜBERHAUPT PROGRAMMTEILE?

Es gibt gleich drei Gründe, ein Programm zu zerlegen. Der erste und einfachste ist der, daß immer bestimmte Routinen als "Bibliothek" vorliegen, auf die erst einmal zugegriffen werden kann. Denkbar ist, daß in einem Spiel erst einmal wichtige Parameter bestimmt werden – Schwierigkeitsgrad, Anzahl der Leben usw. –, bevor der Hauptteil mit dem eigentlichen Spiel eingeladen wird. Auch die Definition des deutschen Zeichensatzes gehört dazu. Ein kurzes Listing legt fest, auf welchen Tasten die Umlaute liegen. Dieses Programm wird gestartet und ist damit im Speicher aktiv, egal welches Anwenderprogramm danach eingelesen wird. Wichtig ist nur, daß zwischen den beiden Teilen kein Reset (Drei-Finger-Griff mit Shift, Esc und Ctrl) ausgeführt wird. Hat man einmal die deutschen Umlaute definiert, dann reicht dies für alle anderen Programme.

So denken natürlich auch unsere Autoren, die in diesem Fall den Bibliotheksteils 'Umlaute' einfach mitschicken. Und es wird von ihnen natürlich ebenso verfahren, wenn der Speicherplatz knapp wird. Das Listing zum deutschen Zeichensatz zum Beispiel nimmt im Speicher des CPC wesentlich mehr Platz ein als die einmal aktivierten Parameter. Je aufwendiger ein Programm ist und je mehr Optionen der Autor dem Anwender bieten will, desto geiziger muß er mit dem verfügbaren Speicher umgehen. Kein Wunder, daß so zum Beispiel Günther Radestock, der Programmierer von GPaint, die Hardcopy-Routine als separates Programm ablegte.

Aber gerade bei diesem Zeichenprogramm ist noch etwas anderes zu beachten, und damit kommen wir zum dritten Grund, warum es "Mehrteiler" gibt. Es sind dies die Maschinenprogramme. Streng genommen handelt es sich dabei nur um eine schier endlos lange Zahlenkette, die in den richtigen Speicheradressen und in der richtigen Reihenfolge den Programmablauf bewirkt. Aber wie soll man solche Zahlen dem Leser mitteilen? Für jede Adresse einen Poke eingeben?

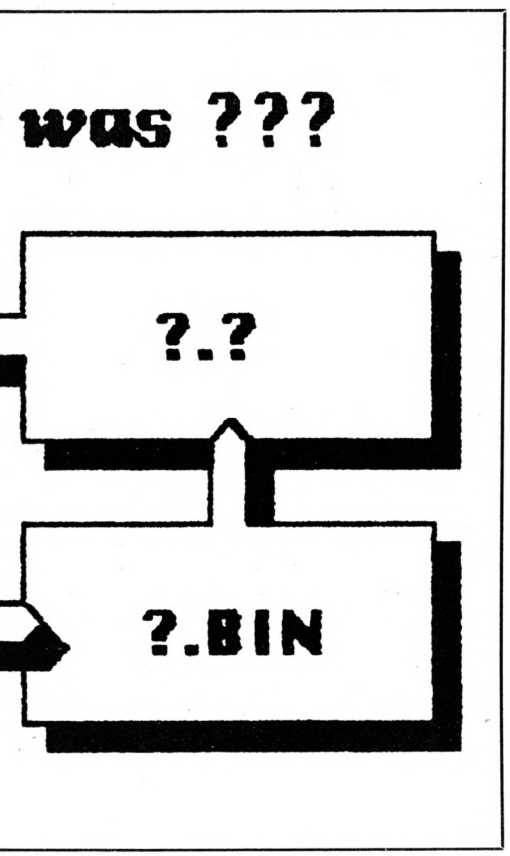
Es ist allgemein üblich, daß man hierzu einen 'Datalader' entwirft und natürlich die dazugehörigen Datas auflistet. Eine FOR-NEXT Schleife liest die hexadezimalen Zahlen, überprüft diese, falls es so programmiert wurde und poked dann den richtigen Wert ab. Dieser Vorgang gehört eigentlich nicht zum Programmablauf, sondern dient der Initialisierung eines Maschinenprogrammes oder auch nur



Mehrteilige Programme führen immer wieder zu Fragen, auch wenn die Dateien durch Extensions gekennzeichnet sind. Der Artikel schafft endlich Klarheit.

von RSX-Befehlen (im Grunde genommen nichts anderes).

Warum, so kann man sich überlegen, soll der Computer bei jedem Programmstart Datas lesen, vergleichen und einpoken? Dies kostet Zeit und Speicherplatz. Letzteres sowohl auf der Diskette als auch im Speicher. Wäre es nicht besser, ein einmal erzeugtes Maschinenprogramm als solches abzuspeichern und bei Bedarf dieses lauffähige



Programm aufzurufen, anstatt es immer wieder neu zu erzeugen? Richtig, es ist sinnvoller, wenngleich eine ungenaue Beschreibung zu Fehlern bei der Bedienung führen kann.

SPEICHERBEREICHE ABSICHERN

Deshalb einmal eine kurze Erklärung dessen, was eigentlich passiert. Dabei wird folgendermaßen vorgegangen: Wie immer wird ein Dataprogramm eingelesen und in die richtigen Speicheradressen abgelegt. Bevor es jedoch gestartet wird, sichert man den Speicherbereich auf Diskette ab. Dies kann einfach mit dem Befehl SAVE "Name", B, Startadresse, Länge geschehen. So wie er hier beschrieben ist, würde der Speicherbefehl auf der Diskette automatisch ein Binärfile abspeichern, welches man an der Extension "BIN" erkennt. Bei der Kassette wird allerdings nicht mit den drei Kennbuchstaben hinter dem Dateinamen gearbeitet.

Der Unterschied zwischen Band und Diskette wird auch beim Einlesen gemacht. Bei einem Laufwerk müssen Sie sich nicht darum kümmern, ob ein Binärfile vorliegt. Starten Sie einach mit RUN "Name". Der Kassettenrekorder verlangt jedoch die Eingabe der Parameter wie beim Abspeichern.

Liegt das Binärfile eines mehrteiligen Programmes auf Diskette (Kassette), dann benötigen Sie den Datalader nicht mehr. Noch einmal sei daran erinnert, daß es Datalader gibt, die immer gestartet werden müssen. Hier ist nur die Rede von den Dataladern, die ein Maschinenprogramm erzeugen und abspeichern.

RATSCHLÄGE ZUM ABTIPPEN BEI BIN-FILE ERZEUGENDEN DATALADERN

Tippen Sie prinzipiell erst einmal alle Teilprogramme ab und speichern Sie diese einzeln, wenn möglich unter den angegebenen Namen. Wenn wir mehrere Teile abdrucken, dann ist dies nötig, und keines der Listings ist überflüssig. Verfahren Sie also nicht so wie einer unserer Leser, der zwei Listings "in einem Aufwasch" hintereinander weggetippt hat und unsere vermeintliche Unzulänglichkeit durch den RENUMBER-Befehl wettmachen wollte.

Wenn der Begleittext den Hinweis auf ein Binärfile gibt, dann starten Sie das entsprechende Programm zuerst. Der Hinweis wird auf jeden Fall im Programmkopf des Dataladers wiederholt. Ist das Binärfile abgespeichert (meist geschieht dies automatisch), dann können Sie den Datalader für immer vergessen; heben Sie sich aber zur Sicherheit noch eine Kopie auf. Starten Sie dann jenes Listing, welches als Hauptprogramm ausgewiesen ist. Es sucht sich "sein" Maschinenprogramm, das Binärfile, und liest es ein. Meist ist damit die Arbeit getan.

Auch bei jenen Dataladern, die kein Maschinenprogramm erzeugen, ist die Sache ähnlich, nur müssen Sie hierbei immer erst mit diesem Programm starten Sie dürfen es also nicht entfernen. Meist steht am Ende der POKE-Schleife eine Befehlszeile, mit der das nachfolgende Hauptprogramm gestartet wird. Das bedeutet aber auch, daß das Hauptprogramm unter diesem Namen abgespeichert sein muß. Auch bei diesem Mehrteiler-Typ dürfte es dann keine Startschwierigkeiten geben.

BIBLIOTHEKEN-START

Bibliotheks-Programmteile verhalten sich zumeist wie ein immer benötigter Datalader, in der Regel wird auch das Hauptprogramm nachgeladen. Es kann jedoch vorkommen, daß man eine Zeichensatz-Definition startet, und nach einer kurzen Dauer meldet sich der Bildschirm mit einem schnöden READY. Damit ist aber noch nichts verloren. Es fehlte ganz einfach der Befehl zum Laden des Hauptprogrammes. Den können Sie selbst mit RUN "Name" im Direktmodus eingeben.

WICHTIG FÜR KASSETTEN-BENUTZER: REIHENFOLGE BEACHTEN

Bei allen beschriebenen Methoden muß der Benutzer des Datenrekorders noch einen wichtigen Punkt beachten. Die Programmteile müssen in der richtigen Reihenfolge abgespeichert werden, um den einwandfreien Start zu gewährleisten.

Vielleicht sind damit einige Unklarheiten beseitigt, die bei den immer größeren und leistungsfähigeren Programmen auftreten können. Wenn sich beim Start von RUN "RSXDEMO" mal wieder nichts rührt, dann wissen Sie, daß hier vorher eine Bibliotheksroutine hätte eingelesen werden müssen.

(GS)

TAS

Konkurrenz für dBase

Man glaubte schon, im Bereich der Dateiverwaltungen gäbe es für den CPC nichts besseres als dBase. Aber die englische Firma Tasman, die sich bereits mit einem Textprogramm hervortat, hatte keine Angst vor der mächtigen Konkurrenz. Das Ergebnis kann sich sehen lassen.

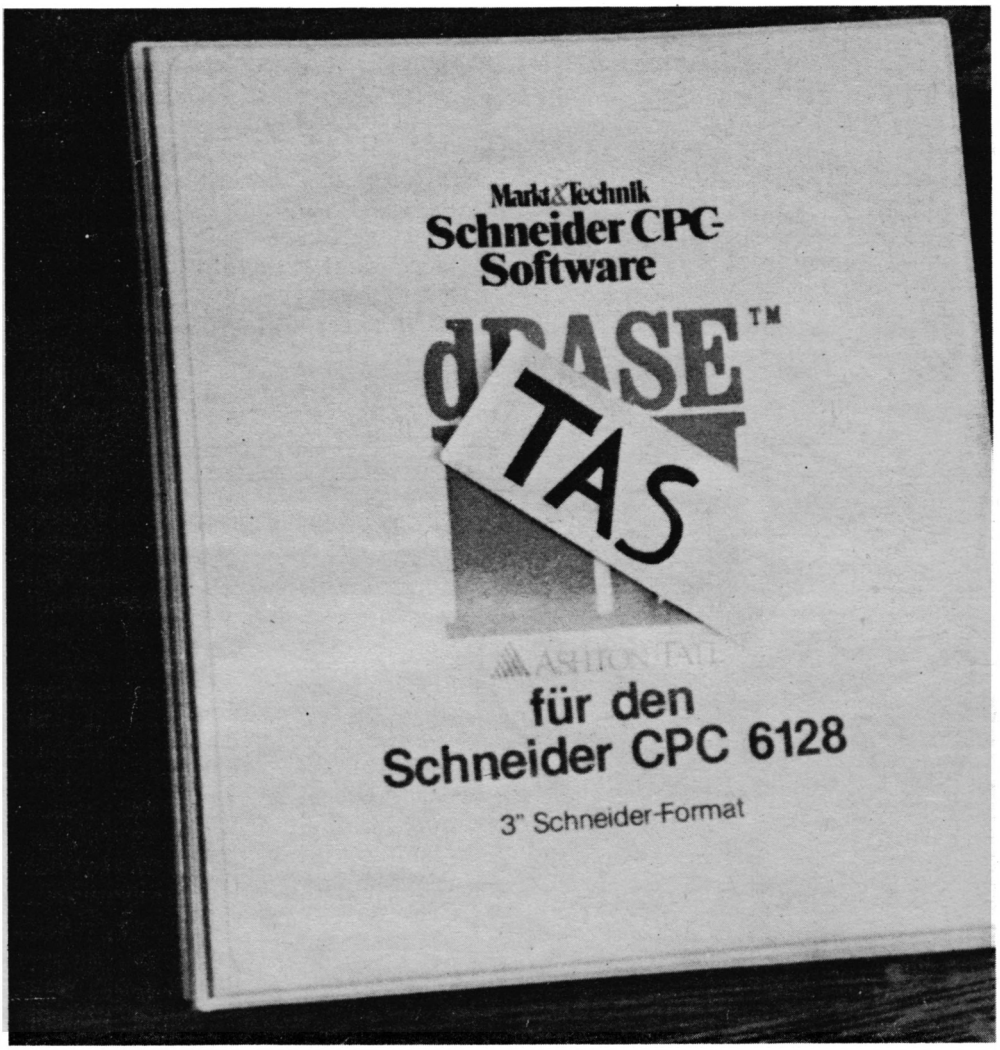
Wenn ein neues Produkt für den Schneider CPC 6128 in der Werbung derart hochgelobt wird wie die hier besprochene relationale Datenbank TAS, ist man immer gespannt, ob sich die vielversprechenden Werbeaussagen in der Praxis auch beweisen, denn die Meßlatte liegt hoch. Eckwerte für Datenbanken auf dem CPC hat der Marktstandard dbase II bereits gesetzt. Ob sich mit TAS nun völlig neue Perspektiven auftun?

RELATIONALE DATENBANK WAS IST DAS?

Viele von uns haben es sich leider angewöhnt, mit computerspezifischen Begriffen sehr gedankenlos umzugehen, ohne vorher zu prüfen, ob die Begriffsinhalte durchgängig geklärt worden sind. Wir wollen diesen Fehler hier nicht wiederholen und zunächst klären, was man sich unter einer relationalen Datenbank vorzustellen hat.

Eine Datenbank verwaltet Daten, das ist sicher nicht neu und auch nicht besonders aufregend. Einfache Dateiverwaltungen, die jeder Basicprogrammierer für eigene Anwendungszwecke ohne große Mühe erstellen kann, erfüllen diesen Sachverhalt auch. Unterschiede ergeben sich erst dann, wenn zwischen Dateien Beziehungen (Relationen) hergestellt werden können. Dies geschieht in der Regel durch sogenannte Index- oder Schlüsselfelder. Adreßdaten lassen sich dann z.B. leicht mit Rechnungsdaten verknüpfen, z.B. über das Schlüsselfeld "Name". Nach diesem Prinzip lassen sich große Datenmengen unterschiedlicher Schwerpunkte sehr effektiv verwalten. Eine relationale Datenbank kann meist aber noch mehr. Sie ist anwenderspezifisch programmierbar mit einer eigenen Programmiersprache, die individuelle Anpassungen jederzeit möglich macht. Diese Anpassungsmöglichkeiten sind aber nicht ohne Auswirkungen auf den Anwender, der mit diesem Werkzeug zwar viele Möglichkeiten erhält, sie aber noch zu beherrschen lernen muß.

Und nun fängt es an, spannend zu werden, da man anfangs doch weit davon entfernt ist, eine spezielle Anwendersoftware zu erwerben.



Tasman wagt den Angriff auf dBase. Diese Datenbank verspricht mehr Leistung und einfachere Bedienung.

BEEINDRUCKENDE MÄCHTIGKEIT

Wenden wir uns nun der Datenbank TAS selbst zu, die durch ihre Zahlenwerte zunächst einen bleibenden Eindruck hinterläßt. Mit TAS lassen sich bis zu 65.535 Datensätze zu je 10.254 Zeichen verwalten,

wobei jedes Feld 254 Zeichen enthalten darf. Bei dieser Angebotsfülle werden eingefleischte Dbase II Fans verschreckt aufhorchen, um sich dann wieder erleichtert zurückzulehnen, wenn sie errechnet haben, daß man dazu etwa 10 Festplatten à 70 Megabyte benötigen würde. Keine Aussage also, was die Praxis anbelangt. Im Vergleich zu

dbase II schneidet TAS dennoch gut ab. Es können z.B. bei TAS bis zu 16 Dateien gleichzeitig geöffnet werden, wobei jede Datei max. 16 Indexfelder mitbringen darf. Bei dbase II sind es nur 2 Dateien mit einem Indexfeld. Der Umfang eines Datensatzes bei dbase II ist mit 1000 Zeichen zudem sehr beschränkt. TAS erlaubt, wie bereits erwähnt, mehr als 10 mal so viel. Soweit die rechnerischen Perspektiven.

Um nun den Besonderheiten von TAS gerecht werden zu können, ist eine Aufschlüsselung der Datenbanksprache notwendig, die sich in ihrer Organisation von dbase II unterscheidet. TAS verfügt über einen "Compiler", der zwar keine eigenständig lauffähigen Programmversionen erzeugen kann, für den Anwender dennoch einige Vorteile mit sich bringt. Zunächst wird jedes mit einem separaten Editor erstellte Quellprogramm mit dem Compiler in einen Zwischencode übersetzt, der sehr kompakt und schnell in der Ausführung ist. Dieser Zwischencode braucht zur Programmausführung entweder TAS selbst oder eines der zusätzlich erhältlichen Runtimemodule, die beim Vertreiber für etwa 60 DM zu haben sind. Welche Vorteile dieses System dem Anwender bringt, liegt auf der Hand. Der Nur-Programm-Anwender braucht nicht, wie unter dbase, das komplette Programmpaket zu erwerben, sondern nur ein weitaus billigeres Runtimemodul. Wenn man jetzt noch die Speicherplatzersparnis des Programmcodes und den Geschwindigkeitsvorteil der Programmausführung in die Waagschale wirft, ergeben sich handfeste Vorteile gegenüber den dbase II Anwendungen. Laute Lobgesänge sind dennoch nicht angebracht, da sicher einige Zeit vergehen wird, bis genügend TAS Anwendungen zu haben sind. Ob sie dann die Verbreitung finden wie die des Marktstandards, ist außerdem fraglich. Dennoch kommt TAS in seiner Programmstruktur professionellen Programmentwicklern entgegen, die dank des compilierten Zwischencodes dem Anwender Einblicke in ihren Quellcode verwehren können.

PROGRAMMENTWICKLUNG MIT HINDERNISSEN

Im Gegensatz zum Konkurrenten dbase II ist bei TAS – wie bereits erwähnt – der Editor nicht im Programm selbst integriert, sondern eine eigenständige Einheit, die nicht mitgeliefert wird. Die Programmvertreiber gehen davon aus, daß man, falls keine Textverarbeitung wie Wordstar vorliegt, den Editor ED. COM von der CP/M Systemdiskette nutzen kann. Zusammen mit der TAS Ablaufverwaltung und dem Compiler ergibt sich ein Trio, mit dem sich leistungsstarke Anwendungen realisieren lassen. In der praktischen Anwendung sieht eine Programmerstellung etwa so aus:

Zuerst legt man die Datenbankspezifikation fest, das heißt, man definiert die Struktur der Datei durch Fixierung des Feldnamens, der Feldgröße und so weiter. Diese Datenstruktur legt TAS in einem Datenverzeichnis (TASDICT) ab, das für Programmentwicklungen dann als eine Art Ablaufschema zur Verfügung steht. Ist diese Arbeit erledigt, kann die eigentliche Programmerstellung mit dem Texteditor beginnen. Selbstverständlich kann man verschiedene Unterprogramme oder Programmodule separat erzeugen und anschließend – genau wie letztlich das Gesamtprogramm – compilieren und durchtesten, bis schließlich eine unter TAS (oder Runtimemodul) lauffähige Programmversion vorliegt.

Doch bis dorthin kann es ein langer Weg sein, der schon so manchem erfahrenen Programmierer schlaflose Nächte bereitet hat. Die Fehlersuche wird mitunter bei TAS zur Tortur, da für jede Syntaxprüfung neue Programmdurchläufe im Editor und Compiler notwendig sind, und selbst dann, wenn sich dort keine Error-Meldung mehr einstellen sollte, ist man vor programmlogischen Hürden, die erst im Ablauf sichtbar werden, nicht gefeilt. Da hat es der dbase II Programmierer doch erheblich einfacher, Programmänderungen auszuführen und neue Versionen durchzutesten. Er muß allerdings langsamere Ausführungsgeschwindigkeiten und speicherplatzintensivere Programmumfänge in Kauf nehmen.

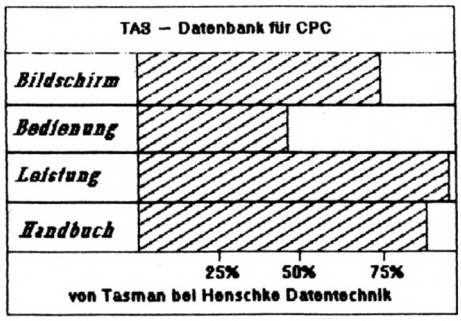
DEUTLICHE VORTEILE IN DER SPRACHE

Die Datenbanksprache von TAS ist zunächst für jeden Neuling sehr komplex und abstrakt. Ein Einstieg ist etwa mit der Neulanderschliessung beim Erlernen einer höheren Programmiersprache wie Pascal oder C vergleichbar. Ähnliche Überlegungen kann man für dbase II

aber auch anstellen. Anfangsschwierigkeiten sind in beiden Programmsprachen zu erwarten. Vergleicht man nun aber die Leistungsfähigkeit beider System, so liegt TAS eindeutig vorn. TAS erlaubt nicht nur die Definition von 254 Variablen, die auch als Arrayvariable (bei dbase II Fehlanzeige) ausgelegt sein können, sondern bietet auch ausreichende Programmstrukturierungsmöglichkeiten, wie IF/THEN Konstruktionen, WHILE and FOR Schleifen sowie GOTO/GOSUB Konstruktionen. Leider verfügt TAS über keine Eingabemöglichkeiten im Direktmodus, so daß z.B. selbst Sortieralgorithmen im Quellcode erstellt werden müssen, bevor sie dem Anwender zur Verfügung stehen. Dieser kleinen Einschränkung, die sich eigentlich schon aus der Compilerkonzeption von TAS ergibt, stehen dann aber die Möglichkeiten gegenüber, den Bildschirmdruck oder allgemeine Ausgabemasken bereits im Quellcode zu erstellen oder Standardfunktionen wie Löschen oder Speichern eines Datensatzes über die TRAP-Funktion als Tastaturabfrage ausführen zu lassen.

Insgesamt verfügt TAS über eine sehr professionelle Sprache mit allen Folgen einer ausgezeichneten Leistungsfähigkeit, die nicht unbedingt leicht zu beherrschen ist.

Die notwendige Einarbeitungszeit in eine neue Programmsprache wird durch ein gut durchdachtes Handbuch deutlich herabgesetzt. Der TAS beiliegenden Loseblattsammlung im Ringbuchordner kann bescheinigt werden, daß sich ihre Autoren einer didaktischen Fragestellung zugunsten des Anwenders nicht entzogen hat. Bereits mit den Kapitelüberschriften (Lehrteil 1) wird dieser Anspruch deutlich. Inhaltlich wird er ihm auch gerecht.



Die Autoren führen zunächst den Anwender Schritt für Schritt in den Programmaufbau einer Adressverwaltung ein. Notwendige Steps werden durchgängig begründet und im Quellcode entwickelt. Dem User steht am Schluß dieses Kurses nicht nur eine fertige Adressverwaltung zur Verfügung, die der Anbieter übrigens als zusätzliches Bonbon als TAS-Anwendung mitliefert, viele der Standardroutinen dieses Programms lassen sich leicht auf andere Anwendungen übertragen.

Im weiteren Verlauf des Handbuches wird der Sprachumfang von TAS als Beschreibung der einzelnen Befehle abgehandelt. Dieser Teil fällt im Vergleich zu dbase II etwas mager aus. Der Mangel wird im Anhang des Buches allerdings durch kommentierte Quelltexte, die die Wirkung der einzelnen Befehle verdeutlichen sollen, ausgeglichen. Die Abrundung des Ganzen bildet die Beschreibung der Einsatzmöglichkeiten der vielen TAS Dienstprogramme, die der Anwender auf seiner Programmdiskette findet. Insgesamt ist das dbase II Handbuch zwar weitaus umfangreicher, in der didaktischen Konzeption aber wirkt es weniger überzeugend.

FAZIT:

Mit der Datenbank TAS erhält der Anwender ein Paket, das neben der eigentlichen Programmsprache einen Compiler, ein erstes Anwendungsprogramm und ein gut strukturiertes Handbuch enthält. Zum Ladenpreis von DM 198,– ist dieses Angebot ein ernstzunehmendes Konkurrenzprodukt zum Marktstandard dbase II. Der Tatsache, daß dbase II dem Heimanwender schnellere Programmiererfolge aufgrund des integrierten Editorkonzeptes und des weniger abstrakten Sprachaufbaus verspricht, hat TAS seine professionellere Konzeption entgegenzusetzen, die anfangs sicher einige Schwierigkeiten mit sich bringen wird, für den fortgeschrittenen Anwender aber eigentlich unverzichtbar erscheint. Deshalb kann TAS gerade dieser Käufergruppe ohne Einschränkungen empfohlen werden, zumal der große Bruder (TAS +) auch auf den IBM-kompatiblen Geräten zu finden ist.

Einsteigern dürfen wir allerdings einen gravierenden Mangel nicht verschweigen, der eher das Programmumfeld betrifft. In puncto weiterführende Literatur zum Programm sieht es sehr mager aus. Kein Vergleich zu dem, was zu dbase II angeboten wird.

CHECKER- der Eingabechecksummer für alle CPCs

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 BASIC-Zeile die Ausgabe einer vierstelligen Hexademizalzahl, die in eckige Klammern eingeschlossen ist. Diese Prüfsumme muß mit der im einzugebenden Listing enthaltenen Summe (am Ende jeder BASIC-Zeile) 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
```

XBC BASIC-COMPILER

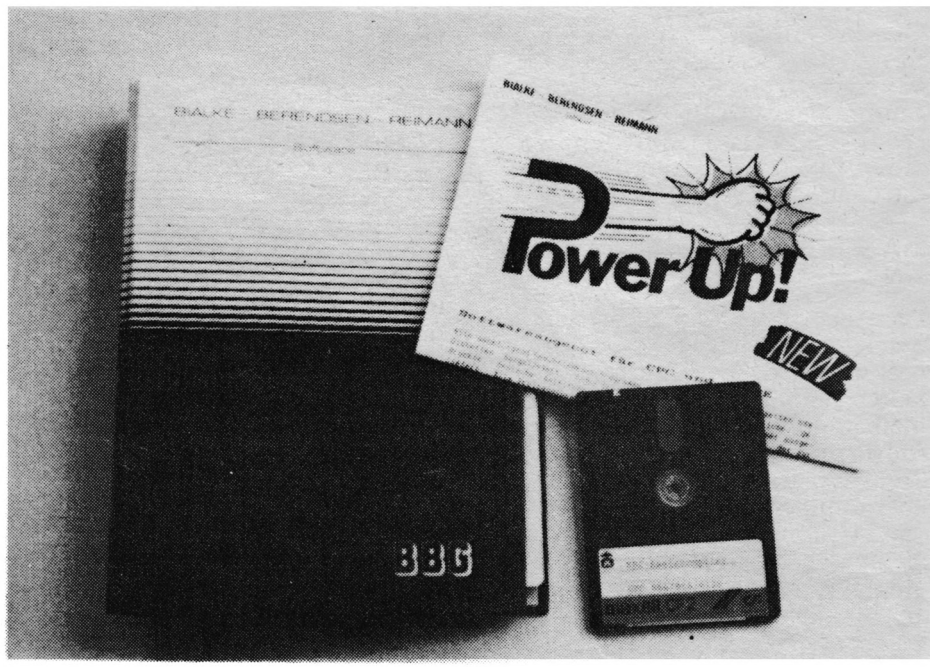
Bedingt einsatzbereit

Unter dem Namen Extended BASIC-Compiler (XBC) ist für die Schneider-CPC-Rechner ein Programm erhältlich, das den Eindruck erweckt, es könne Ihre Programme erheblich beschleunigen. Ob XBC hält, was sein Name verspricht, wollten wir genauer wissen und haben das Programm für Sie unter die Lupe genommen.

Die Aufmachung des XBC-Compilers macht einen semiprofessionellen Eindruck. In einem stabilen DIN-A5-Ordner erhalten Sie die Diskette nebst Handbuch. Letzteres besteht aus einer Sammlung fotokopierter Blätter; aber die Kopien sind wenigstens gut, und die Beschreibungen sind für erfahrene Programmierer leicht zu verstehen. Für Anfänger ist XBC wohl nicht das Wahre, und zwar nicht nur aufgrund des Handbuchs.

STARTSCHWIERIGKEITEN

Zunächst möchten Sie wohl den Compiler starten. Und da wartet auch schon die erste Hürde. Die Diskette wird im Vendor-Format ausgeliefert, so daß Sie erst einmal sämtliche Dateien auf eine Systemdiskette kopieren müssen. Das ist zwar selbstverständlich, sollte aber dennoch in der Beschreibung erwähnt werden. Wer sich mit CP/M noch wenig befaßt hat, dürfte hier Schwierigkeiten haben.
Mit welchem Befehl XBC gestartet wird, verschweigt das Handbuch dezent. Dafür beginnt es mit einer Liste der möglichen Fehlermeldungen. Das mag zwar interessant sein, aber nicht zu diesem Zeitpunkt. Nach einigem Suchen stoßen Sie auf die Information, der Compiler werde mit X gestartet. Ein Blick ins Directory der Diskette schafft Gewißheit: Auch das ist falsch. In einem Nachtrag zur CPC-Version ist schließlich zu erfahren, daß sich auf der Disk zwei Versionen befinden: XC für die 64KByte-Geräte und X6 für den 6128.
Die Befehle für den Compiler und den integrierten Editor sind quer durchs Handbuch verstreut. Die Dokumentation ist jedoch vollständig und enthält auch detaillierte Informationen wie zum Beispiel zur internen Darstellung von Variablen. Daß es auch die Möglichkeit gibt, ein Listing auszudrucken, finden Sie irgendwo gegen Ende des Buches.
Leider hat der Autor ein Stichwortverzeichnis für unnötigen Luxus gehalten; aufgrund des geringen Umfangs von nur fünfzig Seiten ist aber eine Suche im Inhaltsverzeichnis und anschließendes hektisches Blättern gerade noch akzeptabel. Das Ringbuchsystem erweist sich dabei als sehr nützlich. Sie können sich jederzeit die gerade benötigten Blätter auf dem Schreibtisch zurechtlegen.



Ein gutes Handbuch ergänzt die Software

EIN NEUES BASIC

Nachdem Sie den Compiler zum Laufen gebracht haben, wollen Sie wahrscheinlich schnell ein altes BASIC-Programm durchjagen, das Ihnen schon immer viel zu langsam war. Doch so einfach macht es Ihnen XBC nicht. Weder als BASIC- noch als ASCII-Version ließ sich eine bereits vorhandene Datei in den Editor laden. Aber selbst wenn das funktioniert hätte, wäre der Erfolg höchst zweifelhaft gewesen. XBC steht nicht umsonst für „Extended BASIC Compiler": Das Locomotive BASIC der Schneider CPCs ist für ihn das gleiche wie Chinesisch für einen Deutschen.
Die Sprache, die XBC versteht, heißt Extended BASIC und ist eine Eigenentwicklung des Autors. Extended BASIC arbeitet ohne Zeilennummern und verwendet ausschließlich Labels. Außer den bekannten Datentypen existieren noch Byte für Werte bis 255 und Cardinal für positive Ganzzahlen. Besonders störend macht sich bemerkbar, daß XBC eine Variable ohne besondere Kennzeichnung als Cardinal-Wert versteht, während die meisten BASIC-Dialekte den Typ Real, also eine Fließkommazahl, voraussetzen. Die dadurch nötige Umgewöhnung führt anfangs häufig zu Fehlern.

Der Befehl GOSUB wurde zwar implementiert, ist aber eigentlich unnötig, da Unterprogramme ähnlich wie in Pascal durch die Namen ihrer Labels aufgerufen werden können. Als Vorteil gegenüber anderen BASICs sind die zahlreichen Schleifenoperationen zu werten. REPEAT und WHILE gibt es ebenso wie FOR, das beim XBC auch mit DOWNTO funktioniert. Hier wurde eindeutig von Pascal abge-

kupfert, was sich aber keineswegs nachteilig auswirkt.
Einen besonderen Sinn sah der Autor wohl in dem Befehl LOOP, der eine Endlosschleife erzeugt. Ob dies wirklich nützlich ist, sei dahingestellt. Die Möglichkeit, das Programm über den WAIT-Befehl für eine bestimmte Zeitdauer zu unterbrechen, kann auf jeden Fall von Vorteil sein. Mit den herkömmlichen FOR-Schleifen können Sie die Zeitdauer nur grob abschätzen; WAIT erledigt das für Sie auf die Millisekunde genau.
Weitere Features sind etwa die Umwandlung zwischen zwei verschiedenen Datentypen oder die Bestimmung einer maximal zulässigen String-Länge in Array-Definitionen. Arithmetische Funktionen sind in der Regel mehrfach vorhanden, so daß Sie für jeden Datentyp eine eigene Funktion verwenden müssen. Wahrscheinlich versprach man sich dadurch einen Geschwindigkeitszuwachs, wie auch durch die Implementierung spezieller Funktionen für häufig benötigte Rechenvorgänge. So kann zum Beispiel x zum Quadrat mit einer eigenen Funktion SQR berechnet werden. Das kann wiederum zu Verwechslungen mit der SQR-Funktion in anderen BASIC-Versionen führen, wo diese Abkürzung bekanntlich für die Quadratwurzel einer Zahl steht.

MANGELNDE ANPASSUNG AN DEN CPC

Die Anpassung des XBC an den Schneider CPC ist leider nur mangelhaft erfolgt. Die vielfältigen Möglichkeiten für Sound und Textfenster bleiben ungenutzt, von den Grafikroutinen wurden nur einige in den Sprachschatz des Extended BASIC übernommen. Auch die Verwaltung von Dateien läuft völlig anders ab als gewohnt; dafür haben Sie die Möglichkeit, Dateien mit wahlfreiem Zugriff zu erzeugen.
Erfahrene Programmierer können die nicht implementierten Befehle durch die Einbindung von Assembler-Routinen simulieren. Um diese Option jedoch wirklich ausnutzen zu können, sollten Sie außer BASIC wenigstens die Grundzüge von Assembler beherrschen und den Aufbau der CPC-Firmware kennen.

DER EDITOR: ZEILENORIENTIERT, ABER KOMFORTABEL

Wenn Sie schon mit einem völlig neuen BASIC arbeiten müssen, so sollte der zugehörige Editor wenigstens bequem zu bedienen sein. Das Einlesen von einem beliebigen Textverarbeitungsprogramm ist zwar theoretisch möglich; in der Praxis stellen sich Ihnen jedoch diverse Hindernisse in den Weg, die darauf beruhen, daß XBC ein besonderes Format der Texte erwartet. In der Regel lohnt sich der Aufwand einer Dateiumwandlung nicht, so daß Sie auf den integrierten Editor zurückgreifen werden.
Dieser Editor arbeitet zeilenorientiert. Jede Zeile erhält folglich eine Nummer, die jedoch für den Compiler keine Bedeutung hat. Dieser arbeitet, wie bereits erwähnt, ausschließlich mit den von Ihnen vergebenen Labels.
Während des Editierens haben Sie Zugriff auf zahlreiche CP/M-Funktionen. So können Sie sich etwa zwischendurch ein Directory ansehen oder eine Datei umbenennen. Für einen Zeileneditor besitzt das Programm so angenehme Möglichkeiten wie das Verschieben ganzer Programmblöcke oder das Suchen nach einem bestimmten Ausdruck. Das Umbenennen einer Variablen im ganzen Programm ist ohne großen Aufwand möglich – ein Vorteil des Editor-Compiler-Systems gegenüber einem Interpreter.

COMPILIERTE FILES SELBSTÄNDIG LAUFFÄHIG

Ein einmal erstelltes Programm kann im Speicher oder auf Diskette compiliert werden. Auf dem CPC 6128 haben Sie etwa 31,5 KByte für Text und Compilat zur Verfügung; die 64KByte-Geräte bieten Ihnen nur knapp zehn KByte an. Bei Compilierung auf Diskette entspricht dieser freie Platz gleichzeitig der maximalen Größe Ihres BASIC-Programms.
Ihre Programme werden, wenn Sie die entsprechende Option anwählen, als COM-Version auf Diskette abgelegt. Mit dem Compiler erwerben Sie zugleich das Recht, die von Ihnen mit Hilfe dieses Systems erstellten Programme gewerblich zu nutzen. Allerdings haben sämtliche dieser COM-Files einen kleinen Schönheitsfehler: Ein vor dem eigentlichen Programm erscheinendes Titelbild weist Ihre Kunden darauf hin, daß das Programm mit dem XBC-Compiler erstellt wurde. Das müßte nicht sein und wurde von anderen Firmen auch eleganter gelöst, etwa durch eine versteckte Copyright-Notiz in der COM-Datei.
Die Runtime-Bibliothek von XBC ist ziemlich umfangreich. Ein Textprogramm, das nur aus einer Kommentarzeile und dem Befehl END bestand, nahm auf der Diskette ganze 17 KByte in Anspruch. Wenn man die geringe TPA des Betriebssystems CP/M bedenkt, die auch auf dem 6128 nur 48 KByte beträgt, ist das nicht gerade wenig.

HOHE GESCHWINDIGKEITEN

An Schnelligkeit schlägt XBC das normale Locomotive BASIC jedoch um Längen. Ein Kreis wird von XBC in nur sechs Sekunden auf den Bildschirm gebracht, während der normale Interpreter dazu 11,5 Sekunden benötigt. Einfache Rechenoperationen in einem zweitausend Zahlen großen Array bewältigte XBC in zwei Minuten und zwei Sekunden. Locomotive BASIC war hier nur um zehn Sekunden langsamer, was aber hauptsächlich an der Bildschirmausgabe von immerhin viertausend Zahlen gelegen haben dürfte. Der Berechnungsteil lief in XBC wesentlich schneller ab.
Seine wahre Stärke beweist der Compiler bei den transzendenten Funktionen. Die Aufgabe, zwanzigtausend verschiedene Werte der Funktion sin(x)*cos(x) zu berechnen, wurde in sechs Minuten und 37 Sekunden erledigt. Das normale BASIC benötigte dazu zehn Minuten und 21 Sekunden. Bei arithmetischen Operationen war der Compiler stets um einen Faktor von etwa 1,6 schneller als der Locomotive-Interpreter; beim Setzen einzelner Grafikpunkte betrug dieser Faktor sogar 1,9. Bildschirmausgaben im normalen Textmodus liefen dagegen nicht merklich schneller ab.

XBC: COMPILER FÜR SPEZIELLE ANWENDUNGEN

Die mangelnde Sound- und Grafik-Unterstützung machen XBC ungeeignet für effektreiche Programme wie etwa Spiele. Die Übernahme bereits vorhandener Programme können Sie getrost vergessen, da XBC im Grunde genommen kein BASIC-Compiler ist, sondern ein Compiler-BASIC mit eigener Programmiersprache und Entwicklungsumgebung.
Wenn Sie jedoch mathematische oder technische Probleme auf Ihrem CPC lösen wollen, Ihnen hohe Rechengeschwindigkeiten wichtig sind, und Sie außerdem keine Scheu davor haben, BASIC quasi neu zu erlernen, so ist der XBC-Compiler ein nützliches Hilfsmittel. Der Preis in Höhe von 99 Mark hält sich ebenfalls in erschwinglichem Rahmen.

AE□

Sprechen Sie „C“

Die Sprache „C“

auf CPC

und Joyce

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

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

WAS IST C EIGENTLICH?

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

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

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

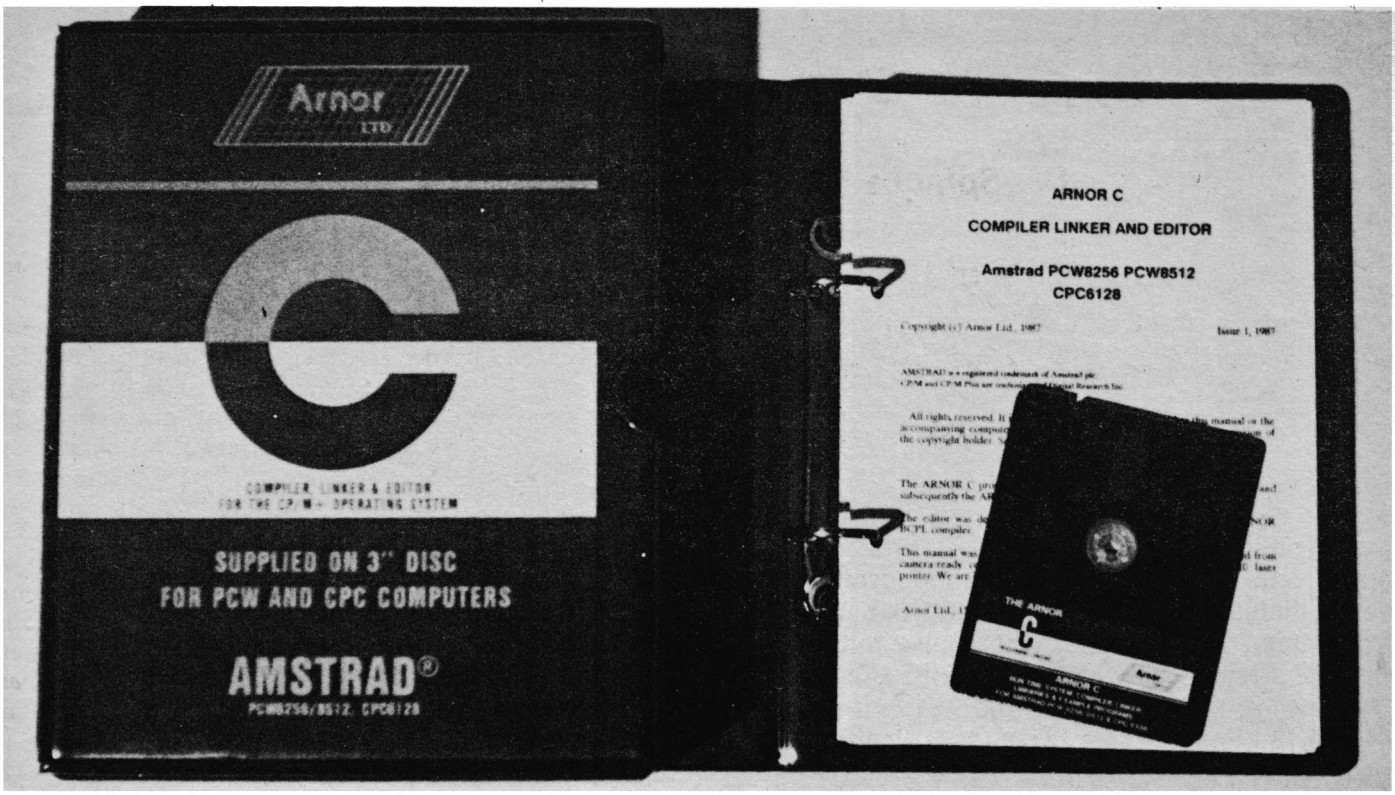
Die wichtigste Frage bei einem C-Compiler ist also immer, ob er sich an den Standard hält und inwieweit der komplette Sprachumfang nach Kernighan und Ritchie implementiert wurde. Kleine Unterschiede und Ergänzungen sind dabei aber nicht immer von Nachteil, sondern manchmal sogar sehr sinnvoll.

WAS IST ARNOR-C?

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

WAS LÄUFT BEI C AB?

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



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

ARNOR-C – EIN TEIL EINES ZWITTERS

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

Arnor wendet zur Speicherplatz-Einsparung also einen kleinen Trick an. Andererseits aber kann jemand, der diesen „Run"-Teil, der bei Arnor RUNC.COM heißt, nicht hat, auch nichts mit diesen Rumpf-C-Programmen anfangen. Die Weitergabe dieser Files ist deshalb meist sinn- und zwecklos. Weshalb wir in der Zwischenüberschrift von einem Zwitter geschrieben haben, erfahren Sie etwas später.

ERST HANDBUCH LESEN!

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

DER LIEFERUMFANG VON ARNOR-C

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

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

DAS LESEN VON README-FILES IST ÄUSSERST WICHTIG!

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

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

JETZT GEHT'S LOS!

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

TROTZ TRICK: DISKETTEN-SPEICHER TUT NOT!

Unsere ersten Tests erfolgten mit einem CPC6128 ohne externes Laufwerk. Da uns damit die Handhabung zu umständlich war, haben wir nach sehr kurzer Zeit dann eine Vortex-Station mit X-Modul angeschlossen. Damit ließ sich weit komfortabler arbeiten, und wir mußten nicht dauernd Disk-Jockey spielen. Über die Handhabung und den Ablauf bei der C-Programmierung zu schreiben, würde hier zu weit führen. C-Einsteiger dürfen sich aber nicht darüber wundern, daß die Vorgänge bis zur Erstellung des ablauffähigen Programmes viel Zeit in Anspruch nehmen können. Dies hat aber nichts mit Arnor oder dessen C-Paket zu tun, sondern ist nun einmal so, auch auf schnelleren und teureren Maschinen. Pascal- und auch Maschinensprache-Programmierer kennen dies wahrscheinlich schon.

DIE HAUPTTEILE VON ARNOR-C

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

DER ZWEITE TEIL DES ZWITTERS

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

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

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

ZUSAMMENFASSUNG ZU ARNOR-C:

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

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

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

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

◆ ◆ ◆

Bezugsquellen-Hinweis:

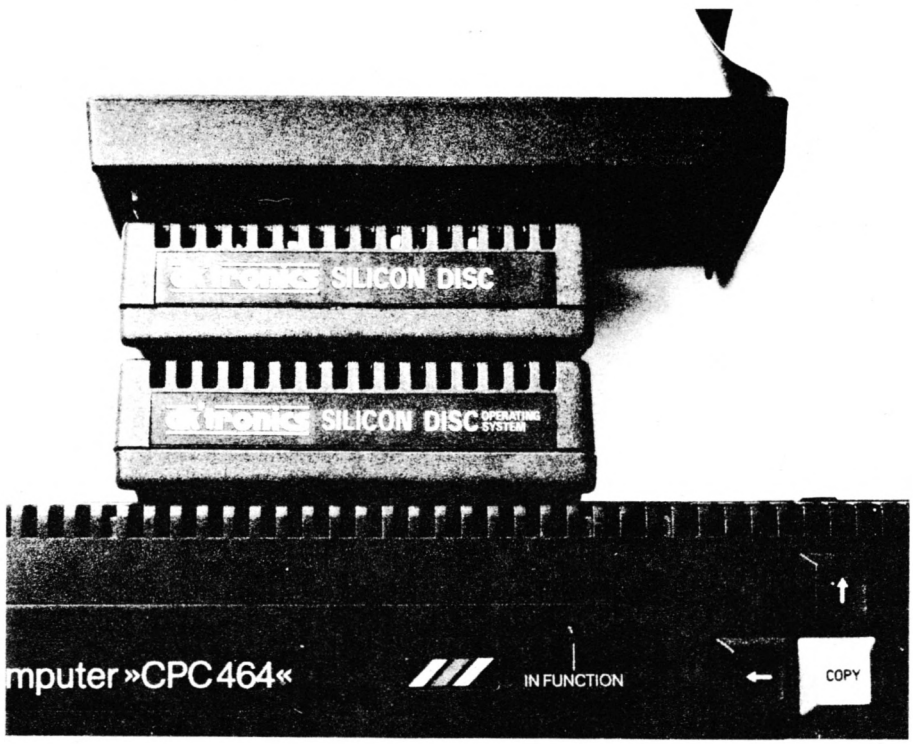
Arnor (Deutschland) Ltd.
Hans-Henny-Jahnn-Weg 21
2000 Hamburg 76

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.

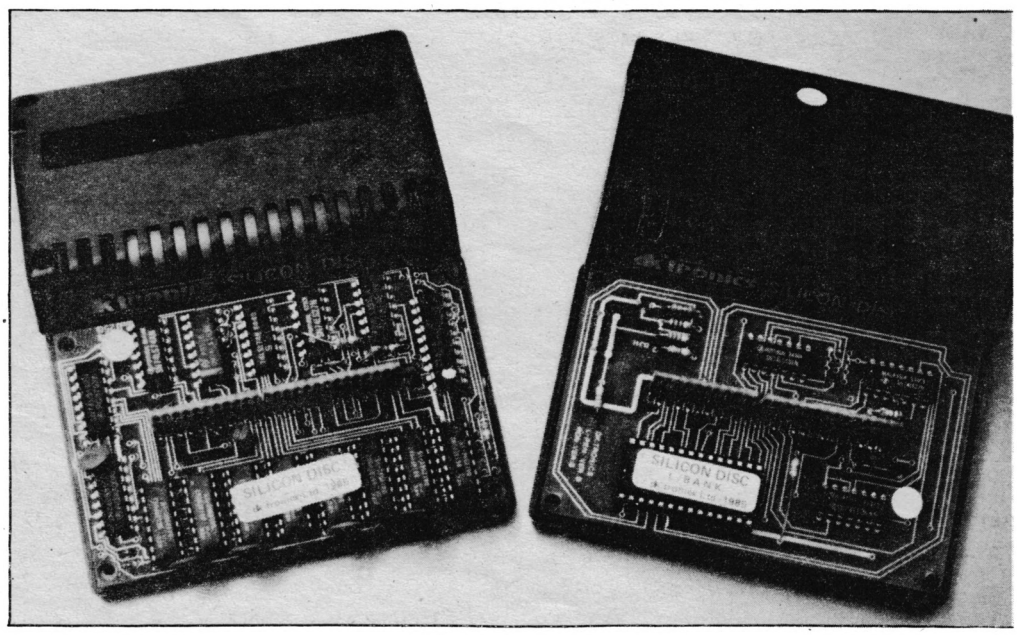
Nachdem der Inhalt einer Diskette übertragen wurde, muß die RAM-Disk mit dem SDISC-Befehl neu initialisiert werden. Dieser Hinweis im Handbuch ist uns beim ersten Durchlesen entgangen. Die Meldung "Drive B: disc missing" verursachte uns daher einiges Kopfzerbrechen. Das hätte sich wohl eleganter lösen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

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.

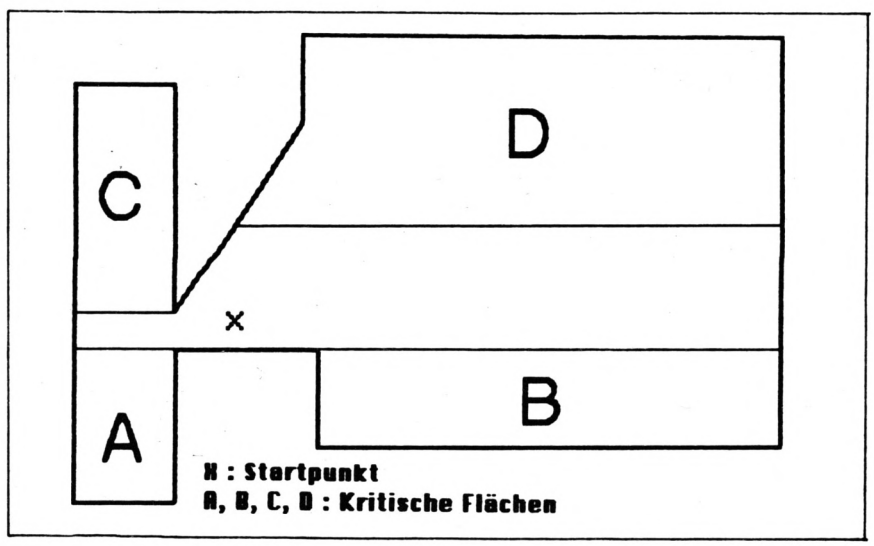


Abbildung 1
Diese Fläche soll ausgefüllt werden

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

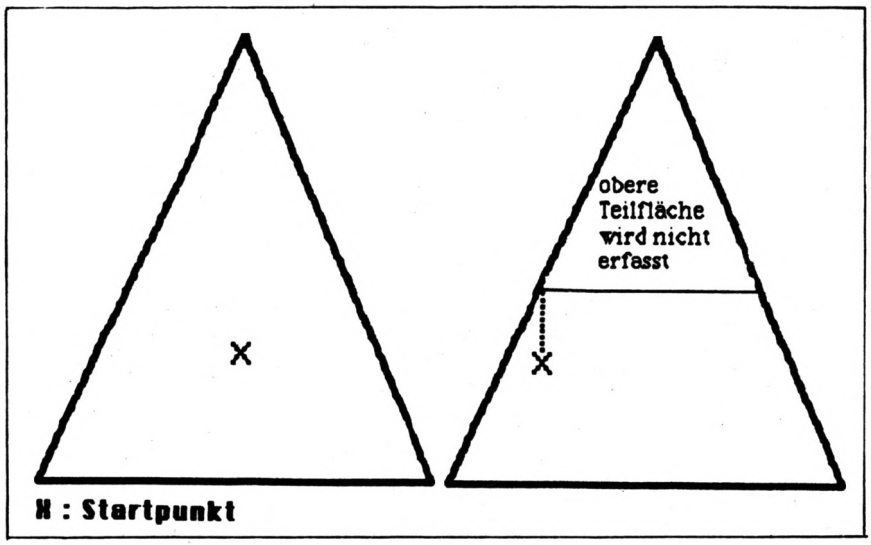


Abbildung 2
Die Wahl des richtigen Startpunktes ist entscheidend

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.

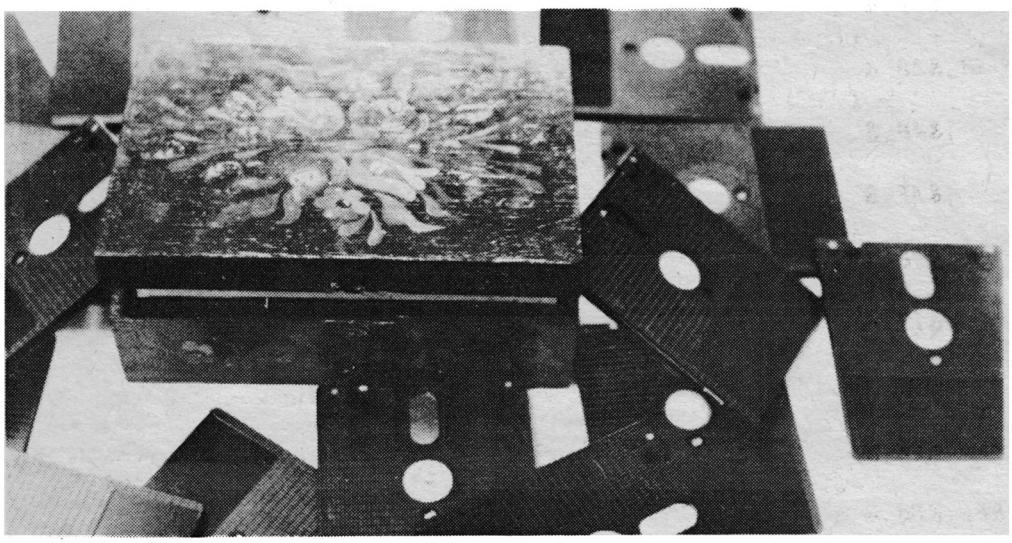
AE□

DRIVE A: DISC FULL

Ohne Ärger

Ab jetzt gibt es keinen Ärger mehr mit der allzu frühen "Disc Full"-Meldung bei Drei-Zoll-Disketten. Jeder CPC-Besitzer kann die Kapazität pro Disc-Seite um 30 KByte erweitern.

Welcher Schneider-Freak kennt nicht die Meldung "Disc Full" auf seiner sündhaft teuren Drei-Zoll-Diskette? 178 KByte sind eben doch noch nicht die Welt. Doch jetzt eilt Hilfe herbei: 208 KByte pro Diskettenseite lassen sich mit dem abgedruckten Programm von jedem CPC-Besitzer frei nutzen. Das sind somit pro Drei-Zoll-Diskette 60 KByte mehr. So etwas wirkt sich auch positiv auf Ihren Geldbeutel aus, sei er nun klein oder groß.
Mit Dutzenden von RSX-Befehlen brauchen die zusätzlichen Diskettenkapazitäten nicht angesprochen zu werden. Die ganz normalen BASIC-Befehle wie SAVE, LOAD oder CAT genügen. Alles andere macht das Maschinensprache-Programm. Es erkennt, ob es sich um eine normale CP/M- oder datenformatierte Diskette handelt oder ob Sie eine für 208 KByte vorbereitete verwenden.



Mit unserem Programm kann die Drei-Zoll-Diskette auf 208 KByte erweitert werden. Kein Ärger mehr mit "Disc Full".

Sie müssen nur Ihre Leerdisketten entsprechend formatieren. Aber auch das ist kein Problem, denn unser Formatierungsprogramm ist aufgrund seiner bequemen Menüsteuerung einfach zu bedienen.
Tippen Sie als erstes das abgedruckte Programm ab oder nutzen Sie unseren Diskettenservice. Speichern Sie das Formatierungsprogramm als BASIC-Programm mit der Befehlsfolge SAVE"FORMAT" ab. Falls sich beim Eintippen ein Fehler in den Datenzeilen eingeschlichen haben sollte, wird Ihnen das mitgeteilt. Es ist daher wichtig, das BASIC-Programm vollständig einzugeben, da sonst eine detaillierte Angabe über die Programmzeile, in der der Fehler aufgetreten ist, nicht korrekt erfolgen kann.
Mit Hilfe des Programms können Sie jetzt Ihre Leerdisketten auf 208 KByte pro Seite formatieren. Danach wird auf ihnen automatisch die Routine DOS 208 abgespeichert. Somit brauchen Sie später das Programm nicht lange zu suchen, da es sich auf jeder neu formatierten Diskette befindet.
Das DOS-208-Programm benötigt zudem nur 2 KByte auf Ihrer Disk. Falls Sie jedoch DOS 208 nicht gespeichert haben möchten, können Sie es problemlos mit ERA,"DOS208.BIN" löschen. Außerdem wird ein File namens "*==> DOS.208 0K" in das Inhaltsverzeichnis der Diskette eingetragen. Durch diesen Vermerk können Sie bei einem CATALOG leicht erkennen, ob es sich um eine normale Daten- oder CP/M-Diskette handelt oder um eine für 208 KByte vorbereitete Diskette. Ist eine 208-KByte-Diskette eingelegt, muß die DOS208-Routine aktiv sein. Dies erkennen Sie daran, daß bei einem CATALOG die Meldung "DOS 208 aktiv" erscheint.
Um eine Diskette vorzubereiten, verfahren Sie wie folgt:

- Starten Sie das abgedruckte Programm mit RUN.
- Als erstes müssen Sie angeben, in welchem Laufwerk Sie Ihre Leerdiskette formatieren möchten.
- Nachdem Sie zwischen Laufwerk A und B gewählt haben, legen Sie Ihre Leerdiskette ein und drücken die Taste "F".

Jetzt formatiert das Programm Ihre Leerdiskette auf 42 Spuren mit je 10 Sektoren à 512 Byte. Im normalen Daten- und CP/M-Format werden nur 40 Spuren à neun Sektoren genutzt, so daß Sie 30 KByte pro Diskettenseite mehr zur Verfügung haben.
Nach dem Formatieren speichert das Programm automatisch die Routine DOS208 ab. Bei dem abschließenden CATALOG erkennen Sie, daß sich jetzt auf Ihrer Diskette ein freier Speicherplatz von 207 K-Byte befindet, den Sie mit den üblichen BASIC-Befehlen LOAD, SAVE, OPENOUT und OPENIN frei nutzen können.

Die Initialisierung der DOS-208-Routine nach dem Einschalten des Computers geschieht auf folgende Art:

- Legen Sie eine Diskette ein, auf der sich die durch das Programm "FORMAT" abgespeicherte Routine DOS208.BIN befindet.
- Laden Sie das Maschinensprache-Programm mit LOAD"DOS208"
- Geben Sie den Befehl CALL &B000 ein, und Sie sehen die Einschaltmeldung von DOS208. Beachten Sie, daß diese Befehlsreihenfolge absichtlich so gewählt worden ist und kein BASIC-Lader für den

```
10 '*****************************  (2397)
15 '*        DOS 208            *  (23EB)
20 '*          VON              *  (239D)
25 '*      JUERGEN ROHLJE       *  (23C1)
30 '*          FUER             *  (234F)
35 '*   SCHNEIDER CPC WELT      *  (23CF)
40 '*    CPC 464/664/6128    JE*  (23FA)
45 '*****************************  (23DD)
50 REM                             (067B)
60 DATA &F3,&08,&D9,&F5,&7D,&32,&F
E,&B0, 1318                        (2C31)
70 DATA &FE,&4C,&28,&0D,&FE,&61,&2
8,&09, 783                         (2BCB)
80 DATA &FE,&70,&28,&05,&F1,&D9,&0
8,&FB, 1128                        (2CA7)
90 DATA &C9,&F1,&D9,&08,&FB,&E5,&C
5,&D5, 1557                        (2CFF)
100 DATA &F5,&3A,&FE,&B0,&FE,&70,&
20,&06, 1137                       (2C24)
110 DATA &21,&58,&B0,&CD,&0E,&B0,&
3A,&7D, 875                        (2BDD)
120 DATA &BE,&DF,&55,&B0,&3A,&4C,&
BE,&CB, 1201                       (2C78)
130 DATA &6F,&28,&3B,&2A,&42,&BE,&
ED,&4B, 820                        (2BBE)
140 DATA &7D,&BE,&0A,&FE,&00,&20,&
05,&22, 650                        (2B89)
150 DATA &FE,&B0,&18,&07,&11,&40,&
00,&19, 567                        (2B93)
160 DATA &22,&FE,&B0,&CD,&B9,&AF,&
21,&F4, 1306                       (2CBD)
170 DATA &B0,&3A,&EB,&B0,&BE,&20,&
0B,&2A, 920                        (2BD7)
180 DATA &FE,&B0,&11,&18,&00,&19,&
36,&00, 550                        (2B39)
190 DATA &18,&0C,&ED,&5B,&FE,&B0,&
01,&19, 820                        (2B1B)
200 DATA &00,&21,&E1,&AF,&ED,&B0,&
F1,&D1, 1296                       (2C2E)
210 DATA &C1,&E1,&C9,&2A,&FE,&B0,&
11,&16, 1130                       (2CB9)
220 DATA &00,&19,&56,&2A,&7D,&BE,&
5E,&21, 595                        (2B60)
230 DATA &EA,&B0,&E5,&DF,&FA,&AF,&
E1,&36, 1566                       (2CB0)
240 DATA &00,&06,&0A,&E5,&C5,&DF,&
FD,&AF, 1093                       (2C97)
250 DATA &C1,&E1,&D0,&3A,&51,&BE,&
23,&77, 1109                       (2C4F)
260 DATA &10,&F1,&C9,&28,&00,&03,&
07,&00, 508                        (2BDC)
270 DATA &D1,&00,&3F,&00,&C0,&00,&
10,&00, 480                        (2B49)
280 DATA &00,&00,&C1,&0A,&20,&32,&
E5,&02, 516                        (2BCF)
290 DATA &04,&00,&00,&FF,&63,&C7,&
07,&5D, 657                        (2B58)
300 DATA &C5,&07,&21,&C3,&3E,&22,&
7F,&BE, 845                        (2B5F)
310 DATA &3E,&AF,&32,&81,&BE,&21,&
18,&B0, 839                        (2BB3)
320 DATA &7E,&FE,&FF,&C8,&CD,&5A,&
BB,&23, 1352                       (2CF5)
330 DATA &18,&F6,&07,&0C,&18,&20,&
44,&4F, 492                        (2BF2)
340 DATA &53,&20,&32,&30,&38,&20,&
69,&73, 521                        (2B42)
350 DATA &74,&20,&69,&6E,&69,&74,&
69,&61, 786                        (2B5B)
360 DATA &6C,&69,&73,&69,&65,&72,&
74,&20, 796                        (2BA1)
370 DATA &18,&0A,&0D,&0A,&A4,&20,&
62,&79, 472                        (2BFF)
380 DATA &20,&4A,&75,&65,&72,&67,&
65,&6E, 752                        (2B00)
390 DATA &20,&52,&6F,&68,&6C,&6A,&
65,&2C, 688                        (2BF2)
400 DATA &20,&31,&39,&38,&37,&0A,&
FF,&30, 562                        (2B61)
410 DATA &C6,&07,&0A,&44,&4F,&53,&
20,&32, 527                        (2B49)
420 DATA &30,&38,&20,&61,&6B,&74,&
69,&76, 679                        (2B2F)
430 DATA &0A,&FF, 265              (1343)
440 dat=0:sz=0:dz=60               (1E83)
450 FOR adr=&AF3E TO &B067         (14AA)
460 READ byte:dat=dat+1            (1D5F)
470 sz=sz+byte                     (1894)
480 POKE adr,byte                  (147F)
490 IF dat<8 AND adr<&B067 THEN 53
0                                  (1DC1)
500 READ chksum                    (0F5A)
510 IF chksum<>sz THEN PRINT"Fehle
r in Zeile ";dz                    (2FFD)
520 dz=dz+10:sz=0:dat=0            (2433)
530 NEXT adr                       (0C56)
540 REM                            (0651)
550 REM Programmteil zum Formatier
en auf 208 KB                      (2E66)
560 REM Nach der Formatierung wird
 die Routine DOS208                (3486)
570 REM automatisch auf der eingel
egten Diskette abgespeichert       (3DD7)
580 REM                            (06A1)
590 POKE &BE7F,&C9                 (0DEC)
600 RESTORE 1060:POKE &BDEE,&C9    (1212)
610 GOSUB 1100                     (0907)
620 MODE 2:PRINT"Waehlen Sie das D
rive, indem Sie formatieren moecht
en":PRINT                          (4298)
630 PRINT" Drive A = Taste 'A'
     Drive B = Taste 'B'"          (388B)
640 a$=UPPER$(INKEY$)              (1016)
650 IF a$="A"THEN POKE &A8A0,10:PO
KE &A8A2,41:POKE &B0C8,0:GOTO 680  (2A1D)
660 IF a$="B"THEN POKE &A8E0,10:PO
```

```
KE &A8E2,41:POKE &B0C8,1:GOTO 680 (2A86)
670 GOTO 640 (0937)
680 PRINT:PRINT"Bitte legen Sie di
e Leerdiskette in Drive ";a$: (3A4D)
690 PRINT:PRINT"und druecken Sie d
ie Taste 'F' zum formatieren" (38B8)
700 b$=UPPER$(INKEY$):IF b$="F"THE
N GOTO 710 ELSE GOTO 700 (2541)
710 PRINT:PRINT"Bitte warten Sie e
inen Moment "; (2915)
720 FOR track=0 TO 41 (13E5)
730 POKE &B0CA,track (12E7)
740 x=&B0D7 (0DF6)
750 FOR n=0 TO 9 (0E68)
760 POKE x+n*4,track (1AB4)
770 NEXT n (0A8C)
780 PRINT"."; (0A28)
790 IF TRACK<39 THEN CALL &B0C7 EL
SE FOR H=1 TO 7:CALL &B0C7:NEXT H (2C25)
800 NEXT track (0E40)
810 PRINT:IF a$="A"THEN POKE &A700
,0 ELSE POKE &A700,1 (1F32)
820 REM (0682)
830 REM Eintrag des Hinweises "* =
==> Dos 208" ins Inhaltsverzeichni
s (44B5)
840 REM (06AA)
850 DATA &1E,&00,&16,&00,&0E,&C1,&
21,&B0,&A9,&DF,&D4,&B0,&C9,&4E,&C6
,&07 (4617)
860 RESTORE 850 (0933)
870 FOR adr=&B0C7 TO &B0D6 (142C)
880 READ byte:POKE adr,byte:NEXT A
DR (25C7)
890 FOR adr=&A9B0 TO &ABAF:POKE ad
r,&E5:NEXT adr (2833)
900 FOR adr=&A9B0 TO &A9CF:POKE ad
r,0:NEXT adr (26E5)
910 name$="*==> DOS208":IF a$="A"T
HEN POKE &B0C8,0 ELSE POKE &B0C8,1
(3316)
920 FOR adr=&A9B1 TO &A9BB:POKE ad
r,ASC(MID$(name$,adr-&A9B0,1)):NEX
T adr (40BF)
930 POKE &A9B9,&B2:CALL &B0C7:CALL
 &B000 (173D)
940 REM (0671)
950 REM Speichern der Routine Dos
208 auf die Diskette (351B)
960 REM (0699)
970 PRINT:PRINT"Speichern der Rout
ine DOS 208 !":SAVE"dos208",b,&AF3
E,&12A (4011)
980 PRINT:PRINT"Die Diskette ist j
etzt fuer 208 KB vorbereitet" (38BC)
990 CAT:PRINT (080E)
1000 PRINT:PRINT"Moechten Sie noch
 eine Diskette formatieren (J
/N)" (40C7)
1010 a$=UPPER$(INKEY$):IF a$="J"TH
EN RUN 540 ELSE IF a$="N"THEN CLS:
POKE &BDEE,195:NEW (365D)
1020 GOTO 1010 (098C)
1030 REM (0627)
1040 REM Routine zum Formatieren d
er Diskette (2B96)
1050 REM (064F)
1060 DATA &1E,&00,&16,&00,&0E,&C1,
&21,&D7,&B0,&DF,&D4,&B0,&C9,&52,&C
6,&07, 1782 (4C25)
1070 DATA &00,&00,&C1,&02,&00,&00,
&C6,&02,&00,&00,&C2,&02,&00,&00,&C
7,&02, 792 (4BB0)
1080 DATA &00,&00,&C3,&02,&00,&00,
&C8,&02,&00,&00,&C4,&02,&00,&00,&C
9,&02, 800 (4B0B)
1090 DATA &00,&00,&C5,&02,&00,&00,
&CA,&02, 403 (2BB9)
1100 dat=0:sz=0:dz=1060 (1F20)
1110 FOR adr=&B0C7 TO &B0FE (144C)
1120 READ byte:dat=dat+1 (1D8B)
1130 sz=sz+byte (18BE)
1140 POKE adr,byte (14AB)
1150 IF dat<16 AND adr<&B0FF THEN
1190 (1EB1)
1160 READ chksum (0F84)
1170 IF chksum<>sz THEN PRINT"Fehl
er in Zeile ";dz:END (3197)
1180 dz=dz+10:sz=0:dat=0 (245C)
1190 NEXT adr (0C81)
1200 RETURN (068A)
```

Maschinensprache-Teil verwendet wurde. Dieses Vorgehen gewährleistet, daß DOS208 zu jedem Zeitpunkt geladen und aktiviert werden kann, ohne daß Daten verloren gehen.
Falls Sie das Programm DOS208 auf eine Diskette kopieren möchten, auf der sich schon Programme befinden, benutzen Sie folgende Befehle:

- Diskette einlegen, die für 208 KByte durch das abgedruckte Formatierungsprogramm vorbereitet worden ist.
- LOAD"DOS208.BIN":CALL &B000
- Diskette im CP/M- oder Data-Format einlegen.
- SAVE"DOS208",b,&AF3E,&12A

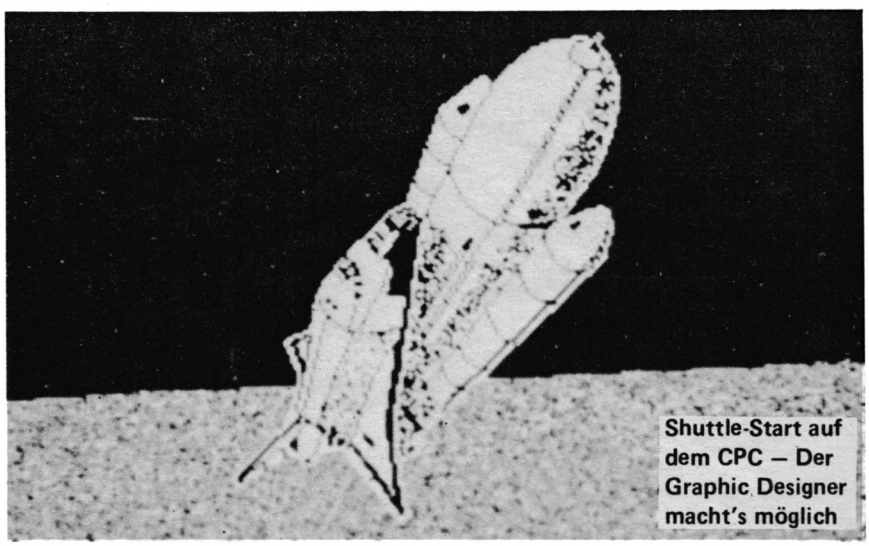
Sie können auch wieder eine CP/M- oder Datendiskette einlegen und sie bearbeiten, als wäre nichts geschiehen. Das Programm DOS208 verwaltet selbständig, falls eine für 208 KByte vorbereitete Diskette eingelegt worden ist, die zusätzlichen 30 KByte. DOS208 läuft sowohl mit einem als auch mit zwei Laufwerken. Sie können sogar einen CONTROL-SHIFT-RESET auslösen. Wenn Sie CALL &B000 eingeben, wird Ihr DOS208 wieder aktiviert.

Jürgen Rohlje□

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.

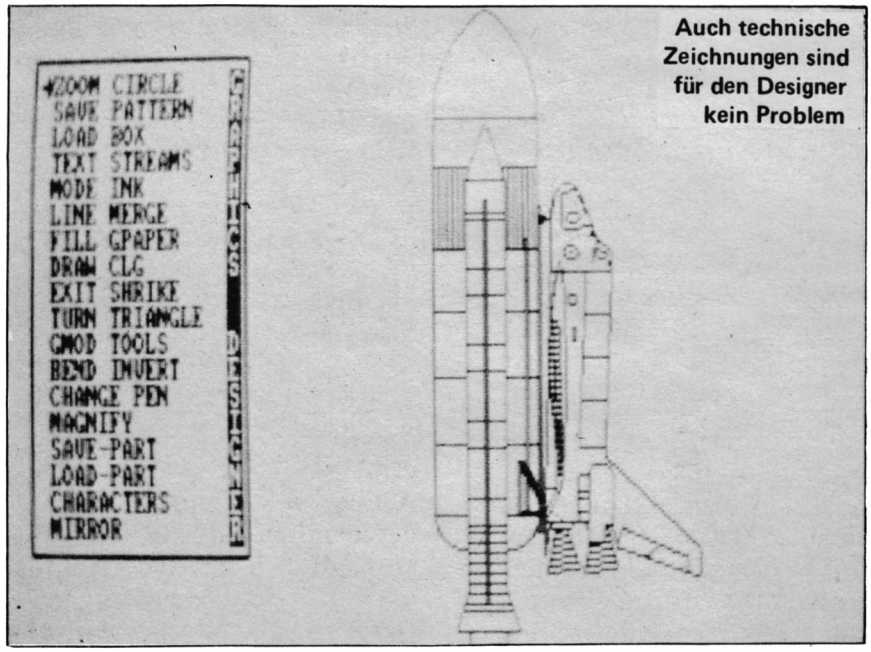
I. 3. Zeichensatz- und Pattern-Operationen

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.

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



Auch technische Zeichnungen sind für den Designer kein Problem

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

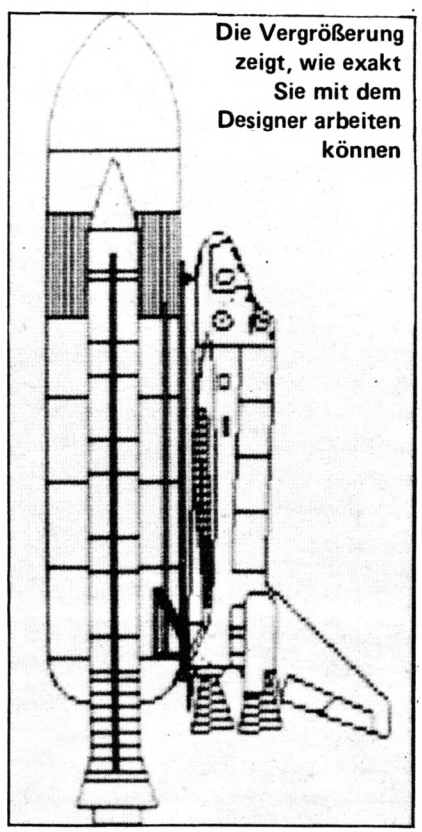
BEND: Diese Funktion „verbiegt" einen ganzen Bildschirminhalt. Es wird ein Effekt erzeugt, als wenn man ein Blatt Papier mit einem Bild verbiegen würde. BEND ist in zwei Unterfunktionen unterteilt: BEND und WAVES.
BEND: Hier wird der Bildschirminhalt einfach verbogen. Nach dem Aufruf der Funktion wird 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

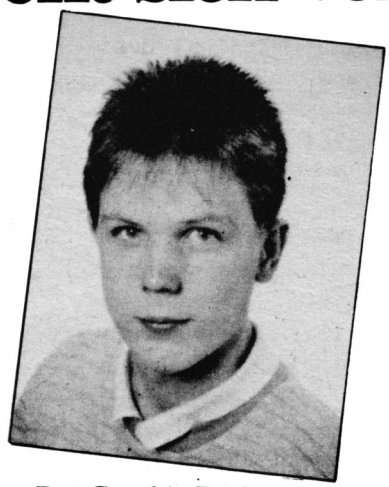
Der FONT-DESIGNER dient zum Erstellen beziehungsweise Verändern der Zeichensätze des GRAPHICS DESIGNER. 18 Zeichensätze sind bereits auf der Diskette enthalten, bis zu acht weitere können drei definiert werden. Diese Zeichensätze sind ebenfalls vom DESIGNER aus zu laden (vergleiche I. 3. CHARACTERS). 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.

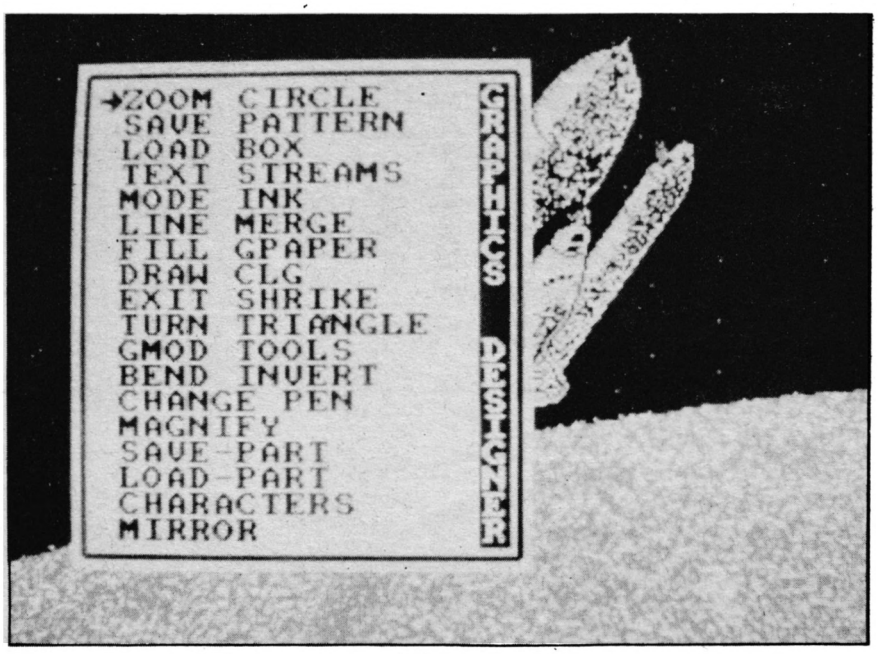
IV. Der Befehlssatz

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,il
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)
```

CPC Graphic Designer

```
**********************************
*    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 modus: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,modus-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$(num):IF NOT INKEY(
0)OR NOT INKEY(72)THEN num=num-1                (704A)
1140 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN num=num+1                                (29BE)
1150 IF NOT INKEY(66)THEN/OUT:GOTO
 1230                                           (1B2A)
1160 IF NOT INKEY(18)OR NOT INKEY(
76)THEN 1200                                    (1E7C)
1170 IF num<0 THEN num=lo                       (1EE7)
1180 IF num>lo THEN num=0                       (1EE4)
1190 GOTO 1130                                  (0AB8)
1200 RETURN                                     (0068A)
1210 /OUT:MODE modus:a=FRE(""):LOA
D b$(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$):/GPAPER,gpap:GOT
O 60                                            (2C08)
1310 /OUT                                       (0A3D)
1320 GOSUB 650:PLOT x,y,farbe:GOTO
 1320                                           (257A)
1330 GOSUB 2150:CLS                             (0CFF)
1340 INPUT"PEN";a:IF a<0 OR a>modu
st-1 THEN 1340                                  (3205)
1350 CLG a:GOTO 50                              (11AD)
1360 CLS                                        (06CF)
1370 INPUT"FACTOR(1-10)";f:IF f<1
OR f>10 THEN 1370                               (32A1)
1380 /OUT                                       (0AC9)
1390 GOSUB 2260:xd=x-xset+modi:yd=
y-yset+2:IF(xd*yd)/(2*(2\modus))>1
920 THEN 50                                     (64F8)
1400 /PART.IN,xset,yset,xd\modi,yd
\2,modus:LOCATE#2,1,1:PRINT#2,CHR$
(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:z=1:r=1:y=4                       (6370)
1480 nama$(d)="":c$=" ":FOR a=z TO
 z+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 z=1 THEN z=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                                 (3B9B)
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*modus:f 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$(32):IF NOT INKEY(0)OR NOT I
NKEY(72)THEN z=z-1               (53A9)
2220 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN z=z+1                     (2513)
2230 IF NOT INKEY(18)OR NOT INKEY(
76)THEN 2250                     (1EBF)
2240 z=MAX(1,z):z=MIN(4,z):GOTO 22
10                               (2A57)
2250 zmo=z-1:GOTO 60             (18AD)
2260 m=0:s=0                     (1298)
2270 IF m=0 THEN IF dar=1 THEN f1=
TEST(x,y):f2=TESTR(4,-4):f3=TESTR(
2,-2):f4=TESTR(2,-2):f5=TEST(x+2,y
):f6=TEST(x+4,y):f7=TEST(x,y-2):f8
=TEST(x,y-4)ELSE PRINT#2,"{A";:MOV
E x,0:DRAW x,400,farbe:MOVE 0,y:DR
AW 640,y:GOTO 2290 ELSE GOSUB 2320
:GOTO 2290                       (0751)
2280 modusf=4\modus:PLOT x,y,farbe
:PLOTR 4,-4,farbe:PLOTR 2,-2,farbe
:PLOTR 2,-2,farbe:PLOT x+2,y,farbe
:PLOT x+4,y,farbe:PLOT x,y-2,farbe
:PLOT x,y-4,farbe:IF farbe=f1 THEN
 PLOT x,y,(farbe+1)MOD modusf    (F1BD)
2290 BORDER farb(farbe)          (18A1)
2300 WHILE INKEY$="":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 00,00,00,00,00,00,00,00,18,1
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 <280F>
6,04,0A1                           <280F>
200 DATA 04,0C,00,18,11,03,18,11,0 <28AD>
3,1E,086                           <28AD>
210 DATA 00,18,11,01,19,01,03,1E,0 <289E>
0,00,065                           <289E>
220 DATA 18,18,00,18,18,00,00,00,1 <2810>
8,18,090                           <2810>
230 DATA 00,18,18,30,0C,18,30,60,3 <287A>
0,18,15C                           <287A>
240 DATA 0C,00,00,00,7E,00,00,7E,0 <289F>
0,00,108                           <289F>
250 DATA 60,30,18,0C,18,30,60,00,3 <28D3>
C,66,1FE                           <28D3>
260 DATA 66,0C,18,00,18,00,7C,C6,D <2813>
E,DE,3A0                           <2813>
270 DATA DE,C0,7C,00,00,00,1C,11,0 <2830>
1,1D,265                           <2830>
280 DATA 11,33,00,18,11,03,1A,11,0 <28A8>
3,7E,11C                           <28A8>
290 DATA 00,18,33,20,20,10,03,1E,0 <2851>
0,10,0CC                           <2851>
300 DATA 10,11,11,11,02,7C,00,1D,1 <28CD>
1,04,0F3                           <28CD>
310 DATA 1C,14,01,7F,00,1D,11,04,1 <2878>
C,14,112                           <2878>
320 DATA 08,78,00,19,33,20,20,15,0 <28B0>
1,1F,141                           <28B0>
330 DATA 00,11,11,01,19,11,11,33,0 <28D8>
0,07,098                           <28D8>
340 DATA 04,04,04,04,00,3F,00,03,0 <285C>
2,02,056                           <285C>
350 DATA 02,22,06,3C,10,11,13,06,1 <2860>
0,10,0C0                           <2860>
360 DATA 11,73,00,18,10,10,10,11,0 <28B2>
1,7F,15D                           <28B2>
370 DATA 00,01,01,01,29,29,21,63,0 <2888>
0,01,0DA                           <2888>
380 DATA 01,21,21,21,21,63,00,10,3 <284E>
0,21,149                           <284E>
390 DATA 21,03,06,1C,00,18,11,03,1 <28A0>
E,10,0A0                           <28A0>
400 DATA 00,78,00,10,30,21,21,21,0 <2872>
2,3B,158                           <2872>
410 DATA 00,18,11,03,16,10,11,73,0 <28CB>
0,18,0EE                           <28CB>
420 DATA 13,00,18,01,03,1E,00,25,2 <280F>
5,04,09B                           <280F>
430 DATA 04,04,00,1E,00,11,11,11,1 <28BB>
1,11,07B                           <28BB>
440 DATA 03,1E,00,11,11,11,11,03,0 <284D>
6,0C,07A                           <284D>
450 DATA 00,21,21,21,01,11,31,63,0 <28E6>
0,03,10C                           <28E6>
460 DATA 06,04,10,30,21,63,00,11,1 <28C4>
1,03,0F3                           <28C4>
470 DATA 06,04,00,1E,00,39,63,46,0 <28FB>
C,19,12F                           <28FB>
480 DATA 01,7F,3C,30,30,30,30,30,3 <28F4>
C,00,1E8                           <28F4>
490 DATA C0,60,30,18,0C,06,02,00,3 <2836>
C,0C,1C4                           <2836>
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7 <28F6>
E,18,156                           <28F6>
510 DATA 18,18,18,00,00,00,00,00,0 <2844>
0,00,048                           <2844>
520 DATA 00,FF,30,18,0C,00,00,00,0 <28F1>
0,00,153                           <28F1>
530 DATA 18,24,5A,42,5A,5A,24,00,7 <28CE>
C,82,2AE                           <28CE>
540 DATA 5A,44,5A,82,7 ,00,1C,22,4 <287C>
C,50,2D0                           <287C>
550 DATA 4C,22,1C,00,78,84,52,5A,5 <281A>
2,84,308                           <281A>
560 DATA 78,00,7C,82,5C,48,5C,82,7 <288D>
C,00,374                           <288D>
570 DATA 7C,82,5C,48,50,88,70,00,3 <2884>
C,42,368                           <2884>
580 DATA 9C,A4,BA,82,7C,00,24,5A,5 <2851>
A,42,412                           <2851>
590 DATA 5A,5A,24,00,38,44,28,28,2 <28B0>
8,44,210                           <28B0>
600 DATA 38,00,1C,22,14,54,B4,84,7 <2899>
8,00,28E                           <2899>
610 DATA 6C,92,44,48,44,92,6C,00,7 <285B>
0,00,3C4                           <285B>
620 DATA 50,54,5A,82,7C,00,44,AA,9 <283E>
2,82,3FE                           <283E>
630 DATA AA,BA,44,00,24,5A,4A,42,5 <2881>
2,5A,35E                           <2881>
640 DATA 24,00,3C,42,99,99,99,42,3 <28A6>
C,00,2EB                           <28A6>
650 DATA 7C,82,42,5C,50,88,70,00,3 <2805>
C,42,362                           <2805>
660 DATA 99,99,99,9D,41,3F,7C,82,4 <2889>
2,44,46C                           <2889>
670 DATA 52,8A,76,00,3E,41,5E,41,3 <2899>
D,41,2EE                           <2899>
680 DATA 3E,00,7C,82,6C,28,28,44,3 <283F>
8,00,274                           <283F>
690 DATA 24,5A,5A,5A,5A,42,3C,00,4 <286A>
4,AA,2F8                           <286A>
700 DATA AA,92,44,28,10,00,44,BA,A <280E>
A,82,3E2                           <280E>
710 DATA 92,AA,44,00,66,99,42,24,4 <2847>
2,99,3C0                           <2847>
720 DATA 66,00,44,AA,92,44,28,28,1 <28DD>
0,00,28A                           <28DD>
730 DATA 7E,81,72,24,4E,81,7E,00,0 <2814>
E;18,308                           <2814>
740 DATA 18,70,18,18,0E,00,18,18,1 <282A>
8,18,126                           <282A>
750 DATA 18,18,18,00,70,18,18,0E,1 <281A>
8,18,126                           <281A>
760 DATA 70,00,76,DC,00,00,00,00,0 <28B5>
0,00,1C2                           <28B5>
770 DATA CC,33,CC,33,CC,33,CC,33,0 <2511>
0,3FC                              <2511>
780 dat=0:sz=0:dz=10               <1E65>
790 FOR adr=&A000 TO &A300         <1400>
800 READ byte$:byte=VAL("&"+byte$) <35FB>
:dat=dat+1                         <35FB>
810 sz=sz+byte                     <183D>
820 POKE adr,byte                  <142A>
830 IF dat<10 AND adr<&A300 THEN 8 <1EEA>
70                                 <1EEA>
840 READ chksum$:chksum=VAL("&"+ch <2BC0>
ksum$)                             <2BC0>
850 IF chksum<>sz THEN PRINT"FEHLE <30EA>
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 <28EA>
,18,030                            <28EA>
20 DATA 18,18,18,00,18,00,6C,6C,6C <282A>
,00,1A4                            <282A>
30 DATA 00,00,00,00,6C,6C,FE,6C,FE <2889>
,6C,3AC                            <2889>
40 DATA 6C,00,18,3E,58,3C,1A,7C,18 <28E4>
,00,204                            <28E4>
50 DATA 00,C6,CC,18,30,66,C6,00,38 <28D6>
,6C,3AA                            <28D6>
60 DATA 38,76,DC,CC,76,00,18,18,30 <2843>
,00,32C                            <2843>
70 DATA 00,00,00,00,0C,18,30,30,30 <28B3>
,18,0CC                            <28B3>
80 DATA 0C,00,30,18,0C,0C,0C,18,30 <28B7>
,00,0C0                            <28B7>
90 DATA 00,66,3C,FF,3C,66,00,00,00 <28D8>
,18,25B                            <28D8>
100 DATA 18,7E,18,18,00,00,00,00,0 <28B8>
0,00,0C6                           <28B8>
110 DATA 00,18,18,30,00,00,00,7E,0 <2877>
0,00,0DE                           <2877>
120 DATA 00,00,00,00,00,00,00,18,1 <285D>
8,00,030                           <285D>
130 DATA 06,0C,18,30,60,C0,80,00,7 <2844>
C,C6,33C                           <2844>
140 DATA CE,D6,E6,C6,7C,00,18,38,1 <2870>
8,18,44C                           <2870>
150 DATA 18,18,7E,00,3C,66,06,3C,6 <28C6>
0,66,258                           <28C6>
160 DATA 7E,00,3C,66,06,1C,06,66,3 <28B9>
C,00,1EA                           <28B9>
170 DATA 1C,3C,6C,CC,FE,0C,1E,00,7 <28B9>
E,62,398                           <28B9>
180 DATA 60,7C,06,66,3C,00,3C,66,6 <284F>
0,7C,302                           <284F>
190 DATA 66,66,3C,00,7E,66,06,0C,1 <2822>
8,18,22E                           <2822>
200 DATA 18,00,3C,66,66,3C,66,66,3 <28FF>
C,00,264                           <28FF>
210 DATA 3C,66,66,3E,06,66,3C,00,0 <28D1>
0,00,1EE                           <28D1>
220 DATA 18,18,00,18,18,00,00,00,1 <2810>
8,18,090                           <2810>
230 DATA 00,18,18,30,0C,18,30,60,3 <287A>
0,18,15C                           <287A>
240 DATA 0C,00,00,00,7E,00,00,7E,0 <289F>
0,00,108                           <289F>
250 DATA 60,30,18,0C,18,30,60,00,3 <28D3>
C,66,1FE                           <28D3>
260 DATA 66,0C,18,00,18,00,7C,C6,D <2813>
E,DE,3A0                           <2813>
270 DATA DE,C0,7C,00,00,1E,21,42,7 <2814>
E,84,39D                           <2814>
280 DATA 84,00,00,1E,11,22,3C,44,F <2814>
8,00,24D                           <2814>
290 DATA 00,1E,22,40,40,84,78,00,0 <28F4>
0,1E,1DA                           <28F4>
300 DATA 11,22,22,44,F8,00,00,1F,1 <285E>
1,20,1E1                           <285E>
310 DATA 38,42,FE,00,00,1F,11,20,3 <2875>
8,40,240                           <2875>
320 DATA F0,00,00,1E,22,40,4C,84,7 <2878>
8,00,2B8                           <2878>
330 DATA 00,22,22,44,7C,88,88,00,0 <284B>
0,3E,252                           <284B>
340 DATA 08,10,10,20,F8,00,00,1F,0 <2841>
2,04,165                           <2841>
350 DATA 44,88,78,00,00,32,14,28,3 <28C1>
0,48,22A                           <28C1>
360 DATA C8,00,00,20,20,40,40,80,F <2867>
C,00,304                           <2867>
370 DATA 00,22,36,5C,44,88,88,00,0 <2839>
0,24,22C                           <2839>
380 DATA 24,58,58,90,90,00,00,1C,2 <2807>
2,44,276                           <2807>
390 DATA 44,88,70,00,00,3E,11,22,3 <2811>
C,40,229                           <2811>
400 DATA F0,00,00,1C,22,44,54,88,7 <2888>
8,00,2C6                           <2888>
410 DATA 00,3E,11,22,3C,44,F4,00,0 <28A8>
0,1C,201                           <28A8>
420 DATA 24,40,38,88,70,00,00,FE,1 <28E0>
0,20,2C2                           <28E0>
430 DATA 20,40,40,00,00,22,22,44,4 <28CD>
4,88,1F4                           <28CD>
440 DATA 70,00,00,22,22,44,84,98,6 <2828>
0,00,274                           <2828>
450 DATA 00,22,22,44,54,A8,D8,00,0 <288C>
0,44,2A0                           <288C>
460 DATA 48,30,50,88,88,00,00,44,4 <284F>
4,88,2E8                           <284F>
470 DATA 70,20,F8,00,00,7E,84,08,1 <2875>
0,62,304                           <2875>
480 DATA FC,00,3C,30,30,30,30,30,3 <28BE>
C,00,264                           <28BE>
490 DATA C0,60,30,18,0C,06,02,00,3 <2836>
C,0C,1C4                           <2836>
500 DATA 0C,0C,0C,0C,3C,00,18,3C,7 <28F6>
E,18,156                           <28F6>
510 DATA 18;18,18,00,00,00,00,00,0 <2844>
0,00,048                           <2844>
520 DATA 00,FF,30,18,0C,00,00,00,0 <28F1>
0,00,153                           <28F1>
530 DATA 00,00,3C,66,7E,66,66,00,0 <2816>
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,C0,00,00,6
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,00,00,1C,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 18,18,18,3E,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,04,00,1
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,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
20                                (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 00,00,00,00,00,00,00,00,18,1
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                          (280B)
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 G3M.LAD      * (255B)
3 '*        ERZEUGT G3M.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=12                    (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"G3M.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*  (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,00,0C,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 G3O.LAD      * (256B)
3 '*        ERZEUGT G3O.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"G3O.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 7F,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
```

CPC Graphic Designer

```
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,A0,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 l=0 TO 3:inki(l)=PEEK(addr
(ad)+&3FFC+l):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 l=0 TO 3:INK l,inki(l):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
```

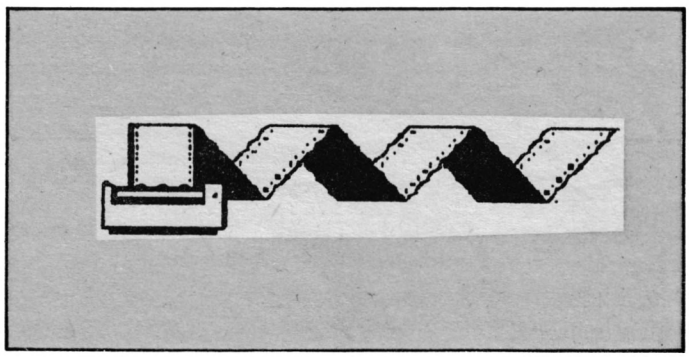
CPC Graphic Designer

```
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
$(24)                               (4412)
370 x1=11:x2=23:y1=8:y2=16          (2790)
380 GOSUB 610                       (09E2)
390 a=2                             (0B5A)
400 FOR n=1 TO 4                    (0EA8)
410 LOCATE 2,n*2:PRINT funk$(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:ux=1:uy=
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$(143);:PLOT rsx+2*n,rsy
-2*i,2:PEN 1                        (47BC)
960 NEXT                            (0645)
970 NEXT                            (0659)
980 PEN 2:rsx=147:rsy=111:rsy=rsy+
16                                  (2CEE)
990 farb=TEST(ux,uy)                (1C58)
1000 LOCATE x,y                     (0F21)
1010 PEN 1:PRINT CHR$(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:ux=ux+16:rsx=rsx+2:IF
 x>8 THEN x=8:ux=113:rsx=161        (61A8)
1050 IF NOT INKEY(8)OR NOT INKEY(7
4)THEN x=x-1:ux=ux-16:rsx=rsx-2:IF
 x<1 THEN x=1:ux=1:rsx=147          (6092)
1060 IF NOT INKEY(0)OR NOT INKEY(7
2)THEN y=y-1:uy=uy+16:rsy=rsy+2:IF
 y<1 THEN y=1:uy=113:rsy=127        (6144)
1070 IF NOT INKEY(2)OR NOT INKEY(7
3)THEN y=y+1:uy=uy-16:rsy=rsy-2:IF
 y>8 THEN y=8:uy=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(ux,uy)               (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)
```

MINI-DATEI

Das Datei-verwaltungs-programm für alle CPCs



Unser Leser Markus Wildi hat ein Dateiprogramm geschrieben, das sich mit einigen professionellen Programmen dieser Art durchaus vergleichen lassen kann. Abtippen lohnt sich!

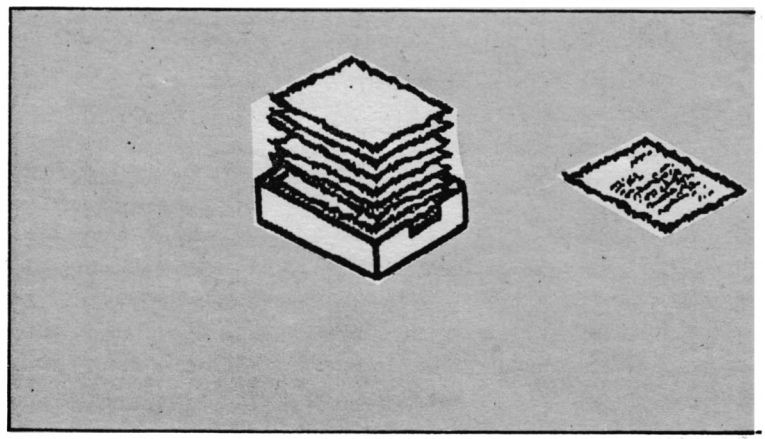
Die mit „Mini-Datei" angelegten Daten können von keinem anderen Dateiverwaltungsprogramm geladen werden!
Bevor Sie das Programm starten, sollten Sie eine leere, im Data-Format formatierte Diskette bereit halten. Sie werden Sie später beim Abspeichern brauchen.
Um Ihnen die einzelnen Menüpunkte und Funktionen zu erklären, erstellen wir eine Adressenverwaltung. Zuerst stehen Ihnen drei Hauptmenüs zur Auswahl:
1. Programm
2. Diskette
3. Sonstiges.

Um eine Datei zu erstellen, in unserem Fall eine Adressenverwaltung, wählen Sie als erstes Menü 3 Sonstiges. Es erscheint ein Hilfsmenü mit folgenden Funktionen:
– Datei laden
– Datei anlegen
– Funktionstasten
– Datum
– Farben
– Eingabemaske
– Druckmaske
– Drucker
– Feldlänge
– Feldanzahl
– Ende.

Wählen Sie nun (mit Cursortasten + Copy) die Funktion – Eingabemaske –. Es wird die Eingabemaske einer Programmverwaltung angezeigt. Da wir jedoch eine Adressenverwaltung erstellen möchten, ändern Sie die Eingabemaske wie folgt:

Programmverwaltung		– Adressenverwaltung
1.	Programmname	1. Anrede
2.	Hersteller	2. Name
3.	Datenträger	3. Vorname
4.	Preis	4. Straße/Nummer
5.	Bestellnummer	5. Plz/Ort
6.	Computer	6. Telefon
7.	Art	7. Beruf
8.	Monitor	8. Bemerkung
9.	Steuerung	9. –
10.	Bewertung	10. –

Es stehen Ihnen also zehn Eingabefelder zur Verfügung. Da wir nicht, wie in der Programmverwaltung, zehn Felder benötigen, sondern nur acht, wählen Sie nun die Funktion – Feldanzahl –.
Bestimmen Sie mit den Cursortasten die Feldanzahl. In unserem Fall wählen Sie die Anzahn von acht Feldern.
Eingabe beendet? (ESC)
Da Sie wieder in Menü 3 sind, wählen Sie nochmals die Funktion – Eingabemaske –. Sie sehen jetzt, daß sich das Programm auf die acht Felder beschränkt hat.



(ESC) verläßt Funktion und kehrt zu Menü 3 zurück.
Die Funktion – Feldlänge – ist genau so zu bedienen wie die Funktion – Feldanzahl –, jedoch kann hier die Feldlänge von jedem einzelnen Feld bestimmt werden.
(ESC) wenn Eingabe beendet!

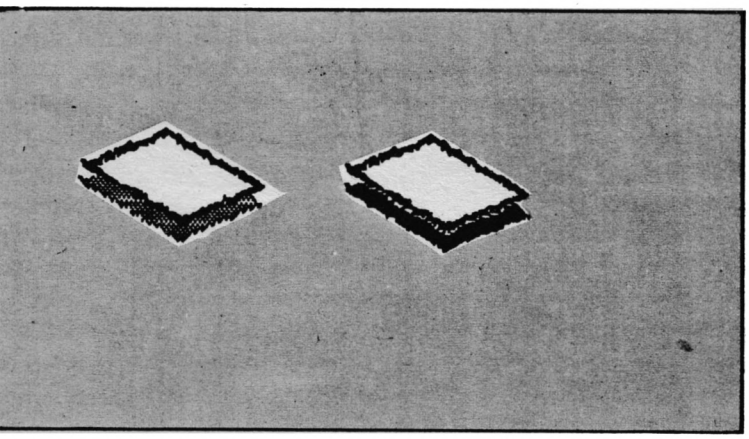
Druckmaske:

Mit dieser Funktion können Sie die Reihenfolge der ersten fünf Felder für den Listen- oder Einzeldruck bestimmen. Mehr dazu später.
Bevor Sie nun Adressen eingeben, sollten Sie eine Datei eröffnen beziehungsweise anlegen. Wählen Sie dazu die Funktion – Datei anlegen –. Geben Sie den Namen Ihrer Datei ein, zum Beispiel: Adressen.Dat, und wählen Sie anschließend Anlegen.
Achtung! Beachten Sie die Anweisung:
Bitte legen Sie eine leere Datendiskette ein . . .
und drücken Sie die Enter-Taste, da sonst beim Anlegen einer beschriebenen Diskette vorhandene Daten gelöscht werden können! Pro angelegte Datei benötigen Sie eine getrennte Datendiskette!
Der Rechner ermittelt die maximalen Datensätze, die auf einer Diskettenseite Platz haben, von selbst. Sobald die Datei angelegt ist, kehrt das Programm zu Menü 3 zurück.
Wenn Sie Ihre Datei später einmal laden möchten, so wählen Sie die Funktion – Datei laden –.

Geben Sie ein: Adressen.Dat und wählen Sie anschließend – Laden –.
Jetzt können Sie Adressen eingeben. Wählen Sie dazu Menü 1 Programm . Es erscheint wieder ein Hilfsmenü mit folgenden Funktionen:
– Eingeben
– Drucken
– Ändern
– Löschen
– Sortieren
– Suchen
– Weiter
– Zurück
– Nummer.

Eingeben:

Es erscheint die von Ihnen erstellte Eingabemaske. Geben Sie nun Ihre persönlichen Daten ein. Möchten Sie weitere Adressen eingeben, so wählen Sie die Funktion – Weiter –. Um Ihnen das Eintippen der Adressen zu erleichtern, stehen Ihnen die Funktionstasten zur Verfügung. (Menü 3: Funktion – Funktionstasten –).



Zurück/Nummer:

Mit – Zurück – blättern Sie eine Adresse zurück. Wählen Sie – Nummer –, so können Sie eine bestimmte Adresse aus dem gesamten Datensatz abrufen.

Suchen:

Wieder erscheint die komplette Eingabemaske. Fahren Sie mit den Cursortasten zum Beispiel auf Name und drücken Sie Copy. Tippen Sie nun den zu suchenden Namen ein, zum Beispiel Huber. Wenn Sie danach die Taste "S" drücken, wird der Rechner alle Adressen suchen, die Huber enthalten. Möchten Sie jedoch alle Adressen, die Huber heißen und in der Südstraße wohnen, so geben Sie beim Feld Straße einfach Südstraße ein . . . Hat der Rechner die gesuchte Adresse gefunden, so meldet er sich mit einem weiterem Hilfsmenü:

Weiter – Menü – Drucken:

Haben Sie sich für –Weiter– entschieden, so sucht der Rechner nach weiteren möglichen Adressen. Bei – Drucken – wird die gefundene Adresse ausgedruckt. (Falls Ihr Drucker nicht empfangsbereit sein sollte, so wählen Sie Menü 3: Funktion – Drucker –.) Bei – Menü – wird der Suchvorgang gestoppt und ins Menü 1 zurückgesprungen.

Sortieren:

Wieder wird die Eingabemaske angezeigt. Wählen Sie mittels Cursortasten + Copy, nach welchem Kriterium sortiert werden soll. Eine weitere Frage erscheint, bei der Sie angeben müssen, zwischen welchen Adressennummern ausgelesen werden soll. Achtung: Die Sortierroutine kann, je nach Anzahl der Datensätze, einige Zeit brauchen; haben Sie Geduld! Für eine volle Diskette werden 24 Stunden und mehr benötigt!

Drucken:

Nun möchten Sie sicher Ihre Daten zu Papier bringen. Wählen Sie die Funktion – Drucken –. Sie können nun die augenblickliche Adresse (Einzeldruck), oder eine Liste (Liste) ausdrucken. Bei der Liste wird noch zusätzlich gefragt, welche Adressen gedruckt werden sollen.

Ändern:

Steuern Sie mit den Tasten (Tab) und (Caps Lock) auf das zu ändernde Feld. Wählen Sie danach mit den Cursortasten + Copy die Funktion – Ändern –, um das gewählte Feld zu korrigieren.

Löschen:

Die augenblickliche Adresse wird gelöscht.

Nun zu Menü 2, dem Diskettenmenü: (ESC, dann Menü 2 wählen):
– Löschen
– Umbenennen
– Starten
– Laden
– Katalog
– User
– Laufwerk.

Löschen:

Es wird nach dem zu löschenden File gefragt.

Umbenennen:

Zuerst wird nach dem alten, dann nach dem neuen Filenamen gefragt. Sollte das gefragte File nicht auf Diskette vorhanden sein, so wird ins Menü zurückgesprungen.

Starten/Laden:

Das gesuchte Programm wird von Diskette gestartet beziehungsweise geladen.
Wichtig: Sollte das gesuchte File nicht auf Diskette vorhanden sein, so wird das Programm unterbrochen, und Sie müssen „Mini-Datei" von neuem starten.

Katalog:

Der Disketteninhalt wird angezeigt.

User:

Die Usernummer (0 bis 16) kann mit den Cursortasten bestimmt werden.

Laufwerk:

Bei dieser Funktion kann zwischen Laufwerk A und B gewechselt werden. Es ist also das Arbeiten mit zwei Laufwerken möglich. *Markus Wildi*□

```
100 '*****************************  (2234)
110 '*   STARTPROGRAMM FUER     *  (2229)
120 '*         MINIDATEI        *  (22F6)
130 '*            VON           *  (2275)
140 '*       MARKUS WILDI       *  (2276)
150 '*           FUER           *  (2228)
160 '*   SCHNEIDER CPC-WELT     *  (2247)
170 '*    CPC 464/664/6128   JE*  (2256)
180 '*****************************  (22D4)
1000 CALL &BBFF:CALL &BB00          (0ED4)
1010 POKE &BDEE,&C9                 (0D54)
1020 MODE 2:BORDER 0:INK 0,0:INK 1
,18                                 (159B)
1030 SYMBOL AFTER 256:MEMORY &8D87
-1                                  (115F)
1040 LOAD"MC1",&A000                (0FB4)
1050 CALL &A000                     (09E8)
1060 LOAD"MC2",&A690                (0FF9)
1070 IF PEEK(6)=&7B THEN GOSUB 126
0 ELSE IF PEEK(6)=&91 THEN GOSUB 1
310                                 (25AA)
1080 SYMBOL AFTER 64                (092E)
1090 SYMBOL 64,60,64,60,66,60,2,60
                                    (1C1C)
1100 SYMBOL 123,198,0,120,12,124,2
04,118,0                            (1E3D)
1110 SYMBOL 125,198,0,102,102,102,
102,62                              (1C45)
1120 SYMBOL 124,198,0,60,102,102,1
02,60                               (1C1F)
1130 SYMBOL 255,255,85,85,85,85,85
,255,255                            (201F)
1140 SYMBOL 126,120,198,198,252,19
8,198,248,192                       (20A2)
1150 SYMBOL 91,219,60,102,102,126,
102,102,0                           (1F96)
1160 SYMBOL 93,198,0,198,198,198,1
98,60,0                             (1E66)
1170 SYMBOL 92,198,56,198,198,198,
198,124,0                           (1F96)
1180 KEY DEF 28,1,123,91:KEY DEF 2
6,1,125,93:KEY DEF 29,1,124,92      (2B44)
1190 KEY DEF 25,1,126,45:KEY DEF 7
1,1,121,89:KEY DEF 43,1,122,90      (2B75)
1200 KEY DEF 25,1,126,63:KEY DEF 2
4,1,94,96:KEY DEF 22,1,60,62        (2BF4)
1210 KEY DEF 31,1,46,58:KEY DEF 19
,1,35,39:KEY DEF 18,1,13,1,30       (2DA8)
1220 KEY DEF 57,1,51,64:KEY DEF 39
,1,44,59:KEY DEF 32,1,48,61         (2BDF)
1230 KEY DEF 17,1,43,42:KEY DEF 41
,1,55,47:KEY DEF 9,1,224,248        (2A0F)
1240 KEY DEF 79,1,127,250:KEY DEF
30,1,45,95:KEY DEF 66,0,0,0,0       (2BA0)
1250 RUN"mdatei"                    (0E67)
1260 POKE &A6A1,&1:POKE &A6A2,&C:P
OKE &A6AA,&1B                       (1F21)
1270 POKE &A6B8,&1:POKE &A6B9,&C:P
OKE &A6C1,&1B                       (1F42)
1280 POKE &A6D2,&1:POKE &A6D3,&C:P
OKE &A6DB,&1B                       (1FB2)
1290 POKE &A6EA,&4E:POKE &A6E6,&E0
:POKE &A6ED,&97:POKE &A6F2,&E0      (28A0)
1300 RETURN                         (0654)
1310 POKE &A6A1,&5:POKE &A6A2,&C:P
OKE &A6AA,&1F                       (1F24)
1320 POKE &A6B8,&5:POKE &A6B9,&C:P
OKE &A6C1,&1F                       (1F48)
1330 POKE &A6D2,&5:POKE &A6D3,&C:P
OKE &A6DB,&1F                       (1FB6)
1340 POKE &A6EA,&52:POKE &A6E6,&E4
:POKE &A6ED,&9B:POKE &A6F2,&E4      (28E5)
1350 RETURN                         (06B8)

100 '*****************************  (2234)
110 '* MINIDATEI-HAUPTPROGRAMM *  (22B4)
120 '*            VON          *  (2261)
130 '*       MARKUS WILDI      *  (2262)
140 '*           FUER          *  (2214)
150 '*    SCHNEIDER CPC-WELT   *  (22E3)
160 '*     CPC 464/664/6128  JE*  (222B)
170 '*****************************  (22C0)
1000 |TAPE:OPENOUT"D":MEMORY HIMEM
-1:CLOSEOUT:|DISC                  (1FA2)
1010 DIM mo$(11)                    (0F75)
1020 x0=23:y1=8:y2=13:y3=8:y4=10:y
5=9:y6=9:y7=14:y8=8:y9=15:z9=1:y10
=8:z10=1:y11=10:y12=18:y13=12:y14=
15:x15=36:y16=8                     (AE2D)
1030 s$=STRING$(80,154):recr$=SPAC
E$(250):datei$="PROGRAMM":in$=CHR$
(24)                                (43D3)
1040 nummer=1:stift=18:ta=29:mo=9:
jahr=1986:nummermax=706             (4A31)
1050 feldl=250:felda=10:lo=5:datum
$="29. OKTOBER   1986"              (422C)
1060 FOR a=0 TO 9:fu$(a)=CHR$(48+a
)+SPACE$(9):feldl(a)=25:NEXT        (4065)
1070 FOR a=0 TO 4:druck(a)=a:NEXT   (242B)
1080 RESTORE 1200                   (09DC)
1090 FOR a=0 TO 9:READ a$:k$(a)=a$
+SPACE$(14-LEN(a$)):NEXT            (3609)
1100 FOR a=0 TO 8:READ a$:p$(a)=a$
+SPACE$(9-LEN(a$)):NEXT             (358A)
1110 FOR a=0 TO 2:READ a$:m$(a)=a$
:NEXT                               (2680)
1120 FOR a=0 TO 6:READ a$:disc$(a)
=a$+SPACE$(10-LEN(a$)):NEXT         (3905)
1130 FOR a=0 TO 10:READ a$:so$(a)=
a$+SPACE$(15-LEN(a$)):NEXT          (38BC)
1140 FOR a=0 TO 2:READ a$:fa$(a)=a
$+SPACE$(6-LEN(a$))+":":NEXT       (3ACA)
1150 FOR a=0 TO 2:READ a$:da$(a)=a
$+SPACE$(11-LEN(a$))+":":NEXT      (3BC6)
1160 FOR a=0 TO 1:READ a$:dl$(a)=a
$+SPACE$(11-LEN(a$))+":":NEXT      (3BCF)
```

```
1170 FOR a=0 TO 1:READ a$:drucker$
(a)=a$+SPACE$(8-LEN(a$)):NEXT      <3B7F>
1180 FOR a=0 TO 11:READ a$:mo$(a)=
a$+SPACE$(9-LEN(a$)):NEXT          <370F>
1190 FOR a=0 TO 2:READ a$:sum$(a)=
a$++SPACE$(7-LEN(a$)):NEXT         <384D>
1200 DATA "PROGRAMMNAME","HERSTELL
ER","DATENTRIGER","PREIS","BESTELL
NUMMER","COMPUTER","ART","MONITOR"
,"STEUERUNG","BEWERTUNG"           <7B65>
1210 DATA "EINGEBEN","DRUCKEN","[N
DERN","L\SCHEN","SORTIEREN","SUCHE
N","WEITER","ZURICK","NUMMER"      <5E52>
1220 DATA "PROGRAMM","DISKETTE","S
ONSTIGES"                          <2839>
1230 DATA "L\SCHEN","UMBENENNEN","
STARTEN","LADEN","CATALOG","USER",
"LAUFWERK"                         <4BF2>
1240 DATA "DATEI LADEN","DATEI ANL
EGEN","FUNKTIONSTASTEN","DATUM","F
ARBEN","EINGABEMASKE","DRUCKMASKE"
,"DRUCKER","FELDL[NGE","FELDANZAHL
","ENDE"                           <8DBB>
1250 DATA "PAPIER","RAND","STIFT"  <1E05>
1260 DATA "DATEINAME","DATENSCTZE"
,"ANLEGEN"                         <2910>
1270 DATA "DATEINAME","LADEN"      <1A2C>
1280 DATA "NLQ 401","DMP 2000"     <1B20>
1290 DATA "JANUAR","FEBRUAR","M[RZ
","APRIL","MAI","JUNI","JULI","AUG
UST","SEPTEMBER","OKTOBER","NOVEMB
ER","DEZEMBER"                     <71E5>
1300 DATA "WEITER","MEN]","DRUCKEN
"                                  <20FF>
1310 MODE 2                        <07EE>
1320 PRINT s$" (c) BY MARKUS WILDI
 JUN.      M I N I - D A T E I"SPC
(11)datum$s$                       <4F39>
1330 LOCATE 23,5:PRINT"PROGRAMM
   DISKETTE      SONSTIGES"        <334C>
1340 WINDOW#2,12,56,10,24          <14F6>
1350 GOSUB 6700                    <092F>
1360 INK 1,stift                   <10CF>
1370 LOCATE x0,5:PRINT in$m$(z0)in
$                                  <2472>
1380 FOR w=1 TO 10:NEXT            <115C>
1390 IF INKEY(1)=0 AND x0<49 THEN
x0=x0+14:z0=z0+1:LOCATE x0-14,5:PR
INT m$(z0-1):GOTO 1370             <5358>
1400 IF INKEY(8)=0 AND x0>23 THEN
x0=x0-14:z0=z0-1:LOCATE x0+14,5:PR
INT m$(z0+1):GOTO 1370             <5366>
1410 IF INKEY(9)=0 THEN 1430       <116F>
1420 GOTO 1390                     <0991>
1430 IF x0=23 THEN i=13:GOSUB 7330
:GOTO 1470                         <2089>
1440 IF x0=37 THEN i=9:GOSUB 7330:
GOTO 3390                          <1F53>
1450 IF x0=51 THEN i=9:GOSUB 7330:
GOTO 3960                          <1F98>
1460 GOTO 1380                     <0991>
1470 IF open=0 THEN GOSUB 6890:GOT
O 1380                             <19D4>
1480 IF feldl<>fl OR felda<>fa THE
N feldl=fl:felda=fa:GOSUB 6700     <46CF>
1490 FOR a=0 TO felda-1            <1799>
1500 IF feldl(a)<>feldlsi(a)THEN f
eldl(a)=feldlsi(a)                 <4540>
1510 NEXT                          <0693>
1520 recr$=SPACE$(fl)              <16A7>
1530 WINDOW#2,63,77,7,24           <1368>
1540 MOVE 502,296:DRAW 592,296:DRA
W 592,134:DRAW 502,134:DRAW 502,29
6                                  <2FAC>
1550 FOR a=0 TO 8:LOCATE 65,8+a:PR
INT p$(a):NEXT                     <27B0>
1560 IF z2>felda THEN z2=felda:y2=
8+felda                            <34D8>
1570 nu=(nummer-1)*(feldl+2):/RECR
EAD,@nu,@recr$:GOSUB 6760          <44A3>
":
LOCATE 14,y2:PRINT in$k$(z2)":"in$
                                   <4559>
1590 '                             <077D>
1600 IF INKEY(2)=0 AND y1<16 THEN
y1=y1+1:z1=z1+1:LOCATE 65,y1-1:PRI
NT p$(z1-1):GOTO 1580              <520A>
1610 IF INKEY(0)=0 AND y1>8 THEN y
1=y1-1:z1=z1-1:LOCATE 65,y1+1:PRIN
T p$(z1+1):GOTO 1580               <5177>
1620 IF INKEY(70)=0 AND y2<13+feld
a-1 THEN y2=y2+1:z2=z2+1:LOCATE 14
,y2-1:PRINT k$(z2-1)":":GOTO 1580  <6187>
1630 IF INKEY(68)=0 AND y2>13 THEN
 y2=y2-1:z2=z2-1:LOCATE 14,y2+1:PR
INT k$(z2+1)":":GOTO 1580          <5680>
1640 IF INKEY(9)=0 THEN i=9:GOSUB
7330:ON z1+1 GOSUB 1670,1790,2220,
2300,2350,2720,3180,3240,3300      <468A>
1650 IF INKEY(66)=0 THEN CLS#2:CLO
SEIN:GOTO 1390                     <198E>
1660 GOTO 1590                     <09BC>
1670 LOCATE 14,y2:PRINT k$(z2)":":
dat$=""                            <28C7>
1680 FOR a=0 TO felda-1:LOCATE 30,
13+a:PRINT SPACE$(25):NEXT         <2DEA>
1690 FOR a=0 TO felda-1            <172A>
1700 LOCATE 14,13+a:PRINT in$k$(a)
":"in$SPC(feldl(a)+1)              <3C81>
1710 LOCATE 30,13+a:laenge=feldl(a
):mini=32:maxi=126:GOSUB 6540      <449A>
1720 dat$=dat$+e$+SPACE$(feldl(a)-
LEN(e$)):d$(a)=e$+SPACE$(feldl(a)-
LEN(e$))                           <5F90>
1730 LOCATE 14,13+a:PRINT k$(a)":"
                                   <1FF0>
```

```
1740 NEXT                                  (0660)
1750 nu=(nummer-1)*(feldl+2)               (2509)
1760 /RECWRITE,@nu,@dat$                   (1E04)
1770 LOCATE 14,y2:PRINT in$k$(z2)"
:"in$                                      (289C)
1780 RETURN                                (0615)
1790 IF druckon=0 THEN GOSUB 7340:
RETURN                                     (19D4)
1800 WINDOW#3,33,47,10,13:CALL &A6
90,3,&9310:CLS#3                           (2361)
1810 MOVE 264,248:DRAW 368,248:DRA
W 368,200:DRAW 264,200:DRAW 264,24
8                                          (2C99)
1820 LOCATE 35,11:PRINT in$"EINZEL
DRUCK"in$:e=2                              (2BB2)
1830 LOCATE 35,12:PRINT"LISTE"             (1499)
1840 FOR w=1 TO 10:NEXT w                  (15CD)
1850 IF INKEY(2)=0 THEN LOCATE 35,
11:PRINT"EINZELDRUCK":LOCATE 35,12
:PRINT in$"LISTE       "in$:e=1            (4AF6)
1860 IF INKEY(0)=0 THEN LOCATE 35,
12:PRINT"LISTE       ":LOCATE 35,11
:PRINT in$"EINZELDRUCK"in$:e=2             (4A48)
1870 IF INKEY(9)=0 THEN IF e=1 THE
N GOSUB 1900:CALL &A693,4,&953A EL
SE von=nummer:bis=von:such=1:GOSUB
 1960                                      (542D)
1880 IF INKEY(66)=0 THEN CALL &A69
3,3,&9310:i=66:GOSUB 7330:RETURN           (2830)
1890 GOTO 1840                             (095D)
1900 WINDOW#4,28,58,22,24:CALL &A6
90,4,&953A:CLS#4                           (23D6)
1910 MOVE 222,54:DRAW 460,54:DRAW
460,26:DRAW 222,26:DRAW 222,54             (29DF)
1920 LOCATE 30,23:PRINT"DRUCKEN VO
N ";:laenge=5:mini=48:maxi=57:GOSU
B 6540                                     (4328)
1930 von=VAL(e$):IF von>nummermax
OR von=0 THEN LOCATE 42,23:PRINT S
PC(5):GOTO 1920                            (4433)
1940 LOCATE 42,23:PRINT USING"####
#";von;:PRINT" BIS ";:GOSUB 6540           (2C0B)
1950 bis=VAL(e$):IF bis<von OR bis
>nummermax THEN LOCATE 52,23:PRINT
 SPC(5):GOTO 1940                          (4938)
1960 such=1:feldldr=0:FOR a=0 TO l
o-1:feldldr=feldldr+feldl(druck(a)
):NEXT                                     (5C68)
1970 IF e=2 THEN 2080                      (10AD)
1980 PRINT#8,"PROGRAMM: MINI-DATEI
     DATEI: "datei$".DAT      AUSDR
UCK: "datum$"      NUMMERN:"von"-"b
is                                         (6EAE)
1990 PRINT#8,"----------------------
     -------"STRING$(LEN(datei$+".
DAT"),"-")"      ----------"STRING$
(LEN(datum$),"-")"      "STRING$(10
+LEN(STR$(von))+LEN(STR$(bis)),"-"
)                                          (9A61)
2000 PRINT#8                               (0814)
2010 IF lo=1 THEN b=0:GOTO 2060 EL
SE bi=lo-2                                 (2874)
2020 FOR b=0 TO bi                         (1273)
2030 IF feldl(b)<=LEN(k$(druck(b))
)THEN PRINT#8,LEFT$(k$(druck(b)),f
eldl(b))":";:GOTO 2050                     (62F3)
2040 PRINT#8,k$(druck(b))SPACE$(fe
ldl(b)-LEN(k$(druck(b))))":";              (4CA7)
2050 NEXT                                  (06CE)
2060 IF feldl(b)<=LEN(d$(druck(b+1
)))THEN PRINT#8,LEFT$(k$(druck(b))
,feldl(b))ELSE PRINT#8,k$(druck(b)
)                                          (7576)
2070 PRINT#8,STRING$(feldldr+4,"="
)                                          (1D7C)
2080 FOR a=von TO bis                      (182B)
2090 IF INKEY(66)=0 THEN GOSUB 703
0:IF jn=1 THEN RETURN                      (1E60)
2100 nu=(a-1)*(feldl+2):/RECREAD,@
nu,@recr$                                  (3AA8)
2110 GOSUB 6760                            (09FD)
2120 IF lo=1 THEN d=0:GOTO 2160            (1946)
2130 FOR d=0 TO bi                         (1259)
2140 PRINT#8,d$(druck(d))SPACE$(fe
ldl(druck(d))-LEN(d$(druck(d))))":
";                                         (56F8)
2150 NEXT                                  (0697)
2160 PRINT#8,d$(druck(d))SPACE$(fe
ldl(druck(d))-LEN(d$(druck(d))))           (524C)
2170 PRINT#8,STRING$(feldldr+4,"-"
)                                          (1D46)
2180 IF INKEY(66)=0 THEN GOSUB 703
0:IF jn=0 THEN CALL &A693,4,&953A:
RETURN                                     (2940)
2190 NEXT                                  (06E7)
2200 such=0                                (0EDC)
2210 RETURN                                (0672)
2220 LOCATE 30,y2:laenge=feldl(z2)
:mini=32:maxi=126:GOSUB 6540               (43BF)
2230 IF e$=""THEN LOCATE 30,y2:PRI
NT d$(z2):RETURN                           (2672)
2240 d$(z2)=e$+SPACE$(feldl(z2)-LE
N(e$))                                     (3237)
2250 LOCATE 30,y2:PRINT d$(z2)             (1B77)
2260 dat$="":FOR a=0 TO felda-1:da
t$=dat$+d$(a):NEXT                         (3C19)
2270 nu=(nummer-1)*(feldl+2)               (251B)
2280 /RECWRITE,@nu,@dat$                   (1E17)
2290 RETURN                                (0613)
2300 FOR a=0 TO felda-1:LOCATE 30,
13+a:PRINT SPACE$(feldl(a)):d$(a)=
"":NEXT                                    (4759)
2310 dat$=SPACE$(feldl)                    (18DE)
2320 nu=(nummer-1)*(feldl+2)               (2580)
2330 /RECWRITE,@nu,@dat$                   (1E7C)
2340 RETURN                                (0678)
```

```
2350 WINDOW#3,4,20,7,8+felda:CALL
&A690,3,&8F1F:CLS#3                    (294E)
2360 MOVE 30,298:DRAW 154,298:DRAW
 154,280-(felda*16):DRAW 30,280-(f
elda*16):DRAW 30,298                   (48F2)
2370 FOR a=0 TO felda-1:LOCATE 6,8
+a:PRINT k$(a):NEXT                    (2F99)
2380 IF z3>felda THEN z3=felda:y3=
8+felda                                (34BB)
2390 LOCATE 6,y3:PRINT in$k$(z3)in
$CHR$(8)" "                            (2C87)
2400 FOR w=1 TO 10:NEXT                (1158)
2410 IF INKEY(2)=0 AND y3<7+felda
THEN y3=y3+1:z3=z3+1:LOCATE 6,y3-1
:PRINT k$(z3-1):GOTO 2390              (5975)
2420 IF INKEY(0)=0 AND y3>8 THEN y
3=y3-1:z3=z3-1:LOCATE 6,y3+1:PRINT
 k$(z3+1):GOTO 2390                    (5095)
2430 IF INKEY(9)=0 THEN 2460           (11AB)
2440 IF INKEY(66)=0 THEN CALL &A69
3,3,&8F1F:i=66:GOSUB 7330:RETURN       (2826)
2450 GOTO 2400                         (0941)
2460 WINDOW#4,23,55,18,20:CALL &A6
90,4,&957F:CLS#4                       (23A6)
2470 MOVE 182,118:DRAW 434,118:DRA
W 434,90:DRAW 182,90:DRAW 182,118      (29D7)
2480 LOCATE 25,19:PRINT"SORTIEREN
VON ";:laenge=5:mini=48:maxi=57:GO
SUB 6540                               (4596)
2490 von=VAL(e$):IF von>nummermax
OR von=0 THEN LOCATE 39,19:PRINT S
PC(5):GOTO 2480                        (44AB)
2500 LOCATE 39,19:PRINT USING"####
#";von;:PRINT" BIS ";:GOSUB 6540       (2C7A)
2510 bis=VAL(e$):IF bis<von OR bis
>nummermax THEN LOCATE 49,19:PRINT
 SPC(5):GOTO 2500                      (49A6)
2520 LOCATE 49,19:PRINT USING"####
#";bis                                 (1C52)
2530 mi=1:FOR a=0 TO z3-1:mi=mi+fe
ldl(a):NEXT                            (3946)
2540 GOSUB 6940                        (0903)
2550 FOR a=von TO bis                  (18D7)
2560 nu1=(a-1)*(feldl+2):dat1$=SPA
CE$(feldl):/RECREAD,@nu1,@dat1$        (513B)
2570 FOR b=a+1 TO bis                  (18B1)
2580 IF INKEY(66)=0 THEN GOSUB 712
0:IF jn=0 THEN 2660                    (20F4)
2590 nu2=(b-1)*(feldl+2):dat2$=SPA
CE$(feldl):/RECREAD,@nu2,@dat2$        (51B5)
2600 IF UPPER$(MID$(dat1$,mi,feldl
(z3)))<UPPER$(MID$(dat2$,mi,feldl(
z3)))THEN 2630                         (5364)
2610 /RECWRITE,@nu2,@dat1$             (2085)
2620 dat1$=dat2$                       (1446)
2630 NEXT b                            (0AEB)
2640 /RECWRITE,@nu1,@dat1$             (20A9)
2650 NEXT a                            (0A0E)
2660 CLS#5                             (08E0)
2670 CALL &A693,4,&957F                (0F6E)
2680 CALL &A693,3,&8F1F                (0F29)
2690 IF nummer>bis THEN 2710           (1ABA)
2700 GOSUB 3190                        (09D2)
2710 RETURN                            (065C)
2720 WINDOW#3,22,66,14,15+felda:CA
LL &A690,3,&8F1F:CLS#3                 (2C00)
2730 MOVE 174,184:DRAW 524,184:DRA
W 524,168-(felda*16):DRAW 174,168-
(felda*16):DRAW 174,184                (45B4)
2740 FOR a=0 TO felda-1:LOCATE 24,
15+a:PRINT k$(a)": "s$(a):NEXT         (3F6D)
2750 IF z14>felda THEN z14=felda:y
14=14+felda                            (3812)
2760 LOCATE 24,y14:PRINT in$k$(z14
)":"in$                                (2A3D)
2770 FOR w=1 TO 10:NEXT                (113C)
2780 IF INKEY(2)=0 AND y14<14+feld
a THEN y14=y14+1:z14=z14+1:LOCATE
24,y14-1:PRINT k$(z14-1)":":GOTO 2
760                                    (65DE)
2790 IF INKEY(0)=0 AND y14>15 THEN
 y14=y14-1:z14=z14-1:LOCATE 24,y14
+1:PRINT k$(z14+1)":":GOTO 2760        (5C5C)
2800 IF INKEY(9)=0 THEN GOSUB 2840
                                       (122E)
2810 IF INKEY(66)=0 THEN CALL &A69
3,3,&8F1F:i=66:GOSUB 7330:RETURN       (280A)
2820 IF INKEY(60)=0 THEN 2870          (125B)
2830 GOTO 2770                         (09D3)
2840 LOCATE 40,y14:laenge=feldl(z1
4):mini=32:maxi=126:GOSUB 6540         (453A)
2850 LOCATE 40,y14:s$(z14)=e$:PRIN
T s$(z14)+SPACE$(feldl(z14)-LEN(s$
(z14)))                                (5518)
2860 RETURN                            (068A)
2870 CALL &A693,3,&8F1F                (0FA5)
2880 GOSUB 6980                        (09EB)
2890 FOR b=1 TO nummermax              (19E6)
2900 nu=(b-1)*(feldl+2)                (20BF)
2910 /RECREAD,@nu,@recr$:such=1:GO
SUB 6760                               (2D57)
2920 ok=0                              (0C33)
2930 FOR c=0 TO felda-1                (17DF)
2940 IF INKEY(66)=0 THEN GOSUB 702
0:IF jn=0 THEN 3000 ELSE GOSUB 698
0                                      (263F)
2950 IF s$(c)=""THEN ok=ok+1:GOTO
2970                                   (26DC)
2960 IF INSTR(UPPER$(d$(c)),UPPER$
(s$(c)))<>0 THEN ok=ok+1               (378E)
2970 NEXT                              (0602)
2980 IF ok=felda THEN 3010             (18EB)
2990 NEXT                              (0629)
3000 CALL &A693,5,&8F1F:such=0:GOS
UB 3190:RETURN                         (2059)
3010 WINDOW#4,34,60,6,10:CALL &A69
```

```
3,5,&8F1F                                   (1E16)
3020 MOVE 270,312:DRAW 474,312:DRA
W 474,248:DRAW 270,248:DRAW 270,31
2                                           (2F88)
3030 LOCATE 38,7:PRINT"DATENSATZ G
EFUNDEN!"                                   (2189)
3040 LOCATE 36,9:PRINT"WEITER /MEN
]   /DRUCKEN"                               (25A3)
3050 LOCATE 22,11:PRINT USING"####
#";b                                        (1AAB)
3060 GOSUB 6790                             (095D)
3070 LOCATE x15,9:PRINT in$sum$(z1
5)in$                                       (28E6)
3080 FOR w=1 TO 10:NEXT w                   (1582)
3090 IF INKEY(1)=0 AND z15<2 THEN
x15=x15+8:z15=z15+1:LOCATE x15-8,9
:PRINT sum$(z15-1):GOTO 3070                (592A)
3100 IF INKEY(8)=0 AND z15>0 THEN
x15=x15-8:z15=z15-1:LOCATE x15+8,9
:PRINT sum$(z15+1):GOTO 3070                (593C)
3110 IF INKEY(9)=0 THEN ON z15+1 G
OTO 3130,3140,3150                          (23D3)
3120 GOTO 3080                              (09D2)
3130 CLS#4:GOSUB 6980:GOTO 2990             (12E3)
3140 CLS#4:such=0:nummer=b:i=9:GOS
UB 7330:RETURN                              (2F35)
3150 IF druckon=0 THEN GOSUB 7340:
GOTO 3090                                   (1CB8)
3160 bi=lo-2                                (1205)
3170 von=b:bis=von:e=2:GOSUB 1960:
GOTO 3080                                   (2F34)
3180 IF nummer<nummermax THEN numm
er=nummer+1 ELSE RETURN                     (35D9)
3190 LOCATE 22,11:PRINT USING"####
#";nummer                                   (1F7D)
3200 nu=(nummer-1)*(feldl+2)                (2562)
3210 /RECREAD,@nu,@recr$                    (1E92)
3220 GOSUB 6760                             (09AC)
3230 RETURN                                 (066E)
3240 IF nummer>1 THEN nummer=numme
r-1 ELSE RETURN                             (2AC2)
3250 LOCATE 22,11:PRINT USING"####
#";nummer                                   (1FF5)
3260 nu=(nummer-1)*(feldl+2)                (25DA)
3270 /RECREAD,@nu,@recr$                    (1E0A)
3280 GOSUB 6760                             (0925)
3290 RETURN                                 (06E6)
3300 LOCATE 14,11:PRINT in$"NUMMER
:"in$"    "                                 (263F)
3310 LOCATE 22,11:laenge=5:mini=48
:maxi=57:GOSUB 6540                         (3223)
3320 nummer=VAL(e$)                         (176B)
3330 IF nummer>nummermax OR nummer
=0 THEN 3300                                (2C71)
3340 LOCATE 14,11:PRINT"NUMMER: "U
SING"#####";nummer                          (293E)
3350 nu=(nummer-1)*(feldl+2)                (2590)
3360 /RECREAD,@nu,@recr$                    (1EC0)
3370 GOSUB 6760                             (09DA)
3380 RETURN                                 (069C)
3390 WINDOW#1,3,33,6,24:CALL &A690
,1,&8D87:CLS#1                              (214C)
3400 WINDOW#3,58,73,8,21                    (1319)
3410 MOVE 20,296:DRAW 260,296:DRAW
 260,24:DRAW 20,24:DRAW 20,296              (2C5C)
3420 MOVE 20,264:DRAW 260,264               (156D)
3430 LOCATE 5,8:PRINT"USER: "USING
"##";user;:PRINT"        LAUFWERK:
 "CHR$(65+drive)                            (46DE)
3440 WINDOW 5,32,10,23:CLS:/DIR:WI
NDOW 1,80,1,25                              (23B0)
3450 MOVE 462,280:DRAW 580,280:DRA
W 580,72:DRAW 462,72:DRAW 462,280           (2FD6)
3460 MOVE 462,152:DRAW 580,152              (1410)
3470 FOR a=0 TO 6:LOCATE 60,9+a:PR
INT disc$(a):NEXT                           (2AB2)
3480 LOCATE 60,y5:PRINT in$disc$(z
5)in$                                       (281A)
3490 FOR w=1 TO 10:NEXT                     (11E0)
3500 IF INKEY(2)=0 AND y5<15 THEN
y5=y5+1:z5=z5+1:LOCATE 60,y5-1:PRI
NT disc$(z5-1):GOTO 3480                    (5539)
3510 IF INKEY(0)=0 AND y5>9 THEN y
5=y5-1:z5=z5-1:LOCATE 60,y5+1:PRIN
T disc$(z5+1):GOTO 3480                     (541A)
3520 IF INKEY(9)=0 THEN 3550                (1156)
3530 IF INKEY(66)=0 THEN CLS#3:CAL
L &A693,1,&8D87:GOTO 1390                   (222E)
3540 GOTO 3490                              (09EB)
3550 ON z5+1 GOTO 3560,3630,3730,3
780,3830,3850,3890                          (29E8)
3560 WINDOW 60,72,18,19                     (110A)
3570 LOCATE 1,1:PRINT"NAME:"                (12DB)
3580 LOCATE 1,2:mini=32:maxi=122:l
aenge=12:GOSUB 6540                         (319F)
3590 IF e$=""OR INSTR(e$,".")>9 TH
EN 3610                                     (20E9)
3600 /ERA,@e$                               (1086)
3610 CLS:WINDOW 1,80,1,25                   (11D4)
3620 GOTO 3830                              (0932)
3630 WINDOW 60,72,17,20                     (117A)
3640 LOCATE 1,1:PRINT"ALTER NAME:"
                                            (186E)
3650 LOCATE 1,2:mini=32:maxi=122:l
aenge=12:GOSUB 6540:file$=e$                (3E7B)
3660 IF e$=""OR INSTR(e$,".")>9 TH
EN 3710                                     (2091)
3670 LOCATE 1,3:PRINT"NEUER NAME:"
                                            (1867)
3680 LOCATE 1,4:GOSUB 6540                  (0E7F)
3690 IF e$=""OR INSTR(e$,".")>9 TH
EN 3710                                     (20CD)
3700 /REN,@e$,@file$                        (19EB)
3710 CLS:WINDOW 1,80,1,25                   (119D)
3720 GOTO 3830                              (09FA)
3730 WINDOW 60,72,18,19                     (115F)
```

```
3740 LOCATE 1,1:PRINT"NAME:"          <1230>
3750 LOCATE 1,2:mini=32:maxi=122:l
aenge=12:GOSUB 6540                   <31F5>
3760 IF e$=""OR INSTR(e$,".")>9 TH
EN CLS:WINDOW 1,80,1,25:GOTO 3830     <2E86>
3770 RUN e$                           <0A28>
3780 WINDOW 60,72,18,19               <11C3>
3790 LOCATE 1,1:PRINT"NAME:"          <1294>
3800 LOCATE 1,2:mini=32:maxi=122:l
aenge=12:GOSUB 6540                   <3158>
3810 IF e$=""OR INSTR(e$,".")>9 TH
EN CLS:WINDOW 1,80,1,25:GOTO 3830     <2EEA>
3820 LOAD e$                          <0A05>
3830 WINDOW 5,32,10,23:CLS:/DIR:WI
NDOW 1,80,1,25                        <23BC>
3840 GOTO 3490                        <0946>
3850 user=user+1:IF user>15 THEN u
ser=0                                 <2C6A>
3860 LOCATE 11,8:PRINT USING"##";u
ser                                   <1927>
3870 /USER,user                       <13DC>
3880 GOTO 3490                        <0996>
3890 drive=drive+1:IF drive>1 THEN
 drive=0                              <2FD1>
3900 LOCATE 31,8:PRINT CHR$(65+dri
ve)                                   <1B6F>
3910 drive$=CHR$(65+drive)            <1D68>
3920 WINDOW 60,72,18,19               <11DB>
3930 /DRIVE,@drive$                   <16D8>
3940 CLS:WINDOW 1,80,1,25             <116A>
3950 GOTO 3830                        <09C7>
3960 WINDOW#1,48,66,8,20:CALL &A69
0,1,&8D87:CLS#1                       <2284>
3970 MOVE 382,278:DRAW 520,278:DRA
W 520,88:DRAW 382,88:DRAW 382,278     <2F04>
3980 FOR a=0 TO 10:LOCATE 50,9+a:P
RINT so$(a):NEXT                      <294D>
3990 LOCATE 50,y6:PRINT in$so$(z6)
in$                                   <265E>
4000 FOR w=1 TO 10:NEXT               <11DE>
4010 IF INKEY(2)=0 AND y6<19 THEN
y6=y6+1:z6=z6+1:LOCATE 50,y6-1:PRI
NT so$(z6-1):GOTO 3990                <53E3>
4020 IF INKEY(0)=0 AND y6>9 THEN y
6=y6-1:z6=z6-1:LOCATE 50,y6+1:PRIN
T so$(z6+1):GOTO 3990                 <52D9>
4030 IF INKEY(9)=0 THEN i=9:GOSUB
7330:ON z6+1 GOTO 4060,4530,5030,5
180,5470,5680,5840,5990,6150,6330,
6450                                  <4E66>
4040 IF INKEY(66)=0 THEN CALL &A69
3,1,&8D87:GOTO 1390                   <1E3E>
4050 GOTO 4000                        <09E1>
4060 WINDOW#2,14,42,17,20:CALL &A6
90,2,&95D7:CLS#2                      <2318>
4070 MOVE 108,134:DRAW 328,134:DRA
W 328,88:DRAW 108,88:DRAW 108,134     <29AF>
4080 LOCATE 16,18:PRINT dl$(0)" "d
atei$".DAT"                           <2678>
4090 LOCATE 16,19:PRINT dl$(1)        <15B5>
4100 LOCATE 16,y12:PRINT in$dl$(z1
2)in$                                 <28F0>
4110 FOR w=1 TO 10:NEXT               <11BA>
4120 IF INKEY(2)=0 AND y12<19 THEN
 y12=y12+1:z12=z12+1:LOCATE 16,y12
-1:PRINT dl$(z12-1):GOTO 4100         <5AB7>
4130 IF INKEY(0)=0 AND y12>18 THEN
 y12=y12-1:z12=z12-1:LOCATE 16,y12
+1:PRINT dl$(z12+1):GOTO 4100         <5A17>
4140 IF INKEY(9)=0 THEN ON z12+1 G
OTO 4170,4220                         <1F40>
4150 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                     <2BA0>
4160 GOTO 4110                        <0937>
4170 LOCATE 29,y12:laenge=8:mini=4
8:maxi=122:GOSUB 6540                 <369D>
4180 IF e$=""THEN 4200                <1128>
4190 datei$=UPPER$(e$)+SPACE$(8-LE
N(e$))                                <253C>
4200 LOCATE 29,y12:PRINT datei$       <19FA>
4210 GOTO 4110                        <099B>
4220 IF datei$=SPACE$(8)THEN 4600     <1896>
4230 GOSUB 7210                       <09B2>
4240 LOCATE 29,y12:PRINT"OK."         <1659>
4250 CLOSEIN                          <0667>
4260 OPENIN datei$+".PRG"             <1544>
4270 FOR a=0 TO 9:INPUT#9,fu$(a):N
EXT                                   <2036>
4280 FOR a=0 TO 9:INPUT#9,k$(a):NE
XT                                    <1F71>
4290 FOR a=0 TO 4:INPUT#9,druck(a)
:NEXT                                 <23B4>
4300 FOR a=0 TO 9:INPUT#9,feldl(a)
:feldlsi(a)=feldl(a):NEXT             <434C>
4310 INPUT#9,papier                   <127F>
4320 INPUT#9,rand                     <1022>
4330 INPUT#9,stift                    <11C6>
4340 INPUT#9,datum$                   <11ED>
4350 INPUT#9,ta                       <0EF3>
4360 INPUT#9,mo                       <0E08>
4370 INPUT#9,jahr                     <108E>
4380 INPUT#9,nummermax                <1527>
4390 INPUT#9,feldl:fl=feldl           <2047>
4400 INPUT#9,felda:fa=felda           <20B4>
4410 CLOSEIN                          <06A7>
4420 OPENIN datei$+".DAT"             <1554>
4430 CALL &A693,2,&95D7               <0F73>
4440 open=1                           <0EC8>
4450 IF felda<5 THEN lo=felda         <1F3F>
4460 BORDER rand:INK 0,papier:INK
1,stift                               <269E>
4470 LOCATE 62,2:PRINT datum$         <14EA>
4480 FOR a=0 TO 9:KEY a,fu$(a):NEX
T                                     <22E4>
4490 FOR a=felda TO 9:k$(a)="":NEX
```

```
T                                                    <2598>
4500 nummer=1                                        <10DB>
4510 GOSUB 6700                                      <09EB>
4520 GOTO 3960                                       <094E>
4530 WINDOW#2,11,39,9,13:CALL &A69
0,2,&95D7:CLS#2                                      <2220>
4540 nummermax=INT(178000/(feldl+2
))                                                   <2920>
4550 MOVE 84,262:DRAW 304,262:DRAW
 304,200:DRAW 84,200:DRAW 84,262                     <2CF8>
4560 LOCATE 13,10:PRINT da$(0)" "d
atei$".DAT"                                          <26B8>
4570 LOCATE 13,11:PRINT da$(1)" "U
SING"#####";nummermax                                <2D0E>
4580 LOCATE 13,12:PRINT da$(2)                       <1580>
4590 LOCATE 13,y11:PRINT in$da$(z1
1)in$                                                <2808>
4600 FOR w=1 TO 10:NEXT                              <118F>
4610 IF INKEY(2)=0 AND y11<12 THEN
 y11=y11+1:z11=z11+1:LOCATE 13,y11
-1:PRINT da$(z11-1):GOTO 4590                        <5A77>
4620 IF INKEY(0)=0 AND y11>10 THEN
 y11=y11-1:z11=z11-1:LOCATE 13,y11
+1:PRINT da$(z11+1):GOTO 4590                        <5A89>
4630 IF INKEY(9)=0 THEN ON z11+1 G
OTO 4660,4640,4710                                   <23D5>
4640 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 401
0                                                    <2BCA>
4650 GOTO 4600                                       <095E>
4660 LOCATE 26,y11:laenge=8:mini=4
8:maxi=122:GOSUB 6540                                <3678>
4670 IF e$=""THEN 4690                               <1157>
4680 datei$=UPPER$(e$)+SPACE$(8-LE
N(e$))                                               <2512>
4690 LOCATE 26,y11:PRINT datei$                      <19C8>
4700 GOTO 4600                                       <09C2>
4710 IF datei$=""THEN 4600                           <159E>
4720 GOSUB 7270                                      <0962>
4730 LOCATE 26,y11:PRINT"OK."                        <1632>
4740 CLOSEOUT                                        <0641>
4750 OPENOUT datei$+".PRG"                           <1518>
4760 FOR a=0 TO 9:PRINT#9,fu$(a):N
EXT                                                  <20EE>
4770 FOR a=0 TO felda-1:PRINT#9,k$
(a):NEXT:FOR a=felda TO 9:k$(a)=ST
RING$(14,32):PRINT#9,k$(a):NEXT                      <5F8C>
4780 FOR a=0 TO 4:PRINT#9,druck(a)
:NEXT                                                <23D7>
4790 FOR a=0 TO 9:PRINT#9,feldl(a)
:feldlsi(a)=feldl(a):NEXT                            <43AB>
4800 PRINT#9,papier                                  <1284>
4810 PRINT#9,rand                                    <10C5>
4820 PRINT#9,stift                                   <1124>
4830 PRINT#9,datum$                                  <113B>
4840 PRINT#9,ta                                      <0EB6>
4850 PRINT#9,mo                                      <0ECA>
4860 PRINT#9,jahr                                    <102F>
4870 PRINT#9,nummermax                               <157F>
4880 PRINT#9,feldl:fl=feldl                          <2074>
4890 PRINT#9,felda:fa=felda                          <2036>
4900 CLOSEOUT                                        <0681>
4910 OPENOUT datei$+".DAT"                           <1528>
4920 FOR a=1 TO nummermax                            <1904>
4930 LOCATE 33,11:PRINT USING"####
#";a                                                 <1A2F>
4940 PRINT#9,SPACE$(fl)                              <12FF>
4950 NEXT                                            <0682>
4960 PRINT#9,SPACE$(128)SPACE$(128
)                                                    <155E>
4970 CLOSEOUT                                        <060D>
4980 open=1                                          <0E02>
4990 CLOSEIN                                         <0632>
5000 OPENIN datei$+".DAT"                            <15DE>
5010 CALL &A693,2,&95D7                              <0FFD>
5020 GOTO 4010                                       <09C9>
5030 WINDOW#2,2,18,13,24:CALL &A69
0,2,&95D7:CLS#2                                      <2273>
5040 MOVE 8,198:DRAW 138,198:DRAW
138,26:DRAW 8,26:DRAW 8,198                          <2454>
5050 FOR a=0 TO 9:LOCATE 3,14+a:PR
INT USING"F#: ";a;:PRINT fu$(a):NE
XT                                                   <375B>
5060 LOCATE 3,y7:PRINT in$;:PRINT
USING"F#:";z7;:PRINT in$                             <2BCC>
5070 FOR w=1 TO 10:NEXT                              <113D>
5080 IF INKEY(2)=0 AND y7<23 THEN
y7=y7+1:z7=z7+1:LOCATE 3,y7-1:PRIN
T USING"F#:";z7-1:GOTO 5060                          <5219>
5090 IF INKEY(0)=0 AND y7>14 THEN
y7=y7-1:z7=z7-1:LOCATE 3,y7+1:PRIN
T USING"F#:";z7+1:GOTO 5060                          <5205>
5100 IF INKEY(9)=0 THEN GOSUB 5130
                                                     <12EA>
5110 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                                    <2B23>
5120 GOTO 5070                                       <09C4>
5130 LOCATE 7,y7:laenge=10:mini=32
:maxi=126:GOSUB 6540                                 <3589>
5140 IF e$=""THEN e$=fu$(z7)                         <1F3A>
5150 fu$(z7)=e$:LOCATE 7,y7:PRINT
fu$(z7)+SPACE$(10-LEN(e$))                           <3DF6>
5160 KEY z7,fu$(z7)                                  <188F>
5170 RETURN                                          <069F>
5180 WINDOW#2,18,52,21,24:CALL &A6
90,2,&95D7:CLS#2                                     <23E6>
5190 MOVE 142,72:DRAW 408,72:DRAW
408,24:DRAW 142,24:DRAW 142,72                       <29BB>
5200 LOCATE 20,22:PRINT"ALTES DATU
M: "datum$                                           <2424>
5210 LOCATE 20,23:PRINT"NEUES DATU
M:"                                                  <1B04>
5220 LOCATE 33,23:PRINT USING"##";
ta                                                   <1852>
5230 FOR w=1 TO 30:NEXT                              <11C2>
```

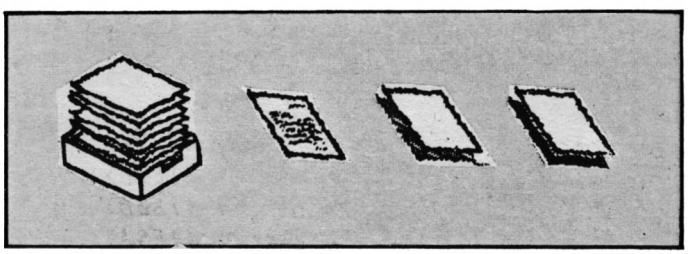
```
5240 IF INKEY(1)=0 AND ta<31 THEN
ta=ta+1:GOTO 5220                    <29BF>
5250 IF INKEY(8)=0 AND ta>1 THEN t
a=ta-1:GOTO 5220                     <28A6>
5260 IF INKEY(9)=0 THEN 5280         <111F>
5270 GOTO 5230                       <09F0>
5280 LOCATE 33,23:PRINT STRING$(3-
LEN(STR$(ta)),"0");MID$(STR$(ta),2
,2);"."                              <39DD>
5290 i=9:GOSUB 7330                  <10F9>
5300 LOCATE 37,23:PRINT mo$(mo)      <1923>
5310 FOR w=1 TO 30:NEXT              <1163>
5320 IF INKEY(1)=0 AND mo<11 THEN
mo=mo+1:GOTO 5300                    <295F>
5330 IF INKEY(8)=0 AND mo>0 THEN m
o=mo-1:GOTO 5300                     <2820>
5340 IF INKEY(9)=0 THEN i=9:GOSUB
7330:GOTO 5360                       <1E60>
5350 GOTO 5310                       <0911>
5360 LOCATE 47,23:PRINT USING"####
";jahr                               <1CEB>
5370 FOR w=1 TO 30:NEXT              <11DB>
5380 IF INKEY(1)=0 AND jahr<9999 T
HEN jahr=jahr+1:GOTO 5360            <30EB>
5390 IF INKEY(8)=0 AND jahr>1986 T
HEN jahr=jahr-1:GOTO 5360            <30C6>
5400 IF INKEY(9)=0 THEN 5420         <1199>
5410 GOTO 5370                       <096F>
5420 datum$=STRING$(3-LEN(STR$(ta)
),"0")+MID$(STR$(ta),2,2)+". "+mo$
(mo)+" "+MID$(STR$(jahr),2,4)        <5FA4>
5430 LOCATE 62,2:PRINT datum$        <146E>
5440 CALL &A693,2,&95D7              <0F5B>
5450 i=9:GOSUB 7330                  <103A>
5460 GOTO 4000                       <09EC>
5470 WINDOW#2,6,20,7,11:CALL &A690
,2,&95D7:CLS#2                       <21F7>
5480 MOVE 48,296:DRAW 146,296:DRAW
 146,230:DRAW 48,230:DRAW 48,296     <2A7C>
5490 LOCATE 8,8:PRINT fa$(0)USING"
 ##";papier                          <23AA>
5500 LOCATE 8,9:PRINT fa$(1)USING"
 ##";rand                            <21AD>
5510 LOCATE 8,10:PRINT fa$(2)USING
" ##";stift                          <2350>
"                                    <21EF>
5530 FOR w=1 TO 10:NEXT              <11D8>
5540 IF INKEY(2)=0 AND y8<10 THEN
y8=y8+1:z8=z8+1:LOCATE 8,y8-1:PRIN
T fa$(z8-1):GOTO 5520                <5201>
5550 IF INKEY(0)=0 AND y8>8 THEN y
8=y8-1:z8=z8-1:LOCATE 8,y8+1:PRINT
 fa$(z8+1):GOTO 5520                 <51F1>
5560 IF INKEY(1)=0 THEN GOSUB 5600
                                     <1217>
5570 IF INKEY(8)=0 THEN GOSUB 5640
                                     <128C>
5580 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                    <2BD2>
5590 GOTO 5530                       <09D7>
5600 IF z8=0 AND papier<26 THEN pa
pier=papier+1:INK 0,papier:LOCATE
15,8:PRINT USING" ##";papier         <55C3>
5610 IF z8=1 AND rand<26 THEN rand
=rand+1:BORDER rand:LOCATE 15,9:PR
INT USING" ##";rand                  <493B>
5620 IF z8=2 AND stift<26 THEN sti
ft=stift+1:INK 1,stift:LOCATE 15,1
0:PRINT USING" ##";stift             <51F8>
5630 RETURN                          <0638>
5640 IF z8=0 AND papier>0 THEN pap
ier=papier-1:INK 0,papier:LOCATE 1
5,8:PRINT USING" ##";papier          <54CD>
5650 IF z8=1 AND rand>0 THEN rand=
rand-1:BORDER rand:LOCATE 15,9:PRI
NT USING" ##";rand                   <4861>
5660 IF z8=2 AND stift>0 THEN stif
t=stift-1:INK 1,stift:LOCATE 15,10
:PRINT USING" ##";stift              <506F>
5670 RETURN                          <0689>
5680 WINDOW#2,22,42,14,14+felda+1:
CALL &A690,2,&95D7:CLS#2             <2E40>
5690 MOVE 176,184:DRAW 328,184:DRA
W 328,168-(felda*16):DRAW 176,168-
(felda*16):DRAW 176,184              <450D>
5700 FOR a=0 TO felda-1:LOCATE 24,
15+a:PRINT USING"##";a+1;:PRINT".
"k$(a):NEXT                          <4418>
5710 IF z9>felda-1 THEN z9=felda:y
9=14+felda                           <37A5>
5720 LOCATE 24,y9:PRINT in$USING"#
#";z9;:PRINT"."in$                   <2B5C>
5730 FOR w=1 TO 10:NEXT              <1169>
5740 IF INKEY(2)=0 AND y9<14+felda
 THEN y9=y9+1:z9=z9+1:LOCATE 24,y9
-1:PRINT USING"##";z9-1;:PRINT".":
GOTO 5720                            <613C>
5750 IF INKEY(0)=0 AND y9>15 THEN
y9=y9-1:z9=z9-1:LOCATE 24,y9+1:PRI
NT USING"##";z9+1;:PRINT".":GOTO 5
720                                  <58EB>
5760 IF INKEY(9)=0 THEN GOSUB 5790
                                     <12BB>
5770 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:IF ae=1
THEN FOR a=0 TO felda-1:LOCATE 14,
13+a:PRINT k$(a)": ":NEXT:GOTO 400
0 ELSE 4000                          <6A98>
5780 GOTO 5730                       <0996>
5790 LOCATE 28,y9:laenge=13:mini=3
2:maxi=126:GOSUB 6540                <3647>
5800 IF e$=""THEN LOCATE 28,y9:PRI
NT k$(z9-1):RETURN                   <28E9>
5810 ae=1                            <0CB4>
5820 k$(z9-1)=e$+SPACE$(14-LEN(e$)
```

```
):LOCATE 28,y9:PRINT k$(z9-1)      <40C6>
5830 RETURN                        <06C8>
5840 IF felda<5 THEN lo=felda ELSE
 lo=5                              <28EA>
5850 IF z10>felda THEN z10=felda:y
10=7+felda                         <37E9>
5860 WINDOW#2,10,31,7,8+lo:CALL &A
690,2,&95D7:CLS#2                  <2787>
5870 MOVE 78,294:DRAW 236,294:DRAW
 236,282-(lo*16):DRAW 78,282-(lo*1
6):DRAW 78,294                     <4274>
5880 FOR a=0 TO lo-1:LOCATE 12,8+a
:PRINT USING"##";a+1;:PRINT". "k$(
druck(a)):NEXT                     <4ABD>
5890 LOCATE 12,y10:PRINT in$USING"
##";z10;:PRINT"."in$               <2D84>
5900 FOR w=1 TO 10:NEXT            <11BD>
5910 IF INKEY(2)=0 AND y10<7+lo TH
EN y10=y10+1:z10=z10+1:LOCATE 12,y
10-1:PRINT USING"##";z10-1;:PRINT"
.":GOTO 5890                       <6448>
5920 IF INKEY(0)=0 AND y10>8 THEN
y10=y10-1:z10=z10-1:LOCATE 12,y10+
1:PRINT USING"##";z10+1;:PRINT".":
GOTO 5890                          <5EA3>
5930 IF INKEY(1)=0 AND druck(y10-8
)<felda-1 THEN druck(y10-8)=druck(
y10-8)+1:GOTO 5970                 <5854>
5940 IF INKEY(8)=0 AND druck(y10-8
)>0 THEN druck(y10-8)=druck(y10-8)
-1:GOTO 5970                       <4F6B>
5950 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                  <2BB7>
5960 GOTO 5900                     <095A>
5970 LOCATE 16,y10:PRINT k$(druck(
y10-8))                            <2930>
5980 GOTO 5900                     <0982>
5990 WINDOW#2,30,42,11,14:CALL &A6
90,2,&95D7:CLS#2                   <23E2>
6000 MOVE 238,230:DRAW 320,230:DRA
W 320,184:DRAW 238,184:DRAW 238,23
0                                  <2971>
6010 LOCATE 32,12:PRINT drucker$(0
):LOCATE 32,13:PRINT drucker$(1)   <300B>
6020 LOCATE 32,y13:PRINT in$drucke
r$(z13)in$                         <2D0A>
6030 FOR w=1 TO 10:NEXT            <11C2>
6040 IF INKEY(2)=0 AND y13<13 THEN
 y13=y13+1:z13=z13+1:LOCATE 32,y13
-1:PRINT drucker$(z13-1):GOTO 6020
                                   <5FDF>
6050 IF INKEY(0)=0 AND y13>12 THEN
 y13=y13-1:z13=z13-1:LOCATE 32,y13
+1:PRINT drucker$(z13+1):GOTO 6020
                                   <5F74>
6060 IF INKEY(9)=0 THEN ON z13+1 G
OTO 6090,6090                      <1F68>
6070 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                  <2BA8>
6080 GOTO 6030                     <0955>
6090 WIDTH 142                     <0827>
6100 IF INP(&F500)=90 THEN GOSUB 6
840:IF dr=1 THEN GOTO 6030         <236D>
6110 PRINT#8,CHR$(15)              <0FA9>
6120 PRINT#8,CHR$(27)CHR$(120)CHR$
(0)CHR$(27)CHR$(83)CHR$(0)         <2B2A>
6130 PRINT#8,CHR$(27)CHR$(72)CHR$(
27)CHR$(51)CHR$(16)                <27A7>
6140 CALL &A693,2,&95D7::druckon=1
:GOTO 4000                         <22BF>
6150 WINDOW#2,8,29,7,8+felda:CALL
&A690,2,&95D7:CLS#2                <29EA>
6160 MOVE 60,296:DRAW 228,296:DRAW
 228,280-felda*16:DRAW 60,280-feld
a*16:DRAW 60,296                   <4436>
6170 FOR a=0 TO felda-1:LOCATE 10,
8+a:PRINT k$(a)": "USING"##";feldl
(a):NEXT                           <489C>
6180 WINDOW#3,12,25,22,23:CALL &A6
90,3,&9E17                         <1FDF>
6190 WINDOW#4,1,25,22,25:CLS#4     <1769>
6200 MOVE 8,56:DRAW 192,56:DRAW 19
2,8:DRAW 8,8:DRAW 8,56             <22C3>
6210 LOCATE 3,23:PRINT"MAXIMUM  :
25 ZEICHEN"                        <237A>
6220 LOCATE 3,24:PRINT"MINIMUM  :
 1 ZEICHEN"                        <23D2>
6230 IF z16>felda-1 THEN z16=felda
-1:y16=7+felda                     <3BB9>
6240 LOCATE 26,y16:PRINT USING"##"
;feldl(z16)                        <2752>
6250 LOCATE 10,y16:PRINT in$k$(z16
)":"in$                            <2A9B>
6260 FOR w=1 TO 10:NEXT            <118F>
6270 IF INKEY(2)=0 AND y16<7+felda
 THEN y16=y16+1:z16=z16+1:LOCATE 1
0,y16-1:PRINT k$(z16-1)":":GOTO 62
50                                 <64EC>
6280 IF INKEY(0)=0 AND y16>8 THEN
y16=y16-1:z16=z16-1:LOCATE 10,y16+
1:PRINT k$(z16+1)":":GOTO 6250     <5B5F>
6290 IF INKEY(1)=0 AND feldl(z16)<
25 THEN feldl(z16)=feldl(z16)+1:GO
TO 6240                            <4AC1>
6300 IF INKEY(8)=0 AND feldl(z16)>
1 THEN feldl(z16)=feldl(z16)-1:GOT
O 6240                             <495E>
6310 IF INKEY(66)=0 THEN feldl=0:F
OR a=0 TO felda-1:feldl=feldl+feld
l(a):NEXT:CALL &A693,2,&95D7:CLS#4
:CALL &A693,3,&9E17:i=66:GOSUB 733
0:GOTO 4010                        <7B69>
6320 GOTO 6260                     <0969>
6330 WINDOW#2,9,34,9,13:CALL &A690
```

```
,2,&95D7:CLS#2                                  <21AA>
6340 MOVE 70,264:DRAW 258,264:DRAW
 258,198:DRAW 70,198:DRAW 70,264                <2C5D>
6350 fda=felda                                  <140D>
6360 LOCATE 11,10:PRINT"MAXIMUM
: 10 FELDER"                                    <24C7>
6370 LOCATE 11,11:PRINT"MINIMUM
:  1 FELD"                                      <220B>
6380 LOCATE 11,12:PRINT in$"FELDAN
ZAHL:"in$"  FELDER"                             <3004>
6390 LOCATE 23,12:PRINT USING"##";
felda                                           <1BC5>
6400 FOR w=1 TO 30:NEXT                         <11EC>
6410 IF INKEY(1)=0 AND felda<10 TH
EN felda=felda+1:GOTO 6390                      <320A>
6420 IF INKEY(8)=0 AND felda>1 THE
N felda=felda-1:GOTO 6390                       <3136>
6430 IF INKEY(66)=0 THEN feldl=0:F
OR a=0 TO felda-1:feldl=feldl+feld
l(a):NEXT:CALL &A693,2,&95D7:i=66:
GOSUB 7330:IF felda<>fda THEN GOSU
B 6700:GOTO 3960 ELSE 4010                      <870C>
6440 GOTO 6400                                  <09C3>
6450 WINDOW#2,8,32,14,16:CALL &A69
0,2,&95D7:CLS#2                                 <2297>
6460 MOVE 62,182:DRAW 250,182:DRAW
 250,152:DRAW 62,152:DRAW 62,182                <274F>
6470 LOCATE 10,15:PRINT in$"WARMST
ART"in$" / KALTSTART":jn=0                      <388E>
6480 FOR w=1 TO 10:NEXT w                       <1520>
6490 IF INKEY(8)=0 THEN LOCATE 10,
15:PRINT in$"WARMSTART"in$" / KALT
START":jn=0                                     <41AF>
6500 IF INKEY(1)=0 THEN LOCATE 10,
15:PRINT"WARMSTART / "in$"KALTSTAR
T"in$:jn=1                                      <4174>
6510 IF INKEY(66)=0 THEN CALL &A69
3,2,&95D7:i=66:GOSUB 7330:GOTO 400
0                                               <2B1A>
6520 IF INKEY(9)=0 THEN IF jn=1 TH
EN CALL 0 ELSE i=9:GOSUB 7330:GOTO
 1310                                           <2BDF>
6530 GOTO 6480                                  <09F2>
6540 e$="":l=0:CALL &BB81                       <1816>
6550 WHILE INKEY$<>"":WEND                      <0DC2>
6560 IF PEEK(6)=&80 THEN POKE &B4E
8,0 ELSE POKE &B632,0                           <1EDB>
6570 a$=INKEY$:IF a$=""THEN 6570                <19B4>
6580 IF a$=CHR$(13)THEN CALL &BB84
:RETURN                                         <1887>
6590 IF a$=CHR$(32)THEN 6650                    <1521>
6600 IF a$<>CHR$(127)THEN 6630                  <15F3>
6610 IF LEN(e$)>=1 THEN PRINT CHR$
(8)CHR$(16);                                    <1E9C>
6620 IF LEN(e$)>=1 THEN e$=LEFT$(e
$,LEN(e$)-1):l=l-1                              <3540>
6630 IF a$<CHR$(mini)THEN 6570                  <1AE6>
6640 IF a$>CHR$(maxi)THEN 6570                  <1AB4>
6650 l=l+1:IF l>laenge THEN l=laen
ge:GOTO 6570                                    <3430>
6660 e$=e$+a$                                   <13D8>
6670 PRINT a$;                                  <0BCE>
6680 IF l=laenge THEN 6570                      <186E>
6690 GOTO 6570                                  <0905>
6700 LOCATE 1,7:PRINT CHR$(20)                  <1106>
6710 MOVE 94,216:DRAW 94,248:DRAW
216,248:DRAW 216,216                            <20D9>
6720 MOVE 94,216:DRAW 444,216:DRAW
 444,200-(felda*16):DRAW 94,200-(f
elda*16):DRAW 94,216                            <4504>
6730 LOCATE 14,11:PRINT"NUMMER: "U
SING"#####";nummer                              <29C7>
6740 FOR a=0 TO felda-1:LOCATE 14,
13+a:PRINT k$(a)": ":NEXT                       <3527>
6750 RETURN                                     <06FD>
6760 mitte=0                                    <0F9C>
6770 FOR c=0 TO felda-1:d$(c)=MID$
(recr$,1+mitte,feldl(c)):mitte=mit
te+feldl(c):NEXT                                <6A5D>
6780 IF such=1 THEN RETURN                      <11E7>
6790 WINDOW 30,54,13,12+felda:CLS:
WINDOW 1,80,1,25                                <27B8>
6800 FOR a=0 TO felda-1                         <172A>
6810 LOCATE 30,13+a:PRINT d$(a)                 <1CD9>
6820 NEXT                                       <0624>
6830 RETURN                                     <069C>
6840 WINDOW#3,6,42,22,24:CALL &A69
0,3,&9C87:CLS#3                                 <226B>
6850 MOVE 48,56:DRAW 328,56:DRAW 3
28,24:DRAW 48,24:DRAW 48,56                     <29B9>
6860 LOCATE 8,23:PRINT"DRUCKER IST
 NICHT EMPFANGSBEREIT!"                         <2F8F>
6870 FOR a=1 TO 600:IF INP(&F500)=
26 THEN CALL &A693,3,&9C87:RETURN
ELSE NEXT                                       <2C3F>
6880 CALL &A693,3,&9C87:dr=1:RETUR
N                                               <19F4>
6890 WINDOW#1,2,19,5,7                          <116E>
6900 MOVE 16,328:DRAW 128,328:DRAW
 128,296:DRAW 16,296:DRAW 16,328                <2CC3>
6910 LOCATE 4,6:PRINT"DATEI LADEN!
"                                               <19F2>
6920 FOR w=1 TO 1500:NEXT                       <124C>
6930 CLS#1:RETURN                               <0A62>
6940 WINDOW#5,61,76,22,24                       <14B5>
6950 MOVE 486,54:DRAW 606,54:DRAW
606,24:DRAW 486,24:DRAW 486,54                  <2CCA>
6960 LOCATE 63,23:PRINT"BITTE WART
EN!"                                            <1C9E>
6970 RETURN                                     <06B6>
6980 WINDOW#5,5,21,20,22:CALL &A69
0,5,&8F1F:CLS#5                                 <22BF>
6990 MOVE 38,88:DRAW 162,88:DRAW 1
62,56:DRAW 38,56:DRAW 38,88                     <27FF>
7000 LOCATE 7,21:PRINT"BITTE WARTE
```

```
N!"                                    (1BC1)
7010 RETURN                            (0606)
7020 CALL &A693,5,&8F1F                (0FA6)
7030 WINDOW#6,49,63,19,23:CALL &A6
90,6,&90B7:CLS#6                       (238B)
7040 MOVE 392,104:DRAW 496,104:DRA
W 496,38:DRAW 392,38:DRAW 392,104      (2CCC)
7050 LOCATE 51,20:PRINT"UNTERBRUCH
!"                                     (1A9A)
7060 LOCATE 51,22:PRINT"JA   / "in
$"NEIN"in$:jn=1                        (2E8D)
7070 FOR w=1 TO 10:NEXT                (11E6)
7080 IF INKEY(8)=0 THEN LOCATE 51,
22:PRINT in$"JA  "in$" / NEIN":jn=
0                                      (37CB)
7090 IF INKEY(1)=0 THEN LOCATE 51,
22:PRINT"JA   / "in$"NEIN"in$:jn=1
                                       (3795)
7100 IF INKEY(9)=0 THEN CALL &A693
,6,&90B7:i=9:GOSUB 7330:RETURN         (26CE)
7110 GOTO 7070                         (09F5)
7120 WINDOW#6,31,45,9,13:CLS#5:CAL
L &A690,6,&9867:CLS#6                  (26F2)
7130 MOVE 248,264:DRAW 352,264:DRA
W 352,198:DRAW 248,198:DRAW 248,26
4                                      (2C1F)
7140 LOCATE 33,10:PRINT"UNTERBRUCH
!"                                     (1AB5)
7150 LOCATE 33,12:PRINT"JA   / "in
$"NEIN"in$:jn=1                        (2E16)
7160 FOR w=1 TO 10:NEXT                (1199)
7170 IF INKEY(8)=0 THEN LOCATE 33,
12:PRINT in$"JA  "in$" / NEIN":jn=
0                                      (37B5)
7180 IF INKEY(1)=0 THEN LOCATE 33,
12:PRINT"JA   / "in$"NEIN"in$:jn=1
                                       (3736)
7190 IF INKEY(9)=0 THEN CALL &A693
,6,&9867:i=9:GOSUB 7330:GOSUB 6940
:RETURN                                (2B7F)
7200 GOTO 7160                         (097E)
7210 WINDOW#3,8,42,7,11::CALL &A69
0,3,&9A5F:CLS#3                        (229E)
7220 MOVE 64,296:DRAW 328,296:DRAW
 328,232:DRAW 64,232:DRAW 64,296       (2CE5)
7230 LOCATE 10,8:PRINT"BITTE DATEN
DISKETTE EINLEGEN..."                  (2D18)
7240 LOCATE 12,10:PRINT"UND DIE EN
TER-TASTE DRJCKEN"                     (2AA3)
7250 WHILE INKEY$<>CHR$(13):WEND       (115A)
7260 CALL &A693,3,&9A5F:RETURN         (1171)
7270 WINDOW#3,14,49,20,24:CALL &A6
90,3,&9A5F:CLS#3                       (23DF)
7280 MOVE 110,88:DRAW 386,88:DRAW
386,24:DRAW 110,24:DRAW 110,88         (2943)
7290 LOCATE 16,21:PRINT"BITTE LEER
E DISKETTE EINLEGEN..."                (2FC7)
7300 LOCATE 18,23:PRINT"UND DIE EN
TER-TASTE DRJCKEN"                     (2AE7)
7310 WHILE INKEY$<>CHR$(13):WEND       (11D2)
7320 CALL &A693,3,&9A5F:RETURN         (11E9)
7330 WHILE INKEY(i)<>-1:WEND:RETUR
N                                      (153F)
7340 WINDOW#3,1,26,16,18:CALL &A69
0,3,&8F1F:CLS#3                        (22B3)
7350 MOVE 8,152:DRAW 200,152:DRAW
200,120:DRAW 8,120:DRAW 8,152          (24B8)
7360 LOCATE 3,17:PRINT"DRUCKER INI
TIALISIERT?"                           (24BC)
7370 FOR w=1 TO 1500:NEXT w            (163F)
7380 CALL &A693,3,&8F1F                (0FF3)
7390 RETURN                            (06FE)
100 '*****************************    (2234)
110 '*        MCLOADER 1          *    (222F)
120 '*   ERZEUGT >>MC1.BIN<<      *    (22E4)
130 '*     FUER MINIDATEI         *    (22C4)
140 '*          VON               *    (22A7)
150 '*     MARKUS WILDI           *    (22DA)
160 '*          FUER              *    (223C)
170 '*   SCHNEIDER CPC-WELT       *    (225B)
180 '*    CPC 464/664/6128    JE*       (226A)
190 '*****************************    (22E8)
640 MEMORY &9FFF                       (098A)
650 a=&A000:e=&A180:zb=1000:e=e+1      (2CE0)
660 FOR i=a TO e:IF i=e THEN SAVE"
mc1.bin",b,&A000,&180:END              (39E2)
670 READ d$:POKE i,VAL("&"+d$)         (1D5A)
730 IF i<e THEN NEXT i                 (1526)
1001 DATA 01,0A,A0,21,22,A0,CD,D1      (1E3A)
1002 DATA BC,C9,12,A0,C3,CF,A0,C3      (1ED8)
1003 DATA 0D,A1,52,45,43,52,45,41      (1E90)
1004 DATA C4,52,45,43,57,52,49,54      (1E65)
1005 DATA C5,00,00,00,00,00,00,00      (1EC3)
1006 DATA 00,00,00,00,00,00,00,00      (1E82)
1007 DATA 00,C3,CF,A0,C3,0D,A1,FE      (1E03)
1008 DATA 02,C2,B4,A0,DD,6E,02,DD      (1E56)
1009 DATA 66,03,2B,7E,FE,04,C2,B4      (1E5C)
1010 DATA A0,23,11,26,A0,01,05,00      (1EEE)
1011 DATA ED,B0,DD,6E,00,DD,66,01      (1E6E)
1012 DATA 2B,7E,FE,02,C2,B4,A0,23      (1EEE)
1013 DATA 11,2E,A0,01,03,00,ED,B0      (1EB2)
1014 DATA 3A,2A,A0,FE,81,30,0D,21      (1E1B)
1015 DATA 00,00,22,2B,A0,3E,00,32      (1E54)
1016 DATA 2D,A0,37,C9,FE,98,D2,B4      (1EE1)
1017 DATA A0,21,29,A0,CB,FE,3A,2A      (1E8A)
1018 DATA A0,FE,97,28,0F,21,2A,A0      (1EBB)
1019 DATA 34,2B,CB,3E,2B,CB,1E,2B      (1E62)
1020 DATA CB,1E,18,EA,2A,28,A0,22      (1E90)
1021 DATA 2B,A0,3A,27,A0,CB,3F,32      (1EF0)
1022 DATA 2D,A0,3A,2E,A0,FE,00,CA      (1E68)
1023 DATA B4,A0,37,C9,37,3F,C9,2A      (1EAD)
1024 DATA 2B,A0,23,22,2B,A0,11,29      (1E14)
1025 DATA 00,2A,7D,BE,19,3A,2B,A0      (1E3E)
1026 DATA 77,23,3A,2C,A0,77,C9,CD      (1E9B)
1027 DATA 37,A0,D0,CD,DC,A0,CD,F8      (1E0D)
1028 DATA A0,ED,B0,C9,CD,BE,A0,21      (1E5A)
```

```
1029 DATA 80,A1,CD,5A,A1,CD,B7,A0 (1ED3)
1030 DATA 21,00,A2,CD,5A,A1,CD,B7 (1E3C)
1031 DATA A0,21,80,A2,CD,5A,A1,C9 (1E06)
1032 DATA 3A,2D,A0,5F,16,00,21,80 (1E3E)
1033 DATA A1,19,ED,5B,2F,A0,3A,2E (1E15)
1034 DATA A0,4F,06,00,C9,CD,37,A0 (1E63)
1035 DATA D0,CD,BE,A0,21,80,A1,CD (1E28)
1036 DATA 5A,A1,CD,F8,A0,EB,ED,B0 (1ED4)
1037 DATA CD,BE,A0,21,80,A1,CD,61 (1EDA)
1038 DATA A1,CD,B7,A0,21,00,A2,CD (1E2F)
1039 DATA 5A,A1,CD,F8,A0,EB,ED,B0 (1EDA)
1040 DATA CD,BE,A0,21,00,A2,CD,61 (1E65)
1041 DATA A1,CD,B7,A0,21,80,A2,CD (1E25)
1042 DATA 5A,A1,CD,F8,A0,EB,ED,B0 (1EE0)
1043 DATA CD,BE,A0,21,80,A2,CD,61 (1E0A)
1044 DATA A1,C9,DF,5E,A1,C9,92,D3 (1EB9)
1045 DATA 07,DF,65,A1,C9,68,A1,07 (1E5E)
1046 DATA E5,D5,C5,E5,11,08,00,CD (1E6D)
1047 DATA 98,CA,CD,10,D4,D2,A9,D3 (1E88)
1048 DATA EB,E3,CD,F3,D9,C3,A6,D3 (1E9B)
1049 DATA 00                      (090A)
100 '*************************** (2234)
110 '*        MCLOADER 2         * (2227)
120 '*   ERZEUGT >>MC2.BIN<<     * (22B4)
130 '*     FUER MINIDATEI        * (22C4)
140 '*          VON              * (22A7)
150 '*      MARKUS WILDI         * (228A)
160 '*          FUER             * (223C)
170 '*   SCHNEIDER CPC-WELT      * (225B)
180 '*    CPC 464/664/6128   JE* (226A)
190 '*************************** (22E8)
650 a=&A690:e=&A6F7:zb=1000:e=e+1 (2C14)
660 FOR i=a TO e:IF i=e THEN SAVE"
mc2.bin",b,&A690,&67:END          (3916)
670 READ d$:POKE i,VAL("&"+d$)    (1D5A)
730 IF i<e THEN NEXT i            (1526)
1001 DATA EF,99,A6,EF,B0,A6,EF,C7 (1EB5)
1002 DATA A6,CD,E1,A6,D5,E5,7E,02 (1EDA)
1003 DATA CD,F9,0B,03,15,20,F7,E1 (1EFD)
1004 DATA D1,CD,13,0C,1D,20,ED,C9 (1EBA)
1005 DATA CD,E1,A6,D5,E5,0A,77,CD (1E24)
1006 DATA F9,0B,03,15,20,F7,E1,D1 (1ECB)
1007 DATA CD,13,0C,1D,20,ED,C9,CD (1E1C)
1008 DATA E1,A6,D5,E5,0A,5F,7E,02 (1EF7)
1009 DATA 73,CD,F9,0B,03,15,20,F4 (1E48)
1010 DATA E1,D1,CD,13,0C,1D,20,EA (1E4A)
1011 DATA C9,D5,DD,7E,02,CD,E8,10 (1E1B)
1012 DATA F5,CD,56,12,CD,95,0B,F1 (1E79)
1013 DATA E5,CD,E8,10,E1,C1,C9,00 (1EB0)
```



MOTORWAY

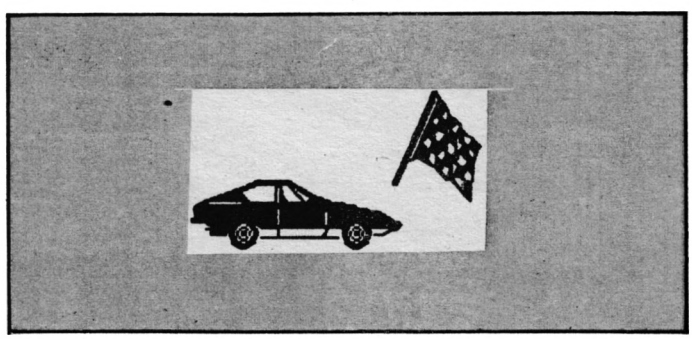
Vollgas ist angesagt

Dieses Spiel, eine Rennsimulation, wird sicher allen Autofreaks unter unseren Lesern gefallen. Sie sollten sich jedoch diese Beschreibung genau durchlesen, um mit der Tipparbeit keine Schwierigkeiten zu haben. Wie immer ist auch dieses Programm in der lauffähigen Version auf unserer Softbox enthalten.

Mit Hilfe des Programms >MOTORDAT.ERZ< wird die Datei >MOTORDAT.< erzeugt. Sie enthält die Werte für die zu symbolisierenden Zeichen.
Nach dem Starten des Programms >MOTORBIL.ERZ< entsteht das Titelbild für das Vorprogramm >MOTORVOR.BAS<, welches unter dem Namen >MOTORBIL.BIN< automatisch abgespeichert wird.
Das Erzeugungsprogramm >MOTORWAD.ERZ< stellt eine Datei >MOTORWAY.DAT< her, die die Eintragungstabelle enthält. Alle drei Programme müssen nur einmal vor dem ersten Spiel gestartet werden. Danach sind sie überflüssig, es sei denn, durch das Programm >MOTORAD.ERZ< soll, da die Rennstrecke geändert wurde, wieder eine neue Eintragungstabelle erstellt werden.

Start des Spiels:

>MOTORWAY< läuft nur auf dem CPC 6128. Da die



zweite 64 kByte-Bank genutzt wird, muß vor dem Start das Programm >BANKMAN< geladen werden.
Nun kann das Vorprogramm >MOTORVOR< gestartet werden. Hierbei gelten folgende Tastenbelegungen:

a) >1< und >2< Lautstärkeregulation der Musik während des Titelbildes
b) >RETURN< Musik aus/an
c) >Leertaste< Hauptprogramm starten.

Nachdem Sie die Leertaste gedrückt haben, wird das Hauptprogramm >MOTORWAY.BAS< ab Zeile 80

```
10 '*****************************  <2397>
11 '*   MOTORWAY-VORPROGRAMM    *  <23EB>
12 '*            VON            *  <2351>
13 '*       HOLGER HAUBER       *  <2365>
14 '*            FUER           *  <23FC>
15 '*    SCHNEIDER CPC-WELT     *  <23E3>
20 '*         CPC 6128       JE*  <23B3>
30 '*****************************  <23BF>
40 '                                <075B>
50 MODE 0:INK 0,0:INK 12,0:BORDER
0:PAPER 0:PEN 1                     <1BEC>
60 ON ERROR GOTO 810                <0B0A>
70 INK 1,15:INK 2,6:INK 3,24:TAG    <17F8>
80 MOVE 116,200,2,1:PRINT"Please w
ait !";                             <2187>
90 MOVE 112,196,1,1:PRINT"Please w
ait !";                             <2106>
100 DIM d(50,10):DIM g(50),sou1(48
),sou2(44),lan(44),i1(12):SYMBOL A
FTER 32:n=1                         <5462>
110 ENV 1,5,3,4,5,-3,8:ENV 2,3,5,1
,1,0,10,2,-1,9,13,-1,20             <3497>
120 RESTORE 910                     <094A>
130 FOR i=1 TO 48:READ sou1(i):NEX
T i                                 <249B>
140 FOR i=1 TO 43:READ sou2(i),lan
(i):NEXT i                          <3110>
150 OPENIN"motordat"                <1071>
160 WHILE NOT EOF                   <09F2>
170 INPUT#9,g(n),d(n,1),d(n,2),d(n
,3),d(n,4),d(n,5),d(n,6),d(n,7),d(
n,8)                                <7B2C>
180 n=n+1                           <1044>
190 WEND                            <06D2>
200 CLOSEIN                         <06B3>
210 FOR i=1 TO n-1                  <138C>
220 SYMBOL g(i),d(i,1),d(i,2),d(i,
3),d(i,4),d(i,5),d(i,6),d(i,7),d(i
,8)                                 <7806>
230 NEXT i                          <0A3D>
240 FOR i=0 TO 232 STEP 4           <1160>
250 MOVE i+88,204,0,0:DRAW i+88,18
0:MOVE i*-1+552,204,0,0:DRAW i*-1+
552,180                             <4492>
260 NEXT i:CLS                      <0C00>
270 FOR i=0 TO 15:INK i,0:NEXT i    <1D59>
280 LOAD"motorbil"                  <10DA>
290 |SCREENSWAP,1,2                 <15D0>
300 CLS                             <0683>
310 m(1)=1:m(2)=2:m(3)=4:m(4)=8     <2C19>
320 MODE 0:PRINT CHR$(23);CHR$(3);
:TAG                                <1800>
330 FOR p=1 TO 4                    <0E9C>
340 GRAPHICS PEN m(p),1             <1398>
350 LOCATE#1,1,25:PRINT#1,CHR$(48+
p);                                 <1EEF>
360 FOR x=0 TO 7                    <0ED6>
370 FOR y=0 TO 14 STEP 2            <11E0>
380 BORDER INT(RND*25)              <0FE8>
390 IF TEST(x*4,y)=0 THEN 420       <1B3B>
400 MOVE(x+6)*32,(y+6)*16:PRINT CH
R$(143);                            <266A>
410 MOVE(x+6)*32,(y+7)*16:PRINT CH
R$(143);                            <263F>
420 NEXT y,x,p                      <1405>
430 BORDER 0                        <07AA>
440 FOR i=1 TO 500:NEXT i           <16B2>
450 LOCATE#1,1,25:PRINT#1," ";      <1631>
460 FOR p=4 TO 1 STEP-1             <1134>
470 FOR i=0 TO 15                   <0F8B>
480 IF(i AND m(p))=0 THEN INK i,0
ELSE INK i,26                       <2B26>
490 NEXT i:FOR k=1 TO 1000:NEXT k:
NEXT p                              <22C1>
500 CLS:FOR i=1 TO 4:GRAPHICS PEN
i,0:NEXT i                          <1F40>
510 i1(1)=26:i1(2)=24:i1(3)=15:i1(
4)=6:i1(5)=4:i1(6)=8:i1(7)=2:i1(8)
=1:i1(9)=13:i1(10)=7:i1(11)=16      <84EF>
520 ii1(1)=10:ii1(2)=12:ii1(3)=18   <2BD8>
530 FOR i=0 TO 15:INK i,0:NEXT i    <1D62>
540 |SCREENSWAP,1,2                 <15C5>
550 onoff=1:n=48:n2=44:ton=12:ton2
=12                                 <35B1>
560 ON SQ(1)GOSUB 830               <0D9B>
570 ON SQ(2)GOSUB 870               <0D71>
580 EVERY 12,3 GOSUB 720            <0ED5>
590 RESTORE 680:ORIGIN 0,0,128,512
,50,0:TAG:FRAME                     <1ED0>
600 READ lau$:IF lau$="*"THEN lau$
="":FOR i=0 TO 9:READ a:lau$=lau$+
CHR$(a):NEXT i ELSE IF lau$="ENDE"
THEN 590                            <639E>
610 lau$=lau$+CHR$(32)              <1902>
620 FOR i=512 TO 128-(LEN(lau$)-1)
*32 STEP-4                          <26B3>
630 MOVE i,34,13,0:PRINT lau$;:NEX
T i                                 <21BA>
640 READ lau$:IF lau$="*"THEN lau$
="":FOR i=0 TO 9:READ a:lau$=lau$+
CHR$(a):NEXT i ELSE IF lau$="ENDE"
THEN 590                            <63EE>
650 lau$=CHR$(32)+lau$              <19B1>
660 FOR i=128-LEN(lau$)*32 TO 488
STEP 4                              <216D>
670 MOVE i,34,13,0:PRINT lau$;:NEX
T i:GOTO 600                        <26D1>
680 DATA copyright,by the,*,32,205
,206,207,208,209,206,210,211,32,an
d,Holger,Hauber,in 1986/87,*****    <6259>
690 DATA special,greetings,to the,
manager of,*,212,213,214,215,215,2
16,217,214,218,215,OLIVER,ABRECHT,
()()()                              <6A61>
700 DATA *,45,205,206,207,208,209,
206,210,211,45,Gustav,Stresemann,r
```

```
oad 53,7530,Pforzheim,GERMANY,/\/\
/\                                  <66C5>
710 DATA ENDE                       <0BA0>
720 FOR k=1 TO 11:INK k,i1(k):NEXT
 k                                  <270F>
730 FOR m=11 TO 1 STEP-1:i1(m+1)=i
1(m):NEXT m:i1(1)=i1(12)            <45D4>
740 FOR k1=1 TO 3:INK k1+12,ii1(k1
):NEXT k1                           <2E54>
750 FOR m1=3 TO 1 STEP-1:ii1(m1+1)
=ii1(m1):NEXT m1:ii1(1)=ii1(4)      <4B71>
760 IF INKEY(47)=0 OR JOY(0)=16 TH
EN CHAIN"motorway",80               <261A>
770 IF INKEY(64)=0 AND ton<>0 THEN
 ton2=ton2-1:IF ton2<8 THEN ton2=8
:ton=ton2 ELSE ton=ton2             <5D4A>
780 IF INKEY(65)=0 AND ton<>0 THEN
 ton2=ton2+1:IF ton2>14 THEN ton2=
14:ton=ton2 ELSE ton=ton2           <5F52>
790 IF INKEY(18)=0 THEN onoff=onof
f XOR 1:IF onoff=1 THEN ON SQ(1)GO
SUB 830:ON SQ(2)GOSUB 870           <4052>
800 RETURN                          <066A>
810 MODE 2:PEN 1:PAPER 0:BORDER 1:
INK 1,24:INK 0,1:CLEAR INPUT:PRINT
"Fehler ";ERR;" in Zeile ";ERL      <3BC6>
820 END                             <06CD>
830 n=n+1:IF n>48 THEN n=1:n2=1:to
n=INT(RND*2)*ton2                   <4086>
840 SOUND 1,sou1(n),25,9,0,1        <1EA1>
850 IF onoff=1 THEN ON SQ(1)GOSUB
830                                 <19D2>
860 RETURN                          <06E2>
870 n2=n2+1:IF n2>43 THEN n2=1:ton
=0                                  <2DAB>
880 SOUND 2,sou2(n2),lan(n2),ton,0
,2:SOUND 4,sou2(n2)+1,lan(n2),ton,
0,1                                 <5C0E>
890 IF onoff=1 THEN ON SQ(2)GOSUB
870                                 <19DC>
900 RETURN                          <0631>
910 DATA 239,213,239,213,239,213,2
39,213,239,213,239,213,239,213,239
,213                                <46BC>
920 DATA 179,159,179,159,179,159,1
79,159,239,213,239,213,239,213,239
,213                                <46DE>
930 DATA 159,142,159,142,179,159,1
79,159,239,213,239,213,239,213,239
,213                                <463A>
940 DATA 142,12.5,142,12.5,106,25,
113,25,142,25,179,12.5,179,12.5,14
2,25,159,25,169,12.5,213,25,213,12
.5,213,100,0,75                     <733C>
950 DATA 142,12.5,142,12.5,142,25,
159,25,169,25,213,12.5,213,12.5,14
2,25,159,25,169,12.5,213,25,213,12
.5,213,100,0,75                     <735C>
960 DATA 142,12.5,142,12.5,113,12.
5,142,12.5,119,12.5,119,12.5,142,5
0,142,12.5,142,12.5,142,12.5,159,3
7.5,169,12.5,213,12.5,213,100,0,75
                                    <86EC>

10 '*****************************  <2397>
11 '*  MOTORWAY-HAUPTPROGRAMM  *   <2351>
12 '* ABSPEICHERN ALS<MOTORWAY>*   <23C7>
13 '*           VON             *  <2353>
14 '*       HOLGER HAUBE        *  <23AE>
15 '*          FUER             *  <2370>
20 '*    SCHNEIDER CPC-WELT     *  <23ED>
30 '*         CPC 6128          *  <23E9>
35 '*****************************  <23C9>
38 '                                <0757>
40 ' *****                          <0D6F>
50 ' SYMBOLE                        <0F97>
60 ' *****                          <0D97>
70 RUN"motorvor"                    <101F>
80 MODE 0                           <0744>
90 FOR i=0 TO 15:INK i,0:NEXT i:BO
RDER 1                              <20F6>
100 SYMBOL AFTER 160                <0900>
110 SYMBOL 160,&X1101110,&X101010,
&X101010,&X1001010,&X1101110,0,0,0
                                    <22A2>
120 SYMBOL 161,&X1101110,&X1001010
,&X1101010,&X1101010,&X1101110,0,0
,0                                  <22A9>
130 SYMBOL 162,&X10110110,&X101101
10,&X10110110,&X10110110,&X1011011
0,0,0,0                             <221E>
140 SYMBOL 163,&X10100110,&X101001
10,&X10110110,&X10010110,&X1001011
0,0,0,0                             <2238>
150 SYMBOL 164,&X10111011,&X101010
11,&X10010011,&X10101011,&X1011101
1,0,0,0                             <22AF>
160 SYMBOL 165,&X11011011,&X100101
1,&X1001011,&X10010011,&X11011011,
0,0,0                               <22F6>
170 SYMBOL 167,0,0,&X10000000,&X10
000000,&X10101101,&X11001010,&X101
01010,0                             <2289>
180 SYMBOL 168,0,&X1000,&X1010,&X1
0010,&X10011,&X10100010,&X10100010
,0                                  <249C>
190 SYMBOL 169,0,0,0,0,&X10000000,
&X10000000,&X10000000,0             <1E97>
200 SYMBOL 170,&X11110,&X10010,&X1
0010,&X10010,&X100100,&X100100,&X1
00100,&X111100                      <287E>
210 SYMBOL 171,&X10,&X10,&X10,&X10
,&X100,&X100,&X100,&X100            <2802>
220 SYMBOL 172,&X11110,&X10,&X10,&
X111110,&X100000,&X100000,&X100000
,&X111100                           <28BB>
```

```
230 SYMBOL 173,&X11110,&X10,&X10,&
X11110,&X100,&X100,&X100,&X111100 <285D>
240 SYMBOL 174,&X10010,&X10010,&X1
0010,&X11110,&X100,&X100,&X100,&X1
00                                <28CE>
250 SYMBOL 175,&X11110,&X10000,&X1
0000,&X11100,&X100,&X100,&X100,&X1
11100                             <281A>
260 SYMBOL 176,&X11110,&X10000,&X1
0000,&X11100,&X100100,&X100100,&X1
00100,&X111100                    <2844>
270 SYMBOL 177,&X11110,&X10010,&X1
0010,&X10010,&X100,&X100,&X100,&X1
00                                <284C>
280 SYMBOL 178,&X11110,&X10010,&X1
0010,&X11110,&X100100,&X100100,&X1
00100,&X111100                    <2820>
290 SYMBOL 179,&X11110,&X10010,&X1
0010,&X11110,&X100,&X100,&X100,&X1
11100                             <28FC>
300 SYMBOL 180,0,0,&X1110,&X100,&X
100,&X100,&X100,0                 <22BF>
310 SYMBOL 181,0,0,&X10101011,&X10
111010,&X10101011,&X10101010,&X101
01011,0                           <2236>
320 SYMBOL 182,0,&X100000,&X100000
0,&X1001010,&X11101010,&X1001010,&
X1001010,&X1000100                <262D>
330 SYMBOL 183,0,&X1000,&X1000,&X1
001000,&X10101000,&X11101000,&X100
01000,&X1100100                   <2644>
340 SYMBOL 184,0,&X1000000,&X10000
00,&X11100100,&X1001010,&X1001110,
&X1001000,&X100110                <264E>
350 SYMBOL 185,0,0,0,&X11010000,&X
10101000,&X10101000,&X10101000,&X1
0101010                           <221B>
360 SYMBOL 186,0,0,&X110110,&X1000
100,&X1010110,&X1010100,&X110110,0
                                  <2291>
370 SYMBOL 187,0,0,&X11101100,&X10
101010,&X11101100,&X10101010,&X101
01010,0                           <224B>
380 SYMBOL 188,0,0,0,0,0,&X1000000
,&X1000000,0                      <1CD5>
390 SYMBOL 189,0,&X1010101,&X10101
11,&X1100101,&X1010101,&X1010101,0
,0                                <2283>
400 SYMBOL 190,&X1000000,&X1010000
0,&X10100000,&X10100000,&X10100000
,&X10100000,&X1000000,0           <26F4>
410 SYMBOL 191,&X100000,&X1100000,
&X10100000,&X100000,&X100000,&X100
000,&X100000,0                    <2653>
420 SYMBOL 192,&X1000000,&X1010000
0,&X100000,&X100000,&X1000000,&X10
000000,&X11100000,0               <26DB>
430 SYMBOL 193,&X11000000,&X100000
,&X100000,&X1000000,&X100000,&X100
000,&X11000000,0                  <26D3>
440 SYMBOL 194,&X1000000,&X1000000
,&X10000000,&X10100000,&X11100000,
&X100000,&X100000,0               <2667>
450 SYMBOL 195,&X11100000,&X100000
00,&X10000000,&X11000000,&X100000,
&X100000,&X11000000,0             <2639>
460 SYMBOL 196,&X1100000,&X1000000
0,&X10000000,&X11000000,&X10100000
,&X10100000,&X1000000,0           <2609>
470 SYMBOL 197,&X11100000,&X100000
,&X100000,&X1000000,&X1000000,&X10
000000,&X10000000,0               <2601>
480 SYMBOL 198,&X1000000,&X1010000
0,&X10100000,&X1000000,&X10100000,
&X10100000,&X1000000,0            <26FE>
490 SYMBOL 199,&X1000000,&X1010000
0,&X10100000,&X1100000,&X100000,&X
10100000,&X1000000,0              <26A8>
500 SYMBOL 200,&X11101010,&X100101
0,&X1001010,&X1001110,0,0,0,0     <2071>
510 SYMBOL 201,&X11001010,&X101011
10,&X11001110,&X10101010,0,0,0,0  <20C7>
520 SYMBOL 202,&X11,&X1111,&X11111
,&X111111,&X1111111,&X1111111,&X11
111111,&X11111111                 <2857>
530 SYMBOL 203,&X11000000,&X111100
00,&X11111000,&X11111100,&X1111111
0,&X11111110,&X11111111,&X11111111
                                  <2833>
540 SYMBOL 204,&X111,&X1111,&X1111
,&X1111,&X11111,&X11111,&X11111,&X
111111                            <28DE>
550 SYMBOL 205,&X11110000,&X111110
00,&X11111000,&X11111000,&X1111010
0,&X11110100,&X11110100,&X11101100
                                  <2856>
560 SYMBOL 206,&X1111,&X11111,&X11
111,&X11111,&X101111,&X101111,&X10
1111,&X110111                     <283D>
570 SYMBOL 207,&X11100000,&X111100
00,&X11110000,&X11110000,&X1111100
0,&X11111000,&X11111000,&X11111100
                                  <289F>
580 SYMBOL 208,0,0,&X11001010,&X10
011011,&X11011011,&X10011011,&X110
10010,0                           <2281>
590 DIM x2(32),y2(32),ii2(32),einm
(35),eins(35),ein$(35)            <471D>
600 OPENIN"motorway.dat":FOR k=1 T
O 35:INPUT#9,einm(k),eins(k),ein$(
k):NEXT k:CLOSEIN                 <5491>
610 DEG:ORIGIN 0,0                <0B41>
620 PAPER 0:WINDOW 1,20,1,19:WINDO
W#1,1,20,20,25:PAPER#1,10:CLS#1   <2C87>
630 GOSUB 4930                    <091F>
640 ON ERROR GOTO 7470            <0B1B>
```

```
650 GOSUB 760                                   (09B2)
660 /SCREENCOPY,2,1                             (1562)
670 GOSUB 1380                                  (0942)
680 /SCREENCOPY,3,1                             (159A)
690 GOSUB 2080                                  (095C)
700 /SCREENCOPY,4,1                             (15D2)
710 GOSUB 2710                                  (0938)
720 /SCREENCOPY,5,1                             (150B)
730 GOSUB 3400                                  (09FB)
740 q3(1)=3:q3(2)=2:q3(4)=4:q3(5)=
5:q3(3)=1                                       (3B0D)
750 GOTO 3910                                   (093A)
760 ' *****                                     (0D11)
770 ' BILD 1                                    (0E8F)
780 ' *****                                     (0D3A)
790 CLS                                         (0659)
800 MOVE 0,96,1:DRAW 88,96:MOVE 54
8,96:DRAW 640,96                                (2385)
810 MOVE 0,120,1:DRAW 640,132'124               (19B9)
820 MOVE 0,122,2:DRAW 640,134'126               (194F)
830 MOVE 0,150,2:DRAW 640,168'156               (193F)
840 MOVE 0,152,3:DRAW 640,170'158               (19E2)
850 MOVE 0,177,3:DRAW 640,201'185               (194D)
860 MOVE 0,179,4:DRAW 640,203'187               (19E4)
870 MOVE 0,197,4:DRAW 640,227'207               (196C)
880 MOVE 0,199,1:DRAW 640,229'209               (191C)
890 MOVE 0,213,1:DRAW 640,249'225               (19CF)
900 MOVE 0,215,2:DRAW 640,251'227               (1989)
910 MOVE 0,226,2:DRAW 640,268'240               (1A00)
920 MOVE 0,228,3:DRAW 640,270'242               (1A0E)
930 MOVE 0,236,3:DRAW 640,284'252               (1A7F)
940 MOVE 0,238,4:DRAW 640,286'254               (1ABD)
950 MOVE 0,245,4:DRAW 640,299'263               (1AC6)
960 MOVE 0,247,1:DRAW 640,301'265               (1AE3)
970 MOVE 0,250,1:DRAW 640,310'270               (1ADA)
980 MOVE 0,252,2:DRAW 640,312'272               (1A58)
990 MOVE 0,256,2:DRAW 640,322'278               (1B9F)
1000 MOVE 0,258,3:DRAW 640,324'280
                                                (1B78)
1010 MOVE 0,258,3:DRAW 640,330'282
                                                (1B9A)
1020 MOVE 0,260,4:DRAW 640,332'284
                                                (1BB9)
1030 MOVE 0,262,4:DRAW 640,340'288
                                                (1B31)
1040 MOVE 0,264,1:DRAW 640,342'290
                                                (1B50)
1050 MOVE 0,264,1:DRAW 640,348'292
                                                (1B76)
1060 MOVE 88,96,7:DRAW 548,96                   (1544)
1070 MOVE 1,100:FILL 1                          (0D64)
1080 MOVE 1,140:FILL 2                          (0D85)
1090 MOVE 1,170:FILL 3                          (0D61)
1100 MOVE 1,180:FILL 4                          (0DBA)
1110 MOVE 1,205:FILL 1                          (0DC5)
1120 MOVE 1,220:FILL 2                          (0DBF)
1130 MOVE 1,230:FILL 3                          (0D15)
1140 MOVE 1,242:FILL 4                          (0DB0)
1150 MOVE 320,276:FILL 1                        (104D)
1160 MOVE 320,284:FILL 2                        (1067)
1170 MOVE 320,300:FILL 4                        (1087)
1180 MOVE 180,282:FILL 4                        (0F61)
1190 RESTORE 1210                               (0908)
1200 GOSUB 4810                                 (09D1)
1210 DATA -134,351,339,95,140,7                 (1CB4)
1220 DATA -134,-692,1045,15,45,7                (1D35)
1230 MOVE 320,100:FILL 7                        (0F11)
1240 GOSUB 4810                                 (0920)
1250 DATA -482,871,886,130,150,0                (1DF0)
1260 DATA -474,-2007,2404,15,30,0               (1E5A)
1270 MOVE 1,150:FILL 0:MOVE 639,15
0:FILL 0                                        (186B)
1280 MOVE 0,300,6:DRAW 640,300,6                (186D)
1290 MOVE 320,350:FILL 6                        (10CB)
1300 PLOT 248,280,2                             (0E9B)
1310 MOVE 244,282,3:DRAW 236,284                (16C1)
1320 MOVE 232,286,4:DRAW 226,288                (16FA)
1330 MOVE 222,290,1:DRAW 214,292                (16A7)
1340 MOVE 210,294,2:DRAW 202,296                (1630)
1350 MOVE 194,296,2:DRAW 190,290                (1662)
1360 MOVE 194,296,2:DRAW 194,292                (1685)
1370 RETURN                                     (06E0)
1380 ' *****                                    (0DEC)
1390 ' BILD 2                                   (0E6E)
1400 ' *****                                    (0D13)
1410 CLS                                        (0634)
1420 MOVE 0,96,1:DRAW 88,96:MOVE 5
44,96:DRAW 640,96                               (2351)
1430 MOVE 0,122,1:DRAW 640,140'124
                                                (1994)
1440 MOVE 0,124,2:DRAW 640,142'126
                                                (19CA)
1450 MOVE 0,152,2:DRAW 640,180'156
                                                (1924)
1460 MOVE 0,154,3:DRAW 640,182'158
                                                (1949)
1470 MOVE 0,180,3:DRAW 640,217'185
                                                (1938)
1480 MOVE 0,182,4:DRAW 640,219'187
                                                (19C3)
1490 MOVE 0,201,4:DRAW 640,247'207
                                                (1942)
1500 MOVE 0,203,1:DRAW 640,249'209
                                                (1982)
1510 MOVE 0,218,1:DRAW 640,273'225
                                                (1A5D)
1520 MOVE 0,220,2:DRAW 640,275'227
                                                (1AD8)
1530 MOVE 0,232,2:DRAW 640,296'240
                                                (1AE5)
1540 MOVE 0,234,3:DRAW 640,298'242
                                                (1AFD)
1550 MOVE 0,242,3:DRAW 640,316'252
                                                (1A83)
1560 MOVE 0,244,4:DRAW 640,318'254
                                                (1A77)
```

```
1570 MOVE 0,252,4:DRAW 640,335'263
                                        (1A0D)
1580 MOVE 0,254,1:DRAW 640,337'265
                                        (1A3C)
1590 MOVE 0,258,1:DRAW 640,350'270
                                        (1B46)
1600 MOVE 0,260,2:DRAW 640,352'272
                                        (1B4F)
1610 MOVE 0,265,2:DRAW 640,366'278
                                        (1BD7)
1620 MOVE 0,267,3:DRAW 640,368'280
                                        (1BC6)
1630 MOVE 0,268,3:DRAW 640,378'282
                                        (1B09)
1640 MOVE 0,270,4:DRAW 640,380'284
                                        (1B53)
1650 MOVE 0,272,4:DRAW 640,392'288
                                        (1B93)
1660 MOVE 0,274,1:DRAW 640,394'290
                                        (1B70)
1670 MOVE 0,275,1:DRAW 640,404'292
                                        (1B8E)
1680 MOVE 88,96,7:DRAW 544,96           (1522)
1690 MOVE 639,130:FILL 1                (0FD7)
1700 MOVE 639,150:FILL 2                (0F70)
1710 MOVE 639,190:FILL 3                (0F93)
1720 MOVE 639,230:FILL 4                (0FAA)
1730 MOVE 639,260:FILL 1                (10D0)
1740 MOVE 639,280:FILL 2                (10EF)
1750 MOVE 639,300:FILL 3                (100B)
1760 MOVE 639,330:FILL 4                (106A)
1770 MOVE 639,340:FILL 1                (10F4)
1780 MOVE 639,360:FILL 2                (100E)
1790 MOVE 639,370:FILL 3                (10AC)
1800 MOVE 80,280:FILL 3                 (0F2A)
1810 MOVE 88,282:FILL 3                 (0FD6)
1820 MOVE 639,385:FILL 4                (10B1)
1830 MOVE 80,286:FILL 4                 (0FEB)
1840 RESTORE 1860                       (0980)
1850 GOSUB 4810                         (09E7)
1860 DATA 3,194,130,35,140,7            (1997)
1870 DATA 3,-535,837,5,45,7             (1816)
1880 MOVE 88,96,7:DRAW 544,96           (15B4)
1890 MOVE 320,100:FILL 7                (0F3B)
1900 GOSUB 4810                         (094C)
1910 DATA -173,314,249,90,135,0         (1C36)
1920 DATA -165,-1450,1767,5,30,0        (1D5D)
1930 MOVE 1,150:FILL 0:MOVE 639,15
0:FILL 0                                (1895)
1940 MOVE 0,300,6:DRAW 640,300,6        (1898)
1950 MOVE 320,350:FILL 6                (10F6)
1960 MOVE 260,258,3:DRAW 236,266        (1729)
1970 MOVE 232,268,4:DRAW 200,276        (16F0)
1980 MOVE 196,276,1:DRAW 176,280        (16ED)
1990 MOVE 172,282,2:DRAW 152,286        (1622)
2000 MOVE 148,288,3:DRAW 136,290        (16BD)
2010 MOVE 132,290,4:DRAW 120,292        (1691)
2020 MOVE 116,292,1:DRAW 104,294        (1637)
2030 PLOT 80,292,1                      (0EA6)
2040 MOVE 76,292,2:DRAW 76,296          (1664)
2050 MOVE 76,296,2:DRAW 80,294          (16E9)
2060 MOVE 88,296,2:DRAW 100,296         (161F)
2070 RETURN                             (065B)
2080 ' *****                            (0D67)
2090 ' BILD 3                           (0EEC)
2100 ' *****                            (0D8F)
2110 CLS                                (06AE)
2120 MOVE 0,96,1:DRAW 88,96:MOVE 5
48,96:DRAW 640,96                       (23DA)
2130 MOVE 640,120,1:DRAW 0,132'124
                                        (1907)
2140 MOVE 640,122,2:DRAW 0,134'126
                                        (19C5)
2150 MOVE 640,150,2:DRAW 0,168'156
                                        (194D)
2160 MOVE 640,152,3:DRAW 0,170'158
                                        (198F)
2170 MOVE 640,177,3:DRAW 0,201'185
                                        (1903)
2180 MOVE 640,179,4:DRAW 0,203'187
                                        (1948)
2190 MOVE 640,197,4:DRAW 0,227'207
                                        (195B)
2200 MOVE 640,199,1:DRAW 0,229'209
                                        (190F)
2210 MOVE 640,213,1:DRAW 0,249'225
                                        (1910)
2220 MOVE 640,215,2:DRAW 0,251'227      (1962)
2230 MOVE 640,226,2:DRAW 0,268'240
                                        (1A4E)
2240 MOVE 640,228,3:DRAW 0,270'242
                                        (1A92)
2250 MOVE 640,236,3:DRAW 0,284'252      (1ADC)
2260 MOVE 640,238,4:DRAW 0,286'254
                                        (1A45)
2270 MOVE 640,245,4:DRAW 0,299'263
                                        (1A7F)
2280 MOVE 640,247,1:DRAW 0,301'265
                                        (1AD0)
2290 MOVE 640,250,1:DRAW 0,310'270
                                        (1AD0)
2300 MOVE 640,252,2:DRAW 0,312'272
                                        (1A55)
2310 MOVE 640,256,2:DRAW 0,322'278
                                        (1B57)
2320 MOVE 640,258,3:DRAW 0,324'280
                                        (1B90)
2330 MOVE 640,258,3:DRAW 0,330'282
                                        (1BB2)
2340 MOVE 640,260,4:DRAW 0,332'284
                                        (1B03)
2350 MOVE 640,262,4:DRAW 0,340'288
                                        (1B64)
2360 MOVE 640,264,1:DRAW 0,342'290
                                        (1B97)
```

```
2370 MOVE 640,264,1:DRAW 0,348'292
                                        (1BB5)
2380 MOVE 88,96,7:DRAW 548,96           (1599)
2390 MOVE 639,100:FILL 1                (0F8E)
2400 MOVE 639,140:FILL 2                (0FA9)
2410 MOVE 639,170:FILL 3                (0F85)
2420 MOVE 639,180:FILL 4                (0FE0)
2430 MOVE 639,205:FILL 1                (0F0C)
2440 MOVE 639,220:FILL 2                (0F03)
2450 MOVE 639,230:FILL 3                (0F5D)
2460 MOVE 639,242:FILL 4                (0FF8)
2470 MOVE 320,276:FILL 1                (10A2)
2480 MOVE 320,284:FILL 2                (10BC)
2490 MOVE 320,298:FILL 4                (105A)
2500 MOVE 460,282:FILL 4                (109B)
2510 MOVE 448,284:FILL 4                (1002)
2520 RESTORE 2540                       (0917)
2530 GOSUB 4810                         (0939)
2540 DATA 774,-692,1045,315,345,7       (1ED0)
2550 DATA 774,351,339,220,265,7         (1C85)
2560 MOVE 320,100:FILL 7                (0F7B)
2570 GOSUB 4810                         (098A)
2580 DATA 1114,-2007,2404,330,345,
0                                       (2098)
2590 DATA 1122,871,886,210,230,0        (1DFC)
2600 MOVE 1,150:FILL 0:MOVE 639,15
0:FILL 0                                (18D5)
2610 MOVE 0,300,6:DRAW 640,300,6        (18D6)
2620 MOVE 320,350:FILL 6                (1035)
2630 PLOT 388,280,2                     (0F3E)
2640 MOVE 392,282,3:DRAW 400,284        (180A)
2650 MOVE 404,286,4:DRAW 412,288        (1876)
2660 MOVE 416,290,1:DRAW 424,292        (1840)
2670 MOVE 428,294,2:DRAW 436,296        (1837)
2680 MOVE 444,296,2:DRAW 448,290        (185E)
2690 MOVE 444,296,2:DRAW 444,292        (1890)
2700 RETURN                             (0648)
2710 ' *****                            (0D54)
2720 ' BILD 4                           (0EDF)
2730 ' *****                            (0D7C)
2740 CLS                                (069D)
2750 MOVE 0,96,1:DRAW 92,96:MOVE 5
48,96:DRAW 640,96                       (233E)
2760 MOVE 640,122,1:DRAW 0,140'124
                                        (19F7)
2770 MOVE 640,124,2:DRAW 0,142'126
                                        (19B1)
2780 MOVE 640,152,2:DRAW 0,180'156
                                        (1935)
2790 MOVE 640,154,3:DRAW 0,182'158
                                        (1983)
2800 MOVE 640,180,3:DRAW 0,217'185
                                        (196E)
2810 MOVE 640,182,4:DRAW 0,219'187
                                        (19AA)
2820 MOVE 640,201,4:DRAW 0,247'207
                                        (193D)
2830 MOVE 640,203,1:DRAW 0,249'209
                                        (19F6)
2840 MOVE 640,218,1:DRAW 0,273'225
                                        (1AC6)
2850 MOVE 640,220,2:DRAW 0,275'227
                                        (1A29)
2860 MOVE 640,232,2:DRAW 0,296'240
                                        (1A4C)
2870 MOVE 640,234,3:DRAW 0,298'242
                                        (1AA8)
2880 MOVE 640,242,3:DRAW 0,316'252
                                        (1AED)
2890 MOVE 640,244,4:DRAW 0,318'254
                                        (1A39)
2900 MOVE 640,252,4:DRAW 0,335'263
                                        (1A8C)
2910 MOVE 640,254,1:DRAW 0,337'265
                                        (1AE3)
2920 MOVE 640,258,1:DRAW 0,350'270
                                        (1BD3)
2930 MOVE 640,260,2:DRAW 0,352'272
                                        (1B34)
2940 MOVE 640,265,2:DRAW 0,366'278
                                        (1B89)
2950 MOVE 640,267,3:DRAW 0,368'280
                                        (1BBB)
2960 MOVE 640,268,3:DRAW 0,378'282
                                        (1BDB)
2970 MOVE 640,270,4:DRAW 0,380'284
                                        (1B45)
2980 MOVE 640,272,4:DRAW 0,392'288
                                        (1B89)
2990 MOVE 640,274,1:DRAW 0,394'290
                                        (1BBE)
3000 MOVE 640,275,1:DRAW 0,404'292
                                        (1BDE)
3010 MOVE 92,96,7:DRAW 548,96           (1594)
3020 MOVE 1,130:FILL 1                  (0D76)
3030 MOVE 1,150:FILL 2                  (0D0A)
3040 MOVE 1,190:FILL 3                  (0D0E)
3050 MOVE 1,230:FILL 4                  (0D1F)
3060 MOVE 1,260:FILL 1                  (0E58)
3070 MOVE 1,280:FILL 2                  (0E75)
3080 MOVE 1,300:FILL 3                  (0E93)
3090 MOVE 1,330:FILL 4                  (0E33)
3100 MOVE 1,340:FILL 1                  (0EBD)
3110 MOVE 1,360:FILL 2                  (0EDA)
3120 MOVE 1,370:FILL 3                  (0E75)
3130 MOVE 560,280:FILL 3                (1097)
3140 MOVE 552,282:FILL 3                (100B)
3150 MOVE 1,385:FILL 4                  (0E79)
3160 MOVE 556,286:FILL 4                (1049)
3170 RESTORE 3190                       (098F)
3180 GOSUB 4810                         (0951)
3190 DATA 637,-535,837,315,355,7        (1D1F)
3200 DATA 637,194,130,220,325,7         (1CF8)
3210 MOVE 320,100:FILL 7                (0F90)
3220 GOSUB 4810                         (09A1)
3230 DATA 805,-1450,1767,330,355,0
                                        (1F41)
```

```
3240 DATA 813,314,249,225,270,0    (1C51)
3250 MOVE 1,150:FILL 0:MOVE 639,15
0:FILL 0                           (18EA)
3260 MOVE 0,300,6:DRAW 640,300,6   (18ED)
3270 MOVE 320,350:FILL 6           (104A)
3280 MOVE 376,258,3:DRAW 400,266   (18B6)
3290 MOVE 404,268,4:DRAW 436,276   (1807)
3300 MOVE 440,276,1:DRAW 460,280   (1863)
3310 MOVE 464,282,2:DRAW 484,286   (189A)
3320 MOVE 488,288,3:DRAW 500,290   (18A1)
3330 MOVE 504,290,4:DRAW 516,292   (18E8)
3340 MOVE 520,292,1:DRAW 532,294   (18CF)
3350 PLOT 560,292,1                (0FB8)
3360 MOVE 564,292,2:DRAW 564,296   (183B)
3370 MOVE 564,296,2:DRAW 560,294   (1836)
3380 MOVE 548,296,2:DRAW 536,296   (1818)
3390 RETURN                        (06B0)
3400 ' *****                       (0DBC)
3410 ' BILD 5                      (0E4A)
3420 ' *****                       (0DE4)
3430 CLS                           (0603)
3440 MOVE 0,300,6:DRAW 640,300     (169B)
3450 MOVE 0,96,1:DRAW 84,96:MOVE 5
52,96:DRAW 640,96                  (23C6)
3460 MOVE 0,124,1:DRAW 116,124:MOV
E 520,124:DRAW 640,124             (230C)
3470 MOVE 1,126,2:DRAW 120,126:MOV
E 516,126:DRAW 639,126             (235A)
3480 MOVE 56,156,2:DRAW 152,156:MO
VE 484,156:DRAW 584,156            (243B)
3490 MOVE 60,158,3:DRAW 156,158:MO
VE 480,158:DRAW 580,158            (2481)
3500 MOVE 108,185,3:DRAW 184,185:M
OVE 452,185:DRAW 532,185           (24A6)
3510 MOVE 112,187,4:DRAW 188,187:M
OVE 448,187:DRAW 528,187           (24F9)
3520 MOVE 148,207,4:DRAW 212,207:M
OVE 428,207:DRAW 492,207           (2495)
3530 MOVE 152,209,1:DRAW 212,209:M
OVE 424,209:DRAW 488,209           (2433)
3540 MOVE 180,225,1:DRAW 232,225:M
OVE 408,225:DRAW 460,225           (24FD)
3550 MOVE 184,227,2:DRAW 236,227:M
OVE 404,227:DRAW 456,227           (249B)
3560 MOVE 208,240,2:DRAW 252,240:M
OVE 388,240:DRAW 432,240           (24C5)
3570 MOVE 212,242,3:DRAW 252,242:M
OVE 384,242:DRAW 428,242           (2401)
3580 MOVE 232,252,3:DRAW 264,252:M
OVE 376,252:DRAW 408,252           (25CE)
3590 MOVE 236,254,4:DRAW 268,254:M
OVE 372,254:DRAW 404,254           (251D)
3600 MOVE 252,263,4:DRAW 276,263:M
OVE 364,263:DRAW 388,263           (29C7)
3610 MOVE 252,265,1:DRAW 280,265:M
OVE 360,265:DRAW 388,265           (290E)
3620 MOVE 264,270,1:DRAW 284,270:M
OVE 356,270:DRAW 376,270           (2AB6)
3630 MOVE 268,272,2:DRAW 288,272:M
OVE 352,272:DRAW 372,272           (2A3A)
3640 MOVE 280,278,2:DRAW 296,278:M
OVE 344,278:DRAW 360,278           (2A65)
3650 MOVE 284,280,3:DRAW 296,280:M
OVE 344,280:DRAW 356,280           (2A03)
3660 MOVE 288,282,3:DRAW 300,282:M
OVE 340,282:DRAW 352,282           (2ABC)
3670 MOVE 292,284,4:DRAW 300,284:M
OVE 340,284:DRAW 348,284           (2A03)
3680 MOVE 296,288,4:DRAW 308,288:M
OVE 332,288:DRAW 344,288           (2A2E)
3690 MOVE 300,290,1:DRAW 308,290:M
OVE 332,290:DRAW 340,290           (2A93)
3700 MOVE 304,292,1:DRAW 312,292:M
OVE 328,292:DRAW 336,292           (2ADF)
3710 ' *****                       (0D28)
3720 MOVE 312,294,2:DRAW 320,300:M
OVE 328,294:DRAW 320,300           (2A37)
3730 MOVE 308,294,2:DRAW 320,300:M
OVE 332,294:DRAW 320,300           (2AEE)
3740 MOVE 86,96,7:DRAW 320,298:DRA
W 550,96:DRAW 86,96                (25E4)
3750 MOVE 0,126,7:DRAW 316,300:MOV
E 324,300:DRAW 640,126             (265D)
3760 ' *****                       (0D8C)
3770 MOVE 50,100:FILL 1:MOVE 590,1
00:FILL 1                          (19DA)
3780 MOVE 100,140:FILL 2:MOVE 540,
140:FILL 2                         (1986)
3790 MOVE 130,170:FILL 3:MOVE 510,
170:FILL 3                         (1930)
3800 MOVE 160,200:FILL 4:MOVE 480,
200:FILL 4                         (190B)
3810 MOVE 200,220:FILL 1:MOVE 440,
220:FILL 1                         (1931)
3820 MOVE 220,230:FILL 2:MOVE 420,
230:FILL 2                         (1936)
3830 MOVE 240,245:FILL 3:MOVE 400,
245:FILL 3                         (1998)
3840 MOVE 260,260:FILL 4:MOVE 380,
260:FILL 4                         (1CD2)
3850 MOVE 270,268:FILL 1:MOVE 370,
268:FILL 1                         (1C91)
3860 MOVE 285,274:FILL 2:MOVE 355,
274:FILL 2                         (1CE3)
3870 MOVE 300,286:FILL 4:MOVE 342,
286:FILL 4                         (1CC2)
3880 MOVE 320,100:FILL 7           (0FD0)
3890 MOVE 320,350:FILL 6           (1026)
3900 RETURN                        (06AE)
3910 ' *****                       (0DBA)
3920 ' ZEIGER AUFBAU               (1592)
3930 ' *****                       (0DE2)
3940 INK 0,12:INK 1,26:INK 2,26:IN
K 3,6:INK 4,6:INK 5,7:INK 6,2:INK
7,13:INK 8,8:INK 9,0:INK 10,4:INK
11,15:INK 12,24:INK 13,26:INK 14,1
```

```
3:INK 15,26                                    <633D>
3950 ORIGIN 0,0,0,640,0,400                    <15D0>
3960 ORIGIN 320,30:MOVE 0,0,4,1:DR
AW 68*SIN(252),44*COS(252)                     <2A9F>
3970 ORIGIN 468,20:MOVE 44*SIN(400
),44*COS(400),11,1:DRAW 58*SIN(400
),58*COS(400)                                  <3FF4>
3980 ORIGIN 580,20:MOVE 44*SIN(320
),44*COS(320),11,1:DRAW 58*SIN(320
),58*COS(320)                                  <3FFE>
3990 ORIGIN 0,0:TAG:MOVE 16,40,6,1
:PRINT CHR$(129);:TAGOFF                       <219F>
4000 PAPER#1,10:PEN#1,8,0:LOCATE#1
,3,2:PRINT#1,CHR$(170)CHR$(170)"."
CHR$(170)CHR$(170)                             <3B61>
4010 TAG:FOR i=458 TO 490 STEP 16:
MOVE i,22,1,1:PRINT CHR$(190);:NEX
T i:MOVE 506,22:PRINT CHR$(188);:F
OR i=522 TO 538 STEP 16:MOVE i,22,
1,1:PRINT CHR$(190);:NEXT i:TAGOFF
                                               <736A>
4020 TAG:MOVE 80,24,8,1:PRINT CHR$
(143);:TAGOFF                                  <1C39>
4030 ORIGIN 0,112:TAG:MOVE 320-32,
0,2,1:PRINT CHR$(202)CHR$(203);:MO
VE 320-96,0,14,1:PRINT CHR$(204)CH
R$(205);:MOVE 320+32,0:PRINT CHR$(
206)CHR$(207);:TAGOFF                          <62A8>
4040 ORIGIN 0,0:PLOT 320,10,4,1:PL
OT 324,10,4,1                                  <2152>
4050 RESTORE 4760                              <099A>
4060 IF INKEY(47)<>0 AND JOY(0)<>1
6 THEN 4060                                    <1BBF>
4070 ' *****                                   <0DF9>
4080 ' SCHLEIFE                                <1008>
4090 ' *****                                   <0D22>
4100 v=0:w=0:g=1:t=0:b=40:h=0:s=0:
s0=400:x=0:a=0:ue=100:lr=1/50000:a
0=25:bv=0.2:hc=2/15000:kc=2/45000:
ab=10:t=TIME/300:t0=t:ws=0:i(1)=26
:i(2)=26:i(3)=6:i(4)=6:dts=32000               <F46A>
4110 lk=1:lrk=0.5:hrk=10000:kw=0.2
:lp=0:le=0:le1=0:l=0:q1=3                      <5A8E>
4120 v1=v:w1=w:g1=g:t1=t:b1=b:h1=h
:s1=s:lp1=lp                                   <5E53>
4130 IF dts>s0/(v+0.0125)+1.5 THEN
 dts=s0/(v+0.0125)+1.5:AFTER dts G
OSUB 4560                                      <5506>
4140 a=0:x=0                                   <1241>
4150 dt=MIN(TIME/300-t,1.8):t=TIME
/300                                           <2D60>
4160 IF INKEY(0)=0 OR JOY(0)=1 THE
N x=1:a=a0*2^(1-g)*(15*w/20000-w*w
/6250000+0.25)*(1-ABS((75-h)/150))
:b=b-bv*dt                                     <8325>
4170 IF INKEY(2)=0 OR JOY(0)=2 THE
N x=-1:a=-ab                                   <2973>
4180 IF INKEY(64)=0 AND v<50 THEN
g=1:GOSUB 4520                                 <22F3>
4190 IF INKEY(65)=0 AND v<100 THEN
 g=2:GOSUB 4520                                <22B4>
4200 IF INKEY(57)=0 AND v<200 THEN
 g=3:GOSUB 4520                                <2203>
4210 IF INKEY(56)=0 THEN g=4:GOSUB
 4520                                          <1AB0>
4220 IF INKEY(8)=0 OR JOY(0)=4 THE
N le=le-lk*dt                                  <2DE2>
4230 IF INKEY(1)=0 OR JOY(0)=8 THE
N le=le+lk*dt                                  <2DB9>
4240 le=le*(1-lrk*dt)                          <2196>
4250 IF ABS(le*v*v)>hrk THEN le=SG
N(le)*hrk/v/v:SOUND 1,40,dt*100                <50D3>
4260 l=l+(le-kw*(q1-3))*dt*v/100               <34A9>
4270 a=a-lr*v*v                                <1E4A>
4280 v=v+a*dt                                  <195C>
4290 IF v<0 THEN v=0                           <13B5>
4300 IF v>245 THEN v=245                       <155F>
4310 s=s+v*dt/3600                             <1D4D>
4320 w=v*2^(1-g)*ue:IF w>5000 THEN
 w=5000                                        <3263>
4330 h=h*(1-v*kc)+hc*w*(x+1)                   <3114>
4340 IF b<0 THEN GOTO 6060                     <113B>
4350 IF h>100 THEN GOTO 6060                   <123F>
4360 ORIGIN 320,30:MOVE 0,0,4,1:DR
AW 68*SIN(252+v1*0.84),44*COS(252+
v1*0.84)                                       <4467>
4370 MOVE 0,0,4,1:DRAW 68*SIN(252+
v*0.84),44*COS(252+v*0.84)                     <3A84>
4380 ORIGIN 320,-300:MOVE 60*le1,S
QR(96100-3600*le1*le1),4,1:DRAW 60
*le1+4,SQR(96100-3600*le1*le1),4,1
:MOVE 60*le,SQR(96100-3600*le*le),
4,1:DRAW 60*le+4,SQR(96100-3600*le
*le),4,1:le1=le                                <C9B3>
4390 ORIGIN 468,20:sb=SIN(320+b1*2
):cb=COS(320+b1*2):MOVE 44*sb,44*c
b,11,1:DRAW 58*sb,58*cb                        <6333>
4400 sb=SIN(320+b*2):cb=COS(320+b*
2):MOVE 44*sb,44*cb,11,1:DRAW 58*s
b,58*cb                                        <5997>
4410 IF h>50 THEN ORIGIN 580,20:sh
=SIN(240+h1*1.6):ch=COS(240+h1*1.6
):MOVE 44*sh,44*ch,11,1:DRAW 58*sh
,58*ch:sh=SIN(240+h*1.6):ch=COS(24
0+h*1.6):MOVE 44*sh,44*ch,11;1:DRA
W 58*sh,58*ch                                  <D11C>
4420 ORIGIN 0,0:TAG:MOVE 16+w1*0.0
36,40,6,1:PRINT CHR$(129);:MOVE 16
+w*0.036,40:PRINT CHR$(129);:TAGOF
F                                              <4A1B>
4430 t1=t-t0:tm=INT(t1/60):ts=INT(
t1)MOD 60:PAPER#1,10:PEN#1,8,0:LOC
ATE#1,3,2:PRINT#1,CHR$(INT(tm/10)+
170)CHR$((tm MOD 10)+170)"."CHR$(I
NT(ts/10)+170)CHR$((INT(ts)MOD 10)
+170)                                          <A63F>
```

```
4440 TAG:FOR i=3 TO 1 STEP-1:MOVE(
i*-1+4)*16+442,22,1,1:PRINT CHR$(1
90+INT(s1/10^(i-1))MOD 10);:MOVE(i
*-1+4)*16+442,22:PRINT CHR$(190+IN
T(s/10^(i-1))MOD 10);:NEXT i          <8EC1>
4450 FOR i=2 TO 1 STEP-1:MOVE(i*-1
+3)*16+506,22,1,1:PRINT CHR$(190+I
NT(s1/10^(i-3))MOD 10);:MOVE(i*-1+
3)*16+506,22:PRINT CHR$(190+INT(s/
10^(i-3))MOD 10);:NEXT i              <8C9F>
4460 lp=(l-kw)*dt*v+lp                <28D2>
4470 llp=lp/20:llp1=lp1/20:IF llp>
138 OR llp<-138 THEN GOTO 6060        <41AA>
4480 ORIGIN 0,112:MOVE 320-96+llp1
,0,14,1:PRINT CHR$(204)CHR$(205);:
MOVE 320+32+llp1,0:PRINT CHR$(206)
CHR$(207);:MOVE 320-32+llp1,0,2,1:
PRINT CHR$(202)CHR$(203);             <762F>
4490 MOVE 320-32+llp,0,2,1:PRINT C
HR$(202)CHR$(203);:MOVE 320-96+llp
,0,14,1:PRINT CHR$(204)CHR$(205);:
MOVE 320+32+llp,0:PRINT CHR$(206)C
HR$(207);:TAGOFF                      <6F00>
4500 ws=ws+v*dt/3600:IF ws>0.4 THE
N ws=0:GOSUB 4630                     <3A6A>
4510 GOTO 4120                        <0943>
4520 ORIGIN 0,0:TAG:MOVE 44+g1*36,
24,8,1:PRINT CHR$(143);:MOVE 44+g*
36,24:PRINT CHR$(143);:TAGOFF:RETU
RN                                    <44F9>
4530 ' *****                          <0D93>
4540 ' RANDWECHSEL                    <130D>
4550 ' *****                          <0DBB>
4560 d=i(1):i(1)=i(2):i(2)=i(3):i(
3)=i(4):i(4)=d                        <4E26>
4570 INK 1,i(1):INK 2,i(2):INK 3,i
(3):INK 4,i(4)                        <30C1>
4580 dts=s0/(v+0.0125)+1.5:AFTER d
ts GOSUB 4560                         <322D>
4590 RETURN                           <0614>
4600 ' *****                          <0D20>
4610 ' BILDWECHSEL                    <133E>
4620 ' *****                          <0D49>
4630 q2=q1                            <10AC>
4640 READ q1                          <0BC1>
4650 IF q1=q2 THEN 4740               <156C>
4660 IF q1=10 THEN q1=q2:GOTO 6210
                                      <1F9E>
4670 GOSUB 5950                       <09AE>
4680 DI                               <0611>
4690 FOR j=1 TO 4:INK j,13:NEXT j     <1D81>
4700 |SCREENSWAP,1,q3(q1)             <20E5>
4710 EI                               <0650>
4720 GOSUB 5950                       <0913>
4730 INK 1,i(1):INK 2,i(2):INK 3,i
(3):INK 4,i(4)                        <3003>
4740 q3(q2)=q3(q1)                    <1E1C>
4750 RETURN                           <0654>
4760 DATA 4,5,4,3,2,1,2,3,2,3,4,5,
4,3,2,3,4,3,4,5,4,3,10                <3515>
4770 RETURN                           <067C>
4780 ' *****                          <0D88>
4790 ' KREISBOGEN                     <1208>
4800 ' *****                          <0DB0>
4810 FOR i=1 TO 2                     <0EE2>
4820 FOR j=1 TO 6:READ a(j):NEXT j
                                      <20FF>
4830 ORIGIN a(1),a(2)                 <1531>
4840 MOVE a(3)*SIN(a(4)),a(3)*COS(
a(4)),a(6)                            <3511>
4850 FOR j=a(4)TO a(5)STEP 5          <1C23>
4860 DRAW a(3)*SIN(j),a(3)*COS(j)     <2736>
4870 NEXT j,i                         <0F1E>
4880 ORIGIN 0,0                       <09BC>
4890 RETURN                           <066E>
4900 ' *****                          <0D7A>
4910 ' ANZEIGE AUFBAU                 <1674>
4920 ' *****                          <0DA2>
4930 MOVE 0,2,12:DRAW 0,94:DRAW 63
6,94:DRAW 636,2:DRAW 0,2              <26E7>
4940 ORIGIN 320,30                    <0CA3>
4950 FOR i=1 TO 2                     <0EFC>
4960 MOVE 0,64+i,9                    <118D>
4970 FOR j=0 TO 360 STEP 5            <12B7>
4980 DRAW(88+i)*SIN(j),(64+i)*COS(
j)                                    <2BFC>
4990 NEXT j,i                         <0F0E>
5000 MOVE 0,0:FILL 11                 <0D89>
5010 MOVE 0,4,9                       <0B55>
5020 FOR j=0 TO 360 STEP 5            <121C>
5030 DRAW 5*SIN(j),4*COS(j)           <1B7A>
5040 NEXT j                           <0AE7>
5050 MOVE 0,0:FILL 9                  <0C1C>
5060 FOR i=64 TO 60 STEP-2            <13EB>
5070 MOVE 0,i,9                       <0E7B>
5080 FOR j=0 TO 360 STEP 36           <1323>
5090 PLOT(i+24)*SIN(j),i*COS(j)       <2651>
5100 NEXT j,i                         <0FE9>
5110 FOR i=64 TO 52 STEP-2            <134F>
5120 MOVE(i+24),0,7                   <130D>
5130 FOR j=90 TO 450 STEP 36          <14FB>
5140 PLOT(i+24)*SIN(j),i*COS(j)       <26B6>
5150 NEXT j,i                         <0F4F>
5160 PEN#1,9,1:LOCATE#1,10,5:PRINT
#1,CHR$(167)CHR$(168)CHR$(169)        <2C77>
5170 TAG                              <06BB>
5180 MOVE-71,4,2,1:PRINT CHR$(160)
;                                     <18BE>
5190 MOVE-59,28,2,1:PRINT CHR$(161
);                                    <19C2>
5200 MOVE-32,43,2,1:PRINT CHR$(162
);                                    <196D>
5210 MOVE 8,43,2,1:PRINT CHR$(163)
;                                     <1763>
5220 MOVE 26,28,2,1:PRINT CHR$(164
);                                    <185A>
```

```
5230 MOVE 43,4,2,1:PRINT CHR$(165)
;                                   (174A)
5240 ORIGIN 320,-300                (0E8D)
5250 MOVE-SQR(315^2-300^2),300,9,0
                                    (1E6A)
5260 FOR j=-SQR(315^2-300^2)TO SQR
(315^2-300^2)STEP 10:DRAW j,SQR(31
5^2-j^2):NEXT j                     (4B3B)
5270 MOVE-SQR(305^2-300^2),300,10:
FOR j=-SQR(305^2-300^2)TO SQR(305^
2-300^2)STEP 5:DRAW j,SQR(305^2-j^
2):NEXT j                           (63E5)
5280 MOVE 0,310:FILL 9:MOVE-90,301
:FILL 9:MOVE 90,301:FILL 9          (25F5)
5290 MOVE-SQR(314^2-300^2),300;7:F
OR j=-SQR(314^2-300^2)TO SQR(314^2
-300^2)STEP 5:DRAW j,SQR(314^2-j^2
):NEXT j                            (62D5)
5300 TAGOFF:ORIGIN 0,0              (0B16)
5310 MOVE 12,90,10,0:DRAW 232,90:D
RAW 224,54:DRAW 4,54:DRAW 12,90     (2B13)
5320 MOVE 16,88,8,0:DRAW 228,88:DR
AW 220,56:DRAW 8,56:DRAW 16,88      (2A98)
5330 MOVE 200,70:FILL 10            (0F4A)
5340 PEN#1,7,1:LOCATE#1,1,2:PRINT#
1,CHR$(180)CHR$(181)                (2555)
5350 FOR i=468 TO 580 STEP 112      (154B)
5360 ORIGIN i,20                    (0D6B)
5370 MOVE 70*SIN(315),70*COS(315),
5,0                                 (1F24)
5380 FOR j=315 TO 405 STEP 5        (14D8)
5390 DRAW 70*SIN(j),70*COS(j)       (1D95)
5400 NEXT j                         (0ABA)
5410 MOVE 40*SIN(315),40*COS(315)   (1B1C)
5420 FOR j=315 TO 405 STEP 5        (1427)
5430 DRAW 40*SIN(j),40*COS(j)       (1DA5)
5440 NEXT j                         (0A0A)
5450 MOVE 70*SIN(315),70*COS(315):
DRAW 40*SIN(315),40*COS(315)        (329A)
5460 MOVE 70*SIN(45),70*COS(45):DR
AW 40*SIN(45),40*COS(45)            (2ED8)
5470 ORIGIN 0,0:MOVE i,65:FILL 12   (160A)
5480 NEXT i                         (0A55)
5490 ORIGIN 468,20                  (0CF0)
5500 FOR i=345 TO 390 STEP 15       (1532)
5510 MOVE 66*SIN(i),66*COS(i),10:D
RAW 62*SIN(i),62*COS(i)             (3961)
5520 NEXT i                         (0AA5)
5530 MOVE 70*SIN(330),70*COS(330),
5:DRAW 40*SIN(330),40*COS(330)      (3437)
5540 MOVE 50*SIN(320),50*COS(320):
FILL 5                              (1E9F)
5550 ORIGIN 580,20                  (0C63)
5560 FOR i=333 TO 369 STEP 18       (15C3)
5570 MOVE 66*SIN(i),66*COS(i),10:D
RAW 62*SIN(i),62*COS(i)             (39D9)
5580 NEXT i                         (0A1E)
5590 MOVE 70*SIN(387),70*COS(387),
5:DRAW 40*SIN(387),40*COS(387)      (34AF)
5600 MOVE 50*SIN(400),50*COS(400):
FILL 5                              (1E76)
5610 ORIGIN 0,0:TAG                 (0B56)
5620 MOVE 442,44,11,0:PRINT CHR$(1
43)CHR$(143);:MOVE 444,46,2,1:PRIN
T CHR$(182)CHR$(183);               (3B16)
5630 MOVE 554,44,11,0:PRINT CHR$(1
43)CHR$(143);:MOVE 556,46,2,1:PRIN
T CHR$(184)CHR$(185);               (3B2C)
5640 TAGOFF                         (066D)
5650 MOVE 8,8,11,0:DRAW 220,8       (149F)
5660 MOVE 8,26,11:DRAW 220,26       (1474)
5670 MOVE 8,8,11:DRAW 8,26          (127A)
5680 MOVE 220,8:DRAW 220,26         (1196)
5690 MOVE 100,16:FILL 6             (0E7E)
5700 MOVE 184,8,11:DRAW 184,26      (1474)
5710 MOVE 148,8,11:DRAW 148,26      (1414)
5720 MOVE 112,8,11:DRAW 112,26      (14B3)
5730 MOVE 76,8,11:DRAW 76,26        (1438)
5740 TAG                            (0630)
5750 MOVE 12,24,13,1:PRINT CHR$(18
6)CHR$(187);                        (1F75)
5760 MOVE 80,24,11:PRINT"1";:MOVE
116,24:PRINT"2";:MOVE 152,24:PRINT
"3";:MOVE 188,24:PRINT"4";          (3B90)
5770 TAGOFF                         (0670)
5780 MOVE 438,24,13,0:DRAW 616,24   (195E)
5790 MOVE 438,8,13,0:DRAW 616,8     (17FD)
5800 MOVE 438,24,13,0:DRAW 438,8    (1857)
5810 MOVE 616,24,13,0:DRAW 616,8    (18D7)
5820 MOVE 500,20:FILL 8             (0F82)
5830 TAG                            (06E4)
5840 MOVE 568,22,1,1:PRINT CHR$(18
9);                                 (1984)
5850 TAGOFF                         (0610)
5860 MOVE 12,42,5,0:DRAW 216,42:DR
AW 216,32:DRAW 12,32:DRAW 12,42     (2B88)
5870 MOVE 82,42,5,0:DRAW 82,52:DRA
W 146,52:DRAW 146,42:MOVE 114,48:F
ILL 5                               (2ECA)
5880 TAG:MOVE 86,50,9,1:PRINT CHR$
(200)CHR$(201);                     (2072)
5890 PEN,0:TAGOFF:MOVE 0,0,,0       (1293)
5900 ORIGIN 0,0,0,640,400,96        (1639)
5910 RETURN                         (066A)
5920 ' *****                        (0D76)
5930 ' ZEIGER                       (0E8C)
5940 ' *****                        (0D9E)
5950 ORIGIN 320,30:MOVE 0,0,4,1:DR
AW 68*SIN(252+v*0.84),44*COS(252+v
*0.84)                              (4285)
5960 ORIGIN 468,20:sb=SIN(320+b*2)
:cb=COS(320+b*2):MOVE 44*sb,44*cb,
11,1:DRAW 58*sb,58*cb               (6114)
5970 IF h>50 THEN ORIGIN 580,20:sh
=SIN(240+h*1.6):ch=COS(240+h*1.6):
MOVE 44*sh,44*ch,11,1:DRAW 58*sh,5
```

```
8*ch ELSE ORIGIN 580,20:MOVE 44*SI
N(320),44*COS(320),11,1:DRAW 58*SI
N(320),58*COS(320)                    <AE38>
5980 ORIGIN 0,0:TAG:MOVE 16+w*0.03
6,40,6,1:PRINT CHR$(129);:TAGOFF      <2D32>
5990 PAPER#1,10:PEN#1,8,0:LOCATE#1
,3,2:PRINT#1,"      "                 <27AB>
6000 TAG:FOR j=3 TO 1 STEP-1:MOVE(
j*-1+4)*16+442,22,1,1:PRINT CHR$(1
90+INT(s/10^(j-1))MOD 10);:NEXT j:
MOVE 506,22:PRINT CHR$(188);          <6683>
6010 FOR j=2 TO 1 STEP-1:MOVE(j*-1
+3)*16+506,22,1,1:PRINT CHR$(190+I
NT(s/10^(j-3))MOD 10);:NEXT j         <5362>
6020 llp=lp/20:ORIGIN 0,112:MOVE 3
20-32+llp,0,2,1:PRINT CHR$(202)CHR
$(203);:MOVE 320-96+llp,0,14,1:PRI
NT CHR$(204)CHR$(205);:MOVE 320+32
+llp,0:PRINT CHR$(206)CHR$(207);:T
AGOFF                                 <8519>
6030 ORIGIN 320,-300:MOVE 60*le1,S
QR(96100-3600*le1*le1),4,1:DRAW 60
*le1+4,SQR(96100-3600*le1*le1),4,1
                                      <688B>
6040 ORIGIN 0,0:TAG:MOVE 44+g*36,2
4,8,1:PRINT CHR$(143);:TAGOFF         <297E>
6050 RETURN                           <0681>
6060 ORIGIN 0,0:WINDOW#2,6,15,10,1
2:PAPER#2,9:CLS#2                     <2211>
6070 MOVE 160,208,11,0:DRAW 480,20
8:DRAW 480,256:DRAW 160,256:DRAW 1
60,208                                <3086>
6080 TAG:MOVE 180,238,2,1:PRINT"GA
ME OVER";:TAGOFF                      <2156>
6090 IF INKEY(47)<>0 THEN 6090        <121E>
6100 AFTER 1 GOSUB 6200               <0BBA>
6110 GOTO 6720                        <092F>
6120 FOR j=0 TO 15:INK j,0:NEXT j     <1D0E>
6130 GOSUB 5920                       <092E>
6140 ORIGIN 0,0,0,640,400,96:MOVE
0,0,0,0                               <1F0B>
6150 ON q1 GOSUB 1380,760,3400,208
0,2710                                <1F19>
6160 IF q1<>3 THEN/SCREENSWAP,1,q3
(3)                                   <25E2>
6170 q4=q3(3)                         <13BF>
6180 q3(3)=1:q3(q1)=q4                <22D9>
6190 GOTO 3910                        <09D0>
6200 RETURN                           <06AF>
6210 ORIGIN 0,0:WINDOW#2,8,13,10,1
2:PAPER#2,9:CLS#2                     <22A6>
6220 MOVE 224,208,11,0:DRAW 416,20
8:DRAW 416,256:DRAW 224,256:DRAW 2
24,208                                <30E2>
6230 TAG:MOVE 246,238,2,1:PRINT"RE
ADY";:TAGOFF                          <1D4D>
6240 IF INKEY(47)<>0 THEN 6240        <12F8>
6250 ' *****                          <0D0B>
6260 ' EINTRAGUNGSLISTE               <183F>
6270 ' *****                          <0D32>
6280 AFTER 1 GOSUB 6200               <0B23>
6290 IF einm(35)*60+eins(35)<tm*60
+ts THEN 6720                         <3380>
6300 TAG:ORIGIN 0,0                   <0B7B>
6310 PAPER 9:CLS:RESTORE 7010         <0E3D>
6320 INK 0,1:INK 1,25:INK 2,24:INK
 3,15:INK 4,6:INK 5,7:INK 6,2:INK
7,13:INK 8,8:INK 9,0:INK 10,4:INK
11,15:INK 12,24:INK 13,26:INK 14,1
3:INK 15,26                           <639A>
6330 FOR j=1 TO 32                    <0F4F>
6340 READ x2(j),y2(j),ii2(j)          <2A32>
6350 NEXT j                           <0A29>
6360 FOR j=1 TO 26                    <0F73>
6370 MOVE x2(j),y2(j),14,1:PRINT C
HR$(143);:MOVE x2(j),y2(j),ii2(j),
1:PRINT CHR$(j+64);                   <612C>
6380 NEXT j                           <0A65>
6390 FOR j=1 TO 7                     <0E95>
6400 READ x3,y3,n3:MOVE x3,y3,4,1:
PRINT CHR$(n3);                       <349E>
6410 NEXT j                           <0AA2>
6420 FOR j=27 TO 32                   <10A6>
6430 MOVE x2(j),y2(j),14,1:PRINT C
HR$(143);:READ n2:MOVE x2(j),y2(j)
,ii2(j),1:PRINT CHR$(n2);             <6629>
6440 NEXT j                           <0ADE>
6450 TAGOFF                           <06C4>
6460 MOVE x2(1)-4,y2(1)+2,6,0:DRAW
 x2(1)+32,y2(1)+2:DRAW x2(1)+32,y2
(1)-16:DRAW x2(1)-4,y2(1)-16:DRAW
x2(1)-4,y2(1)+2                       <7F24>
6470 MOVE 376,232,6,0:DRAW 616,232
:DRAW 616,200:DRAW 376,200:DRAW 37
6,232:MOVE 400,220:FILL 0             <3B68>
6480 m2=1:m21=1:pp2=12:FOR k=1 TO
7:p$(k)="":NEXT k                     <3DFE>
6490 m21=m2                           <111D>
6500 IF INKEY(1)=0 OR JOY(0)=8 THE
N m2=m2+1:IF m2=33 THEN m2=1:GOTO
6540 ELSE GOTO 6540                   <4029>
6510 IF INKEY(8)=0 OR JOY(0)=4 THE
N m2=m2-1:IF m2=0 THEN m2=32:GOTO
6540 ELSE GOTO 6540                   <4099>
6520 IF INKEY(47)=0 OR JOY(0)=16 T
HEN PEN 6:PAPER 0:GOTO 6580           <2269>
6530 GOTO 6490                        <0941>
6540 MOVE x2(m21)-4,y2(m21)+2,9,0:
DRAW x2(m21)+32,y2(m21)+2:DRAW x2(
m21)+32,y2(m21)-16:DRAW x2(m21)-4,
y2(m21)-16:DRAW x2(m21)-4,y2(m21)+
2                                     <B12F>
6550 MOVE x2(m2)-4,y2(m2)+2,6,0:DR
AW x2(m2)+32,y2(m2)+2:DRAW x2(m2)+
32,y2(m2)-16:DRAW x2(m2)-4,y2(m2)-
16:DRAW x2(m2)-4,y2(m2)+2             <A721>
```

```
6560 FOR j=1 TO 100:NEXT j                    <1542>
6570 GOTO 6490                                <0991>
6580 IF m2<27 THEN IF pp2<19 THEN
pp2=pp2+1:LOCATE pp2,12:PRINT CHR$
(m2+64):p$(pp2-12)=CHR$(m2+64):FOR
 j=1 TO 150:NEXT j                            <7052>
6590 IF m2=27 THEN IF pp2<19 THEN
pp2=pp2+1:LOCATE pp2,12:PRINT CHR$
(228):p$(pp2-12)=CHR$(228):FOR j=1
 TO 150:NEXT j                                <6409>
6600 IF m2=28 THEN IF pp2<19 THEN
pp2=pp2+1:LOCATE pp2,12:PRINT CHR$
(42):p$(pp2-12)=CHR$(42):FOR j=1 T
O 150:NEXT j                                  <6450>
6610 IF m2=29 THEN IF pp2<19 THEN
pp2=pp2+1:LOCATE pp2,12:PRINT CHR$
(46):p$(pp2-12)=CHR$(46):FOR j=1 T
O 150:NEXT j                                  <64C4>
6620 IF m2=30 THEN IF pp2<19 THEN
pp2=pp2+1:p$(pp2-12)=CHR$(32):FOR
j=1 TO 150:NEXT j                             <5155>
6630 IF m2=31 THEN IF pp2>12 THEN
LOCATE pp2,12:PRINT CHR$(32):p$(pp
2-12)=CHR$(32):pp2=pp2-1:FOR j=1 T
O 150:NEXT                                    <60E8>
6640 IF m2=32 THEN GOTO 6660                  <137F>
6650 GOTO 6490                                <0932>
6660 PAPER 9:CLS                              <091D>
6670 FOR j=35 TO 0 STEP-1                     <12AE>
6680 IF einm(j)*60+eins(j)>tm*60+t
s THEN 6690 ELSE 6700                         <3C0C>
6690 NEXT j                                   <0AD3>
6700 FOR k=34 TO j+1 STEP-1:einm(k
+1)=einm(k):eins(k+1)=eins(k):ein$
(k+1)=ein$(k):NEXT k                          <7535>
6710 einm(j+1)=tm:eins(j+1)=ts:ein
$(j+1)="":FOR k=1 TO 7:ein$(j+1)=e
in$(j+1)+p$(k):NEXT k                         <7B7D>
6720 ' *****                                  <0DB9>
6730 ' TOP 35                                 <0EC4>
6740 ' *****                                  <0DE1>
6750 PAPER 9:CLS                              <09D1>
6760 INK 0,16:INK 1,19:INK 2,18:IN
K 3,9:INK 4,6:INK 5,7:INK 6,2:INK
7,13:INK 8,8:INK 9,0:INK 10,4:INK
11,15:INK 12,24:INK 13,26:INK 14,1
3:INK 15,26                                   <635F>
6770 RESTORE 7410:PEN 0,0:FOR k=3
TO 5:FOR kk=5 TO 14:READ chr:LOCAT
E kk,k:PRINT CHR$(chr):NEXT kk,k              <5085>
6780 z1=0:z2=0                                <1453>
6790 TAG:ORIGIN 0,0                           <0B51>
6800 FOR k=1 TO 7:MOVE k*24-16,((k
*-1)+8)*25+120,11,1:PRINT USING"##
";k;                                          <40FC>
6810 MOVE k*24+44,((k*-1)+8)*25+12
0,10:PRINT CHR$(41);                          <3035>
6820 MOVE k*24+88,((k*-1)+8)*25+12
0,13:PRINT ein$(k);                           <36F2>
6830 MOVE k*24+322,((k*-1)+8)*25+1
20,10:PRINT USING"##";einm(k);                <3E65>
6840 MOVE k*24+374,((k*-1)+8)*25+1
20,8:PRINT CHR$(46);                          <30C4>
6850 MOVE k*24+398,((k*-1)+8)*25+1
20,11:PRINT USING"##";eins(k);:IF
eins(k)<10 THEN MOVE k*24+398,((k*
-1)+8)*25+120,11:PRINT CHR$(48);              <7D5F>
6860 NEXT k:TAGOFF                            <0CE2>
6870 IF INKEY(0)=0 OR JOY(0)=1 THE
N IF z2>0 THEN z1=z2:z2=z2-7:GOTO
6920                                          <3D42>
6880 IF INKEY(2)=0 OR JOY(0)=2 THE
N IF z2<28 THEN z1=z2:z2=z2+7:GOTO
 6920                                         <3EA5>
6890 IF INKEY(47)=0 OR JOY(0)=16 T
HEN GOTO 6990                                 <1C9B>
6900 IF INKEY(58)=0 THEN GOTO 7440
                                              <13F5>
6910 GOTO 6870                                <0923>
6920 TAG:FOR k=1 TO 7:MOVE k*24-16
,((k*-1)+8)*25+120,11,1:PRINT USIN
G"##";k+z1;:MOVE k*24-16,((k*-1)+8
)*25+120,11,1:PRINT USING"##";k+z2
;                                             <80F3>
6930 MOVE k*24+88,((k*-1)+8)*25+12
0,13:PRINT ein$(k+z1);:MOVE k*24+8
8,((k*-1)+8)*25+120,13:PRINT ein$(
k+z2);                                        <74D8>
6940 MOVE k*24+322,((k*-1)+8)*25+1
20,10:PRINT USING"##";einm(k+z1);:
MOVE k*24+322,((k*-1)+8)*25+120,10
:PRINT USING"##";einm(k+z2);                  <8423>
6950 MOVE k*24+398,((k*-1)+8)*25+1
20,11:PRINT USING"##";eins(k+z1);:
IF eins(k+z1)<10 THEN MOVE k*24+39
8,((k*-1)+8)*25+120,11:PRINT CHR$(
48);                                          <8939>
6960 MOVE k*24+398,((k*-1)+8)*25+1
20,11:PRINT USING"##";eins(k+z2);:
IF eins(k+z2)<10 THEN MOVE k*24+39
8,((k*-1)+8)*25+120,11:PRINT CHR$(
48);                                          <891A>
6970 NEXT k                                   <0A08>
6980 GOTO 6870                                <09B0>
6990 PAPER 0                                  <07C5>
7000 GOTO 6120                                <095B>
7010 DATA 32,168,6                            <0FA1>
7020 DATA 56,192,5                            <0F48>
7030 DATA 76,216,4                            <0FED>
7040 DATA 84,240,3                            <0FFD>
7050 DATA 92,264,2                            <0F8C>
7060 DATA 104,288,3                           <1077>
7070 DATA 112,312,4                           <1094>
7080 DATA 120,336,5                           <102B>
7090 DATA 128,360,6                           <1087>
7100 DATA 144,384,5                           <1017>
```

```
7110 DATA 172,364,4                    (10F3)
7120 DATA 180,340,3                    (107F)
7130 DATA 184,316,2                    (1093)
7140 DATA 184,292,3                    (1028)
7150 DATA 200,268,4                    (10EF)
7160 DATA 228,288,5                    (109C)
7170 DATA 240,312,6                    (10AC)
7180 DATA 252,336,5                    (1045)
7190 DATA 264,360,4                    (1051)
7200 DATA 292,380,3                    (108E)
7210 DATA 312,360,2                    (1069)
7220 DATA 316,336,3                    (108D)
7230 DATA 316,312,4                    (1021)
7240 DATA 312,288,5                    (1030)
7250 DATA 308,264,6                    (10D5)
7260 DATA 300,240,5                    (1008)
7270 DATA 292,216,4                    (106C)
7280 DATA 292,192,3                    (10EC)
7290 DATA 296,168,2                    (10F8)
7300 DATA 316,148,3                    (10E9)
7310 DATA 352,156,4                    (1000)
7320 DATA 372,176,5                    (1045)
7330 DATA 368,316,79                   (11BD)
7340 DATA 400,320,84                   (116D)
7350 DATA 432,328,79                   (1198)
7360 DATA 464,336,82                   (1168)
7370 DATA 496,340,87                   (115C)
7380 DATA 528,336,65                   (1188)
7390 DATA 560,332,89                   (113C)
7400 DATA 228,42,46,32,242,208         (1BB1)
7410 DATA 158,150,156,150,156,32,1
46,156,150,152                         (2DA5)
7420 DATA 149,149,149,151,153,32,3
2,157,147,156                          (2C71)
7430 DATA 145,147,153,145,32,32,14
6,153,146,153                          (2CA7)
7440 |ERA,"motorway.dat"               (1908)
7450 OPENOUT"motorway.dat":FOR k=1
 TO 35:WRITE#9,einm(k),eins(k),ein
$(k):NEXT k:CLOSEOUT                   (545E)
7460 GOSUB 7480:END                    (0BEF)
7470 GOSUB 7480:PRINT"Fehler ";ERR
;" in Zeile ";ERL:END                  (28B3)
7480 MODE 2:PEN 1:PAPER 0:INK 1,24
:INK 0,1:BORDER 1:CLEAR INPUT:RETU
RN                                     (2003)
```

```
10 '*****************************    (2397)
11 '*MOTORDATERZEUGUNGSPROGRAMM*     (235C)
12 '*            VON              *   (2351)
13 '*       HOLGER HAUBER         *   (2365)
14 '*           FUER              *   (236E)
15 '*     SCHNEIDER CPC-WELT      *   (23E3)
20 '*         CPC 6128         JE*   (2399)
30 '*****************************    (23BF)
40 '                                   (075B)
50 OPENOUT"motordat"                   (1015)
60 FOR i=1 TO 14                       (0F73)
70 READ a,b,c,d,e,f,g,h,i              (32EF)
80 WRITE#9,a,b,c,d,e,f,g,h,i           (35A0)
90 NEXT i                              (0A24)
100 CLOSEOUT                           (06EE)
110 DATA 205,230,68,68,92,92,68,68
,230                                   (24F4)
120 DATA 206,24,122,66,66,66,66,94
,24                                    (23A1)
130 DATA 207,112,32,32,32,32,34,46
,110                                   (24D7)
140 DATA 208,124,0,16,16,16,16,124
,124                                   (2496)
150 DATA 209,46,110,64,60,6,2,118,
116                                    (23A2)
160 DATA 210,118,54,32,32,44,32,32
,112                                   (24A8)
170 DATA 211,124,124,0,16,16,16,16
,56                                    (2390)
180 DATA 212,60,54,112,96,60,2,66,
60                                     (224D)
190 DATA 213,0,0,60,50,112,96,98,6
0                                      (210C)
200 DATA 214,0,0,60,50,114,98,98,6
0                                      (2124)
210 DATA 215,24,24,62,24,48,48,54,
28                                     (2220)
220 DATA 216,24,0,24,24,56,48,54,2
8                                      (213E)
230 DATA 217,0,0,62,48,48,28,2,124
                                       (20A3)
240 DATA 218,14,27,24,24,56,48,124
,48                                    (231C)
```

```
10 '*****************************    (2397)
11 '*    ERZEUGUNGSPROGRAMM     *    (23C7)
12 '*          FUER             *    (236A)
13 '*      MOTORWAY.DAT         *    (2305)
14 '*          VON              *    (2391)
15 '*      HOLGER HAUBE         *    (23B0)
16 '*    SCHNEIDER CPC-WELT     *    (23E5)
20 '*        CPC 6128        JE*    (23AE)
30 '*****************************    (23BF)
40 '                                   (075B)
50 OPENOUT"motorway.dat"               (14C9)
60 READ einm,eins,ein$                 (1CC2)
70 FOR i=1 TO 35                       (0FDB)
80 WRITE#9,einm,eins,ein$              (1FFC)
90 NEXT i                              (0A24)
100 CLOSEOUT                           (06EE)
110 END                                (063F)
120 DATA 19,59,"......."               (168F)
```

```
10 '*****************************  <2397>
11 '*MOTORBILERZEUGUNGSPROGRAMM*  <235C>
12 '*          VON              *  <238D>
13 '*      HOLGER HAUBE         *  <2311>
14 '*          FUER             *  <236E>
15 '*     SCHNEIDER AKTIV       *  <239A>
20 '*        CPC 6128        JE*  <2399>
30 '*****************************  <23BF>
40 '                               <075B>
50 MODE 0:INK 0,1:INK 12,0:BORDER
0:PAPER 0:PEN 1                    <1BE6>
60 SYMBOL AFTER 32                 <09B3>
70 SYMBOL 223,0,0,102,0,0,102,0,0  <1A02>
80 SYMBOL 224,0,0,24,0,0,102,0,0   <1A23>
90 SYMBOL 225,0,0,96,0,0,96,0,0    <1A80>
100 SYMBOL 226,0,0,60,0,0,24,0,0   <1A9E>
110 SYMBOL 227,0,0,62,0,0,96,0,0   <1A07>
120 SYMBOL 228,0,0,24,0,0,102,0,0  <1AB3>
130 SYMBOL 229,0,0,126,0,0,96,0,0  <1AA4>
140 SYMBOL 230,0,0,126,0,0,24,0,0  <1AB6>
150 SYMBOL 231,102,0,0,102,0,0,0,0
                                   <1A46>
160 SYMBOL 232,102,0,0,24,0,0,0,0  <1AC8>
170 SYMBOL 233,96,0,0,126,0,0,0,0  <1A01>
180 SYMBOL 234,24,0,0,60,0,0,0,0   <1A89>
190 SYMBOL 235,6,0,0,124,0,0,0,0   <19B0>
200 SYMBOL 236,102,0,0,24,0,0,0,0  <1AE3>
210 SYMBOL 237,96,0,0,96,0,0,0,0   <1A1B>
220 SYMBOL 238,24,0,0,24,0,0,0,0   <1A8E>
230 SYMBOL 239,0,0,0,102,102,0,126
,126                               <1C04>
240 SYMBOL 240,0,0,0,60,102,0,102,
102                                <1C0A>
250 SYMBOL 241,0,0,0,96,96,0,96,96
                                   <1C89>
260 SYMBOL 242,0,0,0,60,24,0,24,24
                                   <1C3D>
270 SYMBOL 243,0,0,0,126,96,0,124,
62                                 <1C6F>
280 SYMBOL 244,0,0,0,60,102,0,102,
102                                <1C7E>
290 SYMBOL 245,0,0,0,126,96,0,124,
124                                <1C0F>
300 SYMBOL 246,0,0,0,126,24,0,24,2
4                                  <1C4F>
310 SYMBOL 247,0,102,102,0,0,0,0,0
                                   <1A2C>
320 SYMBOL 248,0,102,60,0,0,0,0,0  <1ACD>
330 SYMBOL 249,0,96,126,0,0,0,0,0  <1ACE>
340 SYMBOL 250,0,24,60,0,0,0,0,0   <1AEA>
350 SYMBOL 251,0,6,126,0,0,0,0,0   <1945>
360 SYMBOL 252,0,102,60,0,0,0,0,0  <1A63>
370 SYMBOL 253,0,96,96,0,0,0,0,0   <1A4B>
380 SYMBOL 254,0,24,24,0,0,0,0,0   <1AC8>
390 ORIGIN-100,-500,0,640,340,50   <1BF5>
400 DEG                            <0668>
410 r=1                            <0B8D>
420 FOR i=5 TO 60 STEP 0.8         <163C>
430 MOVE 0,0,r:DRAW 1200*SIN(i-1),
1200*COS(i-0.8)                    <32CE>
440 MOVE 0,0,r:DRAW 1200*SIN(i),12
00*COS(i)                          <2981>
450 MOVE 855*SIN(i)-8,855*COS(i):F
ILL r                              <2753>
460 r=r+1:IF r>11 THEN r=1         <2035>
470 NEXT i                         <0A1E>
480 ORIGIN 0,0                     <094A>
490 FOR i=1 TO 61                  <0F8C>
500 READ x,y,x1,y1                 <1B77>
510 MOVE x,y-20,12,0:DRAW x1,y1-20
                                   <2759>
520 NEXT i                         <0A83>
530 DATA   4,120, 32,304 , 32,304,
 64,304 , 64,304, 72,248 , 72,248,
 84,300                            <4997>
540 DATA  84,300,112,300 ,112,300,
124,116 ,124,116, 92,120 , 92,120,
 92,248                            <4917>
550 DATA  92,248, 84,180 , 84,180,
 60,184 , 60,184, 52,248 , 52,248,
 40,120                            <4908>
560 DATA  40,120,  4,120 ,216,120,
212,296 ,212,296,168,300 ,168,300,
172,328                            <4974>
570 DATA 172,328,284,320 ,284,320,
280,288 ,280,288,244,292 ,244,292,
244,120                            <49B3>
580 DATA 244,120,216,120 ,320,116,
324,292 ,364,212,392,116 ,392,116,
364,120                            <49F0>
590 DATA 364,120,348,188 ,348,188,
348,116 ,348,116,320,116 ,352,236,
352,268                            <499E>
600 DATA 392,116,388,304 ,388,304,
420,304 ,420,304,416,184 ,416,184,
428,236                            <49D1>
610 DATA 428,236,448,236 ,448,236,
456,188 ,456,188,456,304 ,456,304,
484,304                            <4927>
620 DATA 484,304,480,116 ,480,116,
452,120 ,452,120,436,176 ,436,176,
420,116                            <4923>
630 DATA 420,116,392,116 ,480,116,
504,292 ,504,292,536,296 ,536,296,
564,116                            <49A9>
640 DATA 564,116,536,116 ,536,116,
528,172 ,528,172,516,172 ,516,172,
512,116                            <49A5>
650 DATA 512,116,480,116 ,520,204,
520,240 ,520,240,524,204 ,524,204,
520,204                            <49C6>
660 DATA 564,120,580,200 ,580,200,
544,296 ,544,296,568,304 ,568,304,
592,248                            <4931>
670 DATA 592,248,596,300 ,596,300,
628,296 ,628,296,596,116 ,596,116,
```

```
564,120                                  (49E1)
680 DATA 328,292,356,292                 (16F3)
690 FOR i=1 TO 6                         (0EB2)
700 READ x,y,rx,ry,e                     (20B9)
710 ORIGIN x,y-20                        (1228)
720 MOVE 0,ry                            (0D5D)
730 FOR a=0 TO e STEP 5                  (1363)
740 DRAW rx*SIN(a),ry*COS(a)             (23F9)
750 NEXT a,i                             (0F58)
760 DATA 164,212,44,88,360               (18DB)
770 DATA 164,212,16,44,360               (180E)
780 DATA 284,206,40,84,360               (18C0)
790 DATA 284,206,16,40,360               (18E1)
800 DATA 356,252,8,16,180                (1792)
810 DATA 356,252,32,40,160               (18C0)
820 ORIGIN 0,-20                         (0B17)
830 MOVE 1,80:FILL 12:MOVE 164,212
:FILL 12:MOVE 284,206:FILL 12:MOVE
 360,252:FILL 12:MOVE 200,280:FILL
 12:MOVE 248,272:FILL 12:MOVE 380,
200:FILL 12                              (5584)
840 DIM i1(12):i1(1)=26:i1(2)=24:i
1(3)=15:i1(4)=6:i1(5)=4:i1(6)=8:i1
(7)=2:i1(8)=1:i1(9)=13:i1(10)=7:i1
(11)=16                                  (8FDB)
850 ORIGIN 0,0,0,640,400,0               (15BF)
860 FOR j=398 TO 0 STEP-398              (1558)
870 FOR i=1 TO 640 STEP 88               (131F)
880 FOR k=0 TO 10                        (0FAD)
890 PLOT i+k*8,j,k+1,0:PLOT(i+k*8)
+4,j,k+1,0                               (3EEF)
900 NEXT k,i,j                           (1450)
910 FOR j=340 TO 50 STEP-290             (1614)
920 FOR i=1 TO 640 STEP 132              (1336)
930 FOR k=0 TO 10                        (0F12)
940 PLOT i+k*12,j,k*-1+11,0:PLOT(i
+k*12)+4,j,k*-1+11,0:PLOT(i+k*12)+
8,j,k*-1+11,0                            (6C70)
950 NEXT k,i,j                           (14B4)
960 RESTORE 1020:TAG:pln=13              (1505)
970 FOR i=7 TO 1 STEP-1                  (11A5)
980 FOR j=1 TO 64                        (0FEC)
990 READ pl:IF pl=1 THEN PLOT j*4+
192,i*2+346,pln,0                        (3372)
1000 NEXT j:pln=pln+1:IF pln>15 TH
EN pln=13                                (2F0E)
1010 NEXT i                              (0A58)
1020 DATA 0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,
0                                        (867A)
1030 DATA 1,1,0,1,1,1,0,0,0,1,1,0,
1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,
1,0,0,0,0,1,1,1,1,0,0,1,1,0,1,1,1,
0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,0,
0                                        (86B6)
1040 DATA 0,1,1,0,0,1,1,0,0,0,1,1,
0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,
0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,
1,0,0,1,1,1,1,1,0,0,0,1,1,0,0,0,0,
0                                        (86A9)
1050 DATA 0,1,1,0,0,1,1,0,0,0,1,1,
0,0,0,0,0,0,1,1,1,1,1,1,0,0,1,1,0,0,
0,0,0,0,1,1,1,1,1,1,0,0,1,1,0,0,1,
1,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,
0                                        (8620)
1060 DATA 0,1,1,1,1,1,0,0,0,0,1,1,
0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,
1,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0,1,
1,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,0,
0                                        (8610)
1070 DATA 0,1,1,0,0,0,0,0,0,0,1,1,
0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,
1,1,0,0,1,1,0,0,0,0,0,0,1,1,0,0,1,
1,0,0,0,1,1,0,1,1,0,0,0,0,0,0,1,1,
0                                        (8662)
1080 DATA 1,1,1,1,0,0,0,0,0,1,1,1,
1,0,0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,1,
1,0,0,0,0,0,1,1,1,1,0,0,0,1,1,0,0,1,
1,0,0,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0,
0                                        (868E)
1090 ii1(1)=10:ii1(2)=12:ii1(3)=18
                                         (2B4F)
1100 FOR j=50 TO 590 STEP 540            (1502)
1110 ORIGIN j,370:DEG:pln=13             (1A7D)
1120 FOR i=10 TO 370 STEP 20             (148D)
1130 MOVE 0,0,pln:DRAW 35*SIN(i),2
0*COS(i)                                 (2925)
1140 pln=pln+1:IF pln>15 THEN pln=
13                                       (291B)
1150 NEXT i,j                            (0F83)
1160 FOR j=80 TO 600 STEP 520            (1554)
1170 ORIGIN j,25:DEG:pln=13              (1940)
1180 MOVE 0,23                           (0A80)
1190 FOR i=0 TO 360 STEP 20              (1369)
1200 DRAW 21*SIN(i),23*COS(i),pln        (2486)
1210 pln=pln+1:IF pln>15 THEN pln=
13                                       (29A7)
1220 NEXT i,j                            (0F0E)
1230 FOR j=40 TO 560 STEP 520            (15B0)
1240 ORIGIN j,25:DEG:pln=13              (19CC)
1250 MOVE 0,23                           (0A0D)
1260 FOR i=360 TO 0 STEP-20              (14DF)
1270 DRAW 21*SIN(i),23*COS(i),pln        (2413)
1280 pln=pln+1:IF pln>15 THEN pln=
13                                       (2933)
1290 NEXT i,j                            (0F9B)
1300 ORIGIN 80,25:DEG:pln=13             (17B8)
1310 MOVE 0,15                           (0A66)
1320 FOR i=360 TO 0 STEP-40              (14A7)
1330 DRAW 15*SIN(i),15*COS(i),pln        (2451)
1340 pln=pln+1:IF pln>15 THEN pln=
13                                       (29AB)
1350 NEXT i                              (0A02)
1360 ORIGIN 560,25:DEG:pln=13            (1838)
```

```
1370 MOVE 0,15                              (0ADE)
1380 FOR i=0 TO 360 STEP 40                 (1336)
1390 DRAW 15*SIN(i),15*COS(i),pln           (24C9)
1400 pln=pln+1:IF pln>15 THEN pln=
13                                          (2924)
1410 NEXT i                                 (0A79)
1420 ORIGIN 0,0:TAG:FOR i=1 TO 8            (152B)
1430 MOVE i*32+160,390,13,1:PRINT
CHR$(i+222);:NEXT i                         (2D37)
1440 FOR i=1 TO 8                           (0E99)
1450 MOVE i*32+160,374,13,1:PRINT
CHR$(i+230);:NEXT i                         (2DD8)
1460 FOR i=1 TO 8                           (0EC1)
1470 MOVE i*32+160,390,14,1:PRINT
CHR$(i+238);:NEXT i                         (2DCE)
1480 FOR i=1 TO 8                           (0EE9)
1490 MOVE i*32+160,374,14,1:PRINT
CHR$(i+246);:NEXT i                         (2D66)
1500 TAGOFF                                 (0603)
1510 SAVE"motorbil",B,&C000,&4000           (1D1F)
1520 MODE 2:PEN 1:PAPER 0:INK 0,1:
INK 1,24:BORDER 1                           (1B77)
```

Vollgas

gestartet. Nun wird die Datei >MOTORWAY.DAT< mit der Eintragungstabelle nachgeladen.
Danach baut der Computer das Armaturenbrett und die fünf Bilder mit den verschiedenen Straßenstellungen, von denen vier in der zweiten Bank abgelegt werden und eine im Bildschirmbereich der ersten Bank bleibt, auf. Während dieser Wartezeit, die etwa 65 Sekunden beträgt, sind alle 16 PENs schwarz gefüllt. Sobald das Bild erscheint, kann durch das Drücken der Leertaste die Fahrsimulation beginnen. Hierbei gelten die folgenden Joystickbelegungen:
>aufwärts< Beschleunigung des Fahrzeugs
>abwärts< Abbremsen des Fahrzeugs
>links< und >rechts< Steuern
>1<, >2<, >3< und >4< Gangschaltung.
Wenn die Meldung >GAME OVER< erscheint, so sind Sie entweder auf den rotweißen Rand gefahren, es ist kein Benzin mehr vorhanden oder die Motortemperatur ist zu hoch. Um zu vermeiden, daß die Temperatur zu schnell ansteigt, sollten Sie nicht zu lange in niedrigen Gängen fahren, sondern rechtzeitig einen höheren Gang einlegen.
Auch ständiges Abbremsen und wieder Beschleunigen sollte vermieden werden, um immer genügend Benzin im Tank zu behalten. Nur so kann es Ihnen gelingen, die rund zehn Kilometer lange Rennstrecke durchzustehen.
Haben Sie die Strecke in einer relativ guten Zeit zurückgelegt, dürfen Sie Ihren Namen in die Tabelle eintragen. Um die Eintragung abzuschließen, fahren Sie mit dem Cursor auf das Feld >END< und drücken den Joy-Button oder die Leertaste. Damit gelangen Sie zu der Top 35 Eintragungstabelle.
Allerdings werden Grünmonitor-Besitzer Schwierigkeiten mit der Farbgebung in der Eintragungstabelle haben. Wenn Sie keinen Farbmonitor besitzen, ist es am besten, die entsprechenden Ink- und Pen-Befehle anzupassen.
Nach dem Drücken der Leertaste muß eines der fünf Bilder, auf das die Meldung >GAME OVER< oder >READY< geprintet wurde, neu aufgebaut werden, bevor die nächste Fahrsimulation beginnen kann. Fehler, die zum Beispiel an der Ganganzeigetafel durch gleichzeitiges Drücken der Gangtasten an den Bildern entstanden sind, können nur, da die anderen vier Straßenstellungen nicht nochmals frisch aufgebaut wurden, durch Unterbrechen des Programms und einen Neustart mit >RUN 80< behoben werden.
Sollte die vorgegebene Rennstrecke mit der Zeit zu langweilig werden, so kann die Data-Zeile in der Programmzeile 4760, die die Werte für die verschiedenen Straßenstellungen enthält, beliebig verändert werden. Dabei stehen die Werte für:

1 = stark linke Kurve
2 = leicht linke Kurve
3 = gerade Strecke
4 = leicht rechte Kurve
5 = stark rechte Kurve
10 = ** ENDE **

Nun hoffen wir, daß Sie viel Spaß mit dieser Fahrsimulation haben werden. Gelingt es Ihnen, unsere Rekordzeit von 3 min und 42 sec zu brechen? *JE* □

RSX Extended BASIC

Das Programm RSX Extended BASIC ist eine äußerst leistungsstarke Befehlserweiterung für den Schneider CPC 464. Dem User stehen 58(!) neue Befehle zur Verfügung, die als RSX in das Betriebssystem eingebunden wurden.

Eine Befehlserweiterung für den CPC 464 hat unser Autor Andre Schröder aus Hannover geschrieben.
Die 58 neuen Befehle umfassen die Gebiete Grafik, Schriftvergrößerung und Manipulation, Disketten- und Kassettenroutinen, Bildschirm- und Speicheroperationen sowie mehrere Toolkits. Extended BASIC läuft sowohl mit Kassette als auch mit Diskette. Für eine sinnvolle Anwendung empfiehlt sich der gleichzeitige Betrieb beider Speichermedien.
Tippen Sie zunächst den BASIC-Lader ein und speichern Sie ihn vorsichtshalber ab. Tippfehler und der damit verbundene Absturz des Computers werden dadurch nicht zur Katastrophe. Danach starten Sie den BASIC-Lader. Es wird nun automatisch die Binärdatei EXTEN.BIN erzeugt und abgespeichert. Geladen wird die MC-Datei künftig durch das kurze BASIC-Programm EXTENDED.BAS, das die Kassettenbesitzer vor das Binärfile setzen sollten.
Allen Befehlen muß der RSX-Strich vorangestellt werden. Sie finden dieses Zeichen rechts neben P auf der Tastatur und erreichen es durch SHIFT und Klammeraffe. Folgende Befehle stehen Ihnen zur Verfügung:

1. Toolkits:

GETKEY,string
Wartet auf Taste und lädt diese in den angegebenen Strings (zum Beispiel !GETKEY,*a$).
* steht für den Klammeraffen.

BREAK.ON/.OFF
Bestimmt den Zustand der ESC-Taste.

ERROR.ON/.OFF
Ein- und Ausschalten der Fehlerabfangroutine. Nützlich zum Beispiel bei Memory full oder File Type Error. Der Befehl ist allerdings mit Vorsicht zu genießen, da ein Absturz nicht ausgeschlossen ist.

HELP
Listet alle neuen Befehle auf.

SCRLOAD,string
Lädt den Bildschirmspeicher mit dem Namen string ein.

SCRSAVE,string
Speichert den Bildschirmspeicher unter dem Namen string ab.

ONLOWRES
In Mode 1 und 2 kann jetzt mit der Schriftgröße von Mode 0 geschrieben werden.

NORMAL
Schaltet wieder auf normale Textausgabe um.

TEXTGET,adresse,x,xl,y,yl
Liest das durch x, xl, y, yl bestimmte Fenster charakterweise in den ab adresse beginnenden Speicherbereich ein.

TEXTPUT,adresse,x,xl,y,yl
Der mit TEXTGET eingelesene Bereich wird wieder ausgegeben.

GETCHR,x,y,string
Das sich auf x, y befindende Zeichen wird in string eingelesen.

PROTECT
Schützt ein BASIC-Programm gegen List und Edit.

UNPROTECT
Hebt den Schutz wieder auf.

2. Disketten-Befehle:

GETFORMAT
Ermittelt das Format der Diskette im Laufwerk A und übergibt es an die Befehle READ/WRITE-SECTOR und BACKUP/RESTORE.

DATA/IBM/SYSTEM
Setzt das jeweilige Format für die obengenannten Befehle.

DISCSPEED,s
Setzt den Wert für die Warteschleife beim Laden von Programmen. Je kleiner der Wert, um so größer die Zeitersparnis (1=25%).

READSECTOR,drive,track,sector, adresse
Ein Sektor wird ab adresse gelesen.

WRITESECTOR,drive,track,sector

adresse
Ein Sektor wird ab adresse geschrieben.

FORMAT,drive
Formatiert die Diskette in drive (0/1) im vorher bestimmten Format.

GETHEADER,name,anfangsstring, längestring,startstring
Der Header des Programms name wird in den drei Strings als Hexzahl abgelegt, die vorher mit vier Characters definiert werden müssen.

FILE.DISC,name
Kopiert das File name von Disk auf Disk.

FILE.TAPE,name
Kopiert das File name von Disk auf Tape.

BACKUP
Speichert eine komplette Diskettenseite im Headerless-Format auf Kassette ab. Es werden jeweils fünf Spuren gelesen.

RESTORE
Eine mit BACKUP abgespeicherte Disk wird wieder eingelesen.

3. Kassetten-Befehle:

TAPESPEED,s
Setzt die Baud-Rate (Speed Write). Je kleiner s, um so größer ist die Baud-Rate (s=0 entspricht 4440 Baud). Der Wert von s muß zwischen null und 150 liegen.

HEADERLESS.OUT,anfang,länge
Speichert den mit anfang und länge definierten Bereich im Headerless-Format ab.

HEADERLESS.IN,anfang,länge
Liest den Bereich wieder ein.

4. Grafik-Befehle:

FOCUS,x,y
Bestimmt die Vergrößerung in x- und y-Richtung.

TEXTPOS,x,y
Bestimmt die Anfangsposition für die vergrößerte Schrift (Grafikposition).

TYP,s
Wählt den entsprechenden Typ für die vergrößerte Schrift (s=1 bis 4).

INVERSE
Invertiert die Schrift.

REVERSE
Schaltet die Invertierung wieder ab.

MIRROR.ON/.OFF
Schaltet Spiegelung ein beziehungsweise aus.

STEP,s
Mit s wird der pixelweise Abstand zweier Zeichen festgelegt.

TEXT,string
Vergrößert string mit den eingestellten Faktoren.

FILL,x,y
Eine Fläche wird ab x, y ausgemalt.

CIRCLE,flag,x,y,xradius,yradius
Ein Kreis oder eine Ellipse wird gezeichnet.
flag=0: normale Zeichnung
flag=1: ausgefüllte Zeichnung

BOX,flag,x,y,xversatz,yversatz
Ein Rechteck wird gezeichnet. Für flag gilt das gleiche wie bei CIRCLE.

5. Speicheroperationen:

HEXDUMP,adresse
Gibt Hexdump des ab adresse beginnenden Speicherbereiches aus. Es empfiehlt sich Mode 2.

DOKE,adresse,wert
Poked einen 16-Bit-Wert an adresse.

ROMDUMP,adresse
Wie HEXDUMP, nur bezieht sich adresse auf einen ROM-Bereich.

DEEK,adresse
Liest einen 16-Bit-Wert von adresse.

DEEK,adresse,string
Lädt einen 16-Bit-Wert in den String string.

STROKE,textstring,adresse,textlänge
Legt den Textstring ab adresse im RAM ab.

STREEK,textstring,adresse,textlänge
Liest RAM ab adresse in textstring ein.

6. Bildschirm-Manipulationen:

FREEZE
Warteschleife auf Rasterstrahlrücklauf.

SCROMPRESS,adresse,längestring
Komprimiert den Bildschirmspeicher und legt ihn ab adresse ab. Ein komprimierter Screen ist nur noch fünf bis 14 KByte lang. Die genaue Länge wird in längestring übergeben.

SCREXPAND,adresse
Lädt einen mit SCROMPRESS verkürzten Bildschirm von adresse ein. Der alte Screen wird hinausgeschoben.

SCRSCROLL,adresse
Scrollt einen sich bei adresse befindenden Screen in den Bildschirm, wobei der alte Bildschirm hinausgescrollt wird.

LINESCROLL,zeile,textstring
Scrollt textstring in der angegebenen Zeile über den Bildschirm.

SHAPE,x,y,xversatz,yversatz, shapeadresse,oldscreenadresse, flag
flag=0: Der Inhalt des Speicherbereiches wird in dem durch xversatz und yversatz be-

stimmten Bildschirmausschnitt dargestellt, während der Originalausschnitt bei oldscreenadresse abgelegt wird

flag=1: Der alte Ausschnitt wird wieder zurückgeholt

Der SHAPE-Befehl ist nicht grafik-, sondern bildschirmorientiert. Das heißt, x+xver. darf nicht größer als 80, x+yversatz nicht größer als 200 sein.

TRANS.ON
Transparente Darstellung der Shapes.

TRANS.OFF
Wieder normale Darstellung der Shapes

7. Sonstiges:

FIND,suchadresse,suchtextstring, übergabestring
Der Speicher wird ab suchadressestring nach suchtextstring abgesucht und die entsprechende Adresse in übergabestring übergeben

Wird ein falscher Parameter eingegeben, so erscheint eine Fehlermeldung. Bei den Befehlen GETHEADER, SCROMPRESS, FIND und DEEK muß vor der Anwendung der entsprechende String mit vier Bytes vordefiniert werden. Jedem String muß ein Klammeraffe vorangestellt werden. *TB* □

```
1 '******************************  (24B1)
2 '*      EXTENDED-BASIC        *  (247D)
3 '*            VON             *  (24F2)
4 '*      ANDRE SCHROEDER       *  (2477)
5 '*           FUER             *  (2405)
6 '*     SCHNEIDER CPC-WELT     *  (2476)
7 '*          CPC 464        TB*  (2402)
8 '******************************  (24BF)
10 MEMORY &9564                   (0998)
20 LOAD"EXTEN.BIN"                (113B)
40 CALL &9565                     (0902)
50 MODE 1                         (070C)
60 PRINT SPC(10);"EXTENDED-BASIC 2
.0"                               (20AE)
70 PRINT                          (068B)
71 PRINT SPC(11);"37877 Bytes frei
"                                 (1EAA)
72 PRINT                          (068F)
80 PRINT"/HELP zur Befehls";      (1A6E)
90 PRINT"uebersicht eingeben."    (1C31)
100 NEW                           (068F)
```

```
10 '******************************  (24C3)
20 '*         GRAFIK-DEMO        *  (2417)
30 '*     ZU EXTENDED BASIC      *  (24F8)
40 '*           VON              *  (243D)
50 '*      ANDRE SCHROEDER       *  (24D3)
60 '*           FUER             *  (2492)
65 '*     SCHNEIDER CPC-WELT     *  (24EC)
70 '*          CPC 464        TB*  (2480)
80 '******************************  (244E)
90 '                              (07BF)
100 MEMORY 9999:DEFINT a-z        (0F4A)
110 '                             (07E7)
120 '**** AUFBAU DES SCHLUSSBILDES
                                  (24A3)
130 '                             (070E)
140 BORDER 0:FOR a=0 TO 3:INK a,0:
NEXT                              (1BCA)
150 CALL &BC07,&40:MODE 1:/INVERSE
                                  (1AA4)
160 GOSUB 1060:f=2:t=3:y=350      (2038)
170 GOSUB 1070:/REVERSE           (13A8)
180 GOSUB 1060:f=0:t=2:GOSUB 1070 (1C58)
190 LOCATE 1,25:PRINT STRING$(15,3
2);                               (1630)
200 CALL &BC07,&C0:MODE 1         (1018)
210 '                             (07AE)
220 '**** AUFBAU DES SHAPES       (1D26)
230 '                             (07D6)
240 PLOT-2,-2,2:/BOX,1,10,375,204,
80                                (229D)
250 PLOT-2,-2,3:/BOX,1,2,387,204,8
0                                 (2169)
260 GOSUB 1060:GOSUB 1060         (0E29)
270 f=2:t=3:GOSUB 1070            (179E)
280 PLOT-2,-2,1:/BOX,0,2,387,204,8
0                                 (21FF)
290 /SHAPE,0,5,27,44,&C000,10000,0
                                  (20D6)
300 '                             (0764)
310 MODE 1:INK 1,25:INK 3,15      (13C7)
320 INK 0,18:BORDER 18            (0EBE)
330 FOR a=500 TO 10 STEP-20       (15A9)
340 FOR b=0 TO a:NEXT             (136A)
350 x=RND*50+1:y=RND*150+1        (1EE0)
360 /SHAPE,x,y,27,44,&C000,10000,1
                                  (2619)
370 NEXT:GOSUB 1040               (0BC4)
380 '                             (0703)
390 MODE 1:INK 2,18:PEN 2         (1083)
400 /REVERSE                      (0EF8)
410 GOSUB 1060                    (093B)
420 /MIRROR.ON:y=200:t=2:f=3      (26B8)
430 GOSUB 1070                    (09B3)
440 /MIRROR.OFF:GOSUB 1040        (1698)
450 '                             (078F)
460 MODE 1:GOSUB 1060             (0CF6)
470 F=1:x=20:y=366:t=4:yu=5       (2BE8)
```

```
480 GOSUB 1070                          (2918)
490 GOSUB 1060                          (29DB)
500 f=2:t=3                             (12B9)
510 FOR x=30 TO 34 STEP 2               (12DB)
520 GOSUB 1070:NEXT                     (0B36)
530 /INVERSE                            (0E88)
540 GOSUB 1060                          (0941)
550 f=2:t=2:GOSUB 1070                  (1757)
560 /REVERSE                            (0E39)
570 f=2:t=2:GOSUB 1070                  (177F)
580 t=1:f=3:x=1:GOSUB 1070              (1E3F)
590 LOCATE 1,25                         (0AD6)
600 PRINT STRING$(15,32);               (10E5)
610 INK 2,0:PEN 1:GOSUB 1040            (117E)
620 '                                   (07E5)
630 FOR a=1 TO 20                       (0FFD)
640 x=RND*50+1:y=RND*150+1              (1E24)
650 /SHAPE,x,y,27,44,10000,11500,0
                                        (26A4)
660 FOR b=1 TO 1000:NEXT                (1259)
670 /SHAPE,x,y,27,44,10000,11500,1
                                        (26D0)
680 NEXT:/TRANS.ON                      (11FC)
690 /SHAPE,2,50,70,67,32800,11189,
0                                       (2424)
700 '                                   (0784)
710 a$="   Extended Basic Version
2.0"                                    (2907)
720 a$=a$+STRING$(9,32)                 (179A)
730 FOR a=8 TO 15                       (0F91)
740 /LINESCROLL,a,@a$                   (1C4E)
750 PEN INT(RND*3+1):NEXT               (12E5)
760 GOSUB 1040                          (0958)
770 /CIRCLE,1,275,230,260,67:GOSUB
 1040                                   (2208)
780 /SHAPE,2,50,70,67,32800,11189,
1                                       (24DD)
790 PLOT-2,-2,2                         (0DC1)
800 /FILL,240,260:/FILL,110,220         (1F9E)
810 /FILL,240,200:/FILL,520,220         (1FE6)
820 '                                   (0776)
830 FOR a=40 TO 600 STEP 80             (1496)
840 PLOT-2,-2,INT(RND*3+1)              (164E)
850 x=RND*50+20:y=RND*50+20             (205A)
860 /CIRCLE,1,a,130,x,y                 (21B6)
870 /CIRCLE,1,640-a,310,x,y             (2605)
880 NEXT                                (06A6)
890 FOR a=1 TO 50                       (0F7D)
900 FOR b=1 TO 300:NEXT                 (12A2)
910 FOR c=1 TO 3                        (0EDD)
920 INK c,RND*25+1                      (120E)
930 NEXT:NEXT:INK 1,25                  (0E65)
940 '                                   (0765)
950 INK 2,0:INK 3,15:GOSUB 1040:CL
S                                       (16E1)
960 /SHAPE,0,3,70,67,32800,11189,1
                                        (23E0)
970 /SHAPE,0,122,70,67,32800,11189
,1                                      (2424)
980 GOSUB 1040                          (0911)
990 FOR y=1 TO 151 STEP 50              (12CF)
1000 FOR x=5 TO 40 STEP 35              (12E1)
1010 /SHAPE,x,y,27,44,&C000,10000,
1                                       (2630)
1020 NEXT:NEXT:GOSUB 1040               (0DAB)
1030 /SCRSCROLL,16384:PEN 1:END         (194F)
1040 FOR a=1 TO 5000:NEXT:RETURN        (14DF)
1050 '                                  (0743)
1060 READ t$,x,y,xv,yv,t,f              (2A72)
1070 /TEXTPOS,x,y                       (184F)
1080 /TYP,t:PLOT-2,-2,f                 (1B54)
1090 /FOCUS,xv,yv                       (18B8)
1100 /TEXT,@t$                          (11B6)
1110 RETURN                             (06D7)
1120 '                                  (07CF)
1130 DATA * * * * * * * *,2,150,6,10
,4,3                                    (232F)
1140 DATA ENDE DER DEMO,2,250,6,10
,1,1                                    (23EA)
1150 DATA EXTENDED,6,370,3,3,4,1        (1DC0)
1160 DATA BASIC,2,340,5,4,1,1           (1AD6)
1170 DATA BY ANDRE S.,2,250,7,8,4,
1                                       (20DB)
1180 DATA DAS IST EINE,20,370,6,6,
2,3                                     (22E0)
1190 DATA DEMO,30,250,17,18,1,1         (1C8F)
1200 DATA NICHT WAHR ?,10,80,6,8,1
,1                                      (21E5)
```

```
100 '*****************************     (2234)
110 '*  DATALADER ERZEUGT DAS  *       (22BE)
120 '*  BINAERFILE (EXTEN.BIN) *       (22C6)
130 '*          VON            *       (2293)
140 '*      ANDRE SCHROEDER    *       (22FE)
150 '*          FUER           *       (2228)
160 '*    SCHNEIDER CPC-WELT   *       (22F7)
170 '*        CPC 464        TB*       (2251)
180 '*****************************     (22D4)
200 MEMORY &9564                        (0914)
650 a=&9565:e=&A672:zb=1000:e=e+1       (2C4D)
660 FOR i=a TO e:IF i=e THEN SAVE"
EXTEN.BIN",B,&9565,&110D:END             (3BEF)
670 READ d$:POKE i,VAL("&"+d$)          (1D5A)
730 IF i<e THEN NEXT i                  (1526)
1001 DATA 3A,CE,97,FE,01,C8,3E,01       (1E80)
1002 DATA 32,CE,97,01,79,95,21,CA       (1EB8)
1003 DATA 97,C3,D1,BC,29,96,C3,6D       (1EAA)
1004 DATA A2,C3,7D,A2,C3,CA,A4,C3       (1EAD)
1005 DATA BE,A4,C3,C4,A4,C3,5F,A2       (1EAC)
1006 DATA C3,38,A2,C3,97,A0,C3,2E       (1E81)
1007 DATA 9F,C3,51,A3,C3,9B,A3,C3       (1EA9)
1008 DATA EA,A2,C3,3E,A4,C3,43,A4       (1E22)
1009 DATA C3,B8,A3,C3,D0,A2,C3,E5       (1E79)
1010 DATA A2,C3,D2,A1,C3,D8,A1,C3       (1EE2)
1011 DATA DE,A1,C3,E4,A1,C3,F1,A1       (1ECB)
1012 DATA C3,1F,A4,C3,12,A4,C3,0C       (1E19)
1013 DATA A0,C3,24,A0,C3,D6,A5,C3       (1EB2)
1014 DATA 49,9E,C3,4E,9E,C3,54,9E       (1E1A)
1015 DATA C3,59,9E,C3,F9,9E,C3,FF       (1E61)
1016 DATA 9E,C3,02,9E,C3,0F,9E,C3       (1E98)
1017 DATA E3,9D,C3,CF,9D,C3,05,9F       (1E4A)
1018 DATA C3,16,9F,C3,6B,9D,C3,F2       (1E75)
```

```
1019 DATA 9E,C3,EE,9E,C3,67,9B,C3 <1EC1>
1020 DATA AE,9B,C3,BD,9B,C3,D0,9B <1EF0>
1021 DATA C3,41,9B,C3,47,9B,C3,4D <1ECB>
1022 DATA 9B,C3,7A,9B,C3,ED,9B,C3 <1E92>
1023 DATA 05,9B,C3,ED,9A,C3,23,A6 <1E88>
1024 DATA C3,21,9B,C3,4F,99,C3,CF <1E85>
1025 DATA 97,C3,6A,98,54,45,58,54 <1E8A>
1026 DATA 47,45,D4,54,45,58,54,50 <1E6F>
1027 DATA 55,D4,53,48,41,50,C5,54 <1E03>
1028 DATA 52,41,4E,53,2E,4F,CE,54 <1E76>
1029 DATA 52,41,4E,53,2E,4F,46,C6 <1E43>
1030 DATA 47,45,54,4B,45,D9,47,45 <1EAB>
1031 DATA 54,43,48,D2,53,43,52,53 <1E53>
1032 DATA 43,52,4F,4C,CC,4C,49,4E <1E5A>
1033 DATA 45,53,43,52,4F,4C,CC,53 <1E33>
1034 DATA 43,52,43,4F,4D,50,52,45 <1EE5>
1035 DATA 53,D3,53,43,52,45,58,50 <1ED1>
1036 DATA 41,4E,C4,48,45,58,44,55 <1E48>
1037 DATA 4D,D0,50,52,4F,54,45,43 <1E61>
1038 DATA D4,55,4E,50,52,4F,54,45 <1E0E>
1039 DATA 43,D4,47,45,54,48,45,41 <1EE3>
1040 DATA 44,45,D2,54,41,50,45,53 <1E8C>
1041 DATA 50,45,45,C4,44,49,53,43 <1EF8>
1042 DATA 53,50,45,45,C4,44,41,54 <1E71>
1043 DATA C1,53,59,53,54,45,CD,49 <1ED7>
1044 DATA 42,CD,52,45,41,44,53,45 <1E61>
1045 DATA 43,54,4F,D2,57,52,49,54 <1E0B>
1046 DATA 45,53,45,43,54,4F,D2,48 <1EC4>
1047 DATA 45,41,44,45,52,4C,45,53 <1EEB>
1048 DATA 53,2E,4F,55,D4,48,45,41 <1E13>
1049 DATA 44,45,52,4C,45,53,53,2E <1EEC>
1050 DATA 49,CE,46,49,4C,45,2E,54 <1E04>
1051 DATA 41,50,C5,46,49,4C,45,2E <1EB6>
1052 DATA 44,49,53,C3,46,4F,52,4D <1EF9>
1053 DATA 41,D4,42,52,45,41,4B,2E <1E47>
1054 DATA 4F,CE,42,52,45,41,4B,2E <1E1C>
1055 DATA 4F,46,C6,42,41,43,4B,55 <1EE0>
1056 DATA D0,52,45,53,54,4F,52,C5 <1E23>
1057 DATA 45,52,52,4F,52,2E,4F,CE <1E22>
1058 DATA 45,52,52,4F,52,2E,4F,46 <1E71>
1059 DATA C6,53,54,52,45,45,CB,53 <1EB3>
1060 DATA 54,52,4F,4B,C5,44,45,45 <1E95>
1061 DATA CB,44,4F,4B,C5,4F,4E,4C <1E1D>
1062 DATA 4F,57,52,45,D3,4E,4F,52 <1E3E>
1063 DATA 4D,41,CC,46,49,4E,C4,52 <1ED8>
1064 DATA 4F,4D,44,55,4D,D0,46,52 <1EA9>
1065 DATA 45,45,5A,C5,46,4F,43,55 <1EE3>
1066 DATA D3,4D,49,52,52,4F,52,2E <1E0F>
1067 DATA 4F,CE,4D,49,52,52,4F,52 <1EF8>
1068 DATA 2E,4F,46,C6,53,54,45,D0 <1E3B>
1069 DATA 49,4E,56,45,52,53,C5,52 <1E0C>
1070 DATA 45,56,45,52,53,C5,54,45 <1E3F>
1071 DATA 58,54,50,4F,D3,54,59,D0 <1ED3>
1072 DATA 54,45,58,D4,53,43,52,4C <1E1C>
1073 DATA 4F,41,C4,53,43,52,53,41 <1E52>
1074 DATA 56,C5,48,45,4C,D0,47,45 <1E45>
1075 DATA 54,46,4F,52,4D,41,D4,43 <1EE6>
1076 DATA 49,52,43,4C,C5,42,4F,D8 <1E02>
1077 DATA 46,49,4C,CC,00,FC,A6,79 <1E95>
1078 DATA 95,00,06,05,CD,5B,A1,FE <1E6B>
1079 DATA 00,C8,ED,53,68,98,DD,66 <1EC2>
1080 DATA 03,DD,6E,02,22,66,98,DD <1E70>
1081 DATA 66,05,DD,6E,04,DD,56,07 <1EFB>
1082 DATA DD,5E,06,DD,7E,08,FE,01 <1E34>
1083 DATA CA,22,98,CD,58,98,ED,4B <1E03>
1084 DATA 66,98,EB,09,EB,CD,4E,98 <1E84>
1085 DATA ED,4B,68,98,ED,42,CD,4E <1EDC>
1086 DATA 98,ED,4B,66,98,EB,ED,42 <1ECA>
1087 DATA EB,CD,4E,98,ED,4B,68,98 <1E41>
1088 DATA 09,CD,4E,98,C9,ED,53,62 <1E7F>
1089 DATA 98,ED,4B,66,98,EB,09,EB <1E07>
1090 DATA ED,53,64,98,ED,4B,68,98 <1E26>
1091 DATA ED,5B,62,98,CD,58,98,ED <1E55>
1092 DATA 5B,64,98,CD,4E,98,2B,2B <1E92>
1093 DATA 0B,0B,78,FE,FF,C8,C3,35 <1E90>
1094 DATA 98,C5,E5,D5,CD,F6,BB,D1 <1E24>
1095 DATA E1,C1,C9,C5,E5,D5,CD,C0 <1EF3>
1096 DATA BB,D1,E1,C1,C9,00,00,00 <1EF3>
1097 DATA 00,00,00,00,00,06,02,CD <1E71>
1098 DATA 5B,A1,FE,00,C8,ED,53,47 <1ED4>
1099 DATA 99,DD,66,03,DD,6E,02,22 <1EE1>
1100 DATA 45,99,3A,C8,B1,FE,00,C2 <1E68>
1101 DATA 8C,98,3E,04,C3,98,98,FE <1E96>
1102 DATA 01,C2,96,98,3E,02,C3,98 <1EF5>
1103 DATA 98,3E,01,32,49,99,3E,00 <1E80>
1104 DATA 32,4B,99,2A,47,99,22,41 <1EA9>
1105 DATA 99,21,23,23,22,07,99,2A <1ECE>
1106 DATA 45,99,22,43,99,3E,00,32 <1EA9>
1107 DATA 4C,99,EB,2A,41,99,CD,F0 <1EB7>
1108 DATA BB,FE,00,C2,25,99,2A,43 <1E83>
1109 DATA 99,ED,4B,49,99,3A,4C,99 <1EFA>
1110 DATA FE,01,20,04,ED,42,18,01 <1EA9>
1111 DATA 09,22,43,99,EB,2A,41,99 <1E8C>
1112 DATA CD,F0,BB,FE,00,20,06,7C <1E03>
1113 DATA FE,03,D8,18,D9,3A,4C,99 <1E1A>
1114 DATA FE,01,C2,0F,99,ED,5B,43 <1E49>
1115 DATA 99,2A,41,99,E5,CD,C0,BB <1E47>
1116 DATA ED,5B,4D,99,E1,E5,CD,F6 <1E33>
1117 DATA BB,E1,23,23,22,41,99,C3 <1E6B>
1118 DATA AC,98,3E,01,32,4C,99,2A <1E7F>
1119 DATA 43,99,22,4D,99,ED,5B,45 <1EB6>
1120 DATA 99,ED,53,43,99,C3,DA,98 <1E04>
1121 DATA 3A,4B,99,FE,01,C8,21,2B <1E39>
1122 DATA 2B,22,07,99,3E,01,32,4B <1E95>
1123 DATA 99,2A,47,99,2B,2B,22,41 <1E41>
1124 DATA 99,C3,AC,98,00,00,00,00 <1E0F>
1125 DATA 00,00,00,00,00,00,00,00 <1E70>
1126 DATA 00,00,06,05,CD,5B,A1,FE <1E0B>
1127 DATA 00,C8,21,00,80,22,E5,9A <1EDC>
1128 DATA ED,53,DD,9A,DD,66,03,DD <1E19>
1129 DATA 6E,02,22,DF,9A,DD,66,05 <1ED2>
1130 DATA DD,6E,04,22,E3,9A,DD,66 <1EFF>
1131 DATA 07,DD,6E,06,22,E1,9A,DD <1ED7>
1132 DATA 7E,08,FE,01,CA,89,99,3E <1EA8>
1133 DATA E9,C3,8B,99,3E,F6,32,0D <1EB3>
1134 DATA 9A,32,36,9A,32,61,9A,32 <1E2F>
1135 DATA 8C,9A,06,0A,2A,DF,9A,CD <1E11>
1136 DATA D3,9A,ED,5B,DD,9A,CD,C1 <1EBC>
1137 DATA BD,22,EB,9A,21,00,00,22 <1E57>
1138 DATA DF,9A,2A,DF,9A,ED,5B,DD <1E07>
1139 DATA 9A,7C,BA,C2,C0,99,13,13 <1E71>
1140 DATA 7D,BB,D0,2A,DF,9A,3A,EB <1E8F>
1141 DATA 9A,47,CD,D3,9A,11,0A,00 <1E60>
1142 DATA CD,C1,BD,22,E7,9A,2A,DD <1E87>
```

```
1143 DATA 9A,3A,EB,9A,47,CD,D3,9A <1E5D>
1144 DATA 11,0A,00,CD,C1,BD,22,E9 <1EDD>
1145 DATA 9A,2A,E1,9A,ED,5B,E7,9A <1E35>
1146 DATA 19,EB,2A,E3,9A,ED,4B,DD <1E70>
1147 DATA 9A,09,CD,E9,BB,2A,E1,9A <1E07>
1148 DATA ED,5B,E7,9A,ED,52,EB,2A <1EAA>
1149 DATA E3,9A,ED,4B,DD,9A,09,CD <1E83>
1150 DATA F6,BB,2A,E1,9A,ED,5B,E9 <1EFE>
1151 DATA 9A,19,EB,2A,E3,9A,ED,4B <1E86>
1152 DATA DF,9A,09,CD,E9,BB,2A,E1 <1EB9>
1153 DATA 9A,ED,5B,E9,9A,ED,52,EB <1E8F>
1154 DATA 2A,E3,9A,ED,4B,DF,9A,09 <1EEB>
1155 DATA CD,F6,BB,2A,E1,9A,ED,5B <1E71>
1156 DATA E7,9A,ED,52,EB,2A,E3,9A <1E26>
1157 DATA ED,4B,DD,9A,ED,42,CD,E9 <1EFE>
1158 DATA BB,2A,E1,9A,ED,5B,E7,9A <1E73>
1159 DATA 19,EB,2A,E3,9A,ED,4B,DD <1E8A>
1160 DATA 9A,ED,42,CD,F6,BB,2A,E1 <1EBA>
1161 DATA 9A,ED,5B,E9,9A,ED,52,EB <1E9F>
1162 DATA 2A,E3,9A,ED,4B,DF,9A,ED <1ECE>
1163 DATA 42,CD,E9,BB,2A,E1,9A,ED <1EA3>
1164 DATA 5B,E9,9A,19,EB,2A,E3,9A <1E77>
1165 DATA ED,4B,DF,9A,ED,42,CD,F6 <1E0F>
1166 DATA BB,2A,E5,9A,7C,FE,80,D2 <1EBD>
1167 DATA AE,9A,2A,DF,9A,06,04,CD <1E36>
1168 DATA D3,9A,01,06,00,09,EB,2A <1EF5>
1169 DATA E5,9A,19,22,E5,9A,C3,CC <1E29>
1170 DATA 9A,2A,DF,9A,ED,5B,DD,9A <1E6E>
1171 DATA ED,52,06,04,CD,D3,9A,01 <1E43>
1172 DATA 0A,00,09,EB,2A,E5,9A,19 <1ECC>
1173 DATA 22,E5,9A,21,DD,9A,35,21 <1E5E>
1174 DATA DF,9A,34,C3,AF,99,54,5D <1E7E>
1175 DATA 21,00,00,0B,19,10,FD,C9 <1E80>
1176 DATA 45,00,47,00,40,01,C8,00 <1EA7>
1177 DATA 64,80,46,00,46,00,0A,00 <1ED4>
1178 DATA CD,15,9B,CD,8C,BC,CD,7A <1E5D>
1179 DATA BC,21,00,C0,11,00,40,3E <1E43>
1180 DATA 02,CD,98,BC,CD,8F,BC,C9 <1EF4>
1181 DATA CD,15,9B,CD,77,BC,21,00 <1EB5>
1182 DATA C0,CD,83,BC,CD,7A,BC,C9 <1E46>
1183 DATA 1A,47,13,1A,6F,13,1A,67 <1E2E>
1184 DATA 11,00,C0,C9,3E,C9,32,5A <1E83>
1185 DATA BB,21,3E,9B,CD,D4,BC,D0 <1E28>
1186 DATA AF,CD,1B,00,3A,9F,A8,3D <1E49>
1187 DATA 32,37,A2,3E,CF,32,5A,BB <1ED8>
1188 DATA C9,44,49,D2,3E,C2,32,6A <1E4E>
1189 DATA 9C,C9,3E,CA,32,6A,9C,C9 <1EEB>
1190 DATA 06,02,CD,5B,A1,FE,00,C8 <1EC4>
1191 DATA ED,53,E0,9B,ED,53,E2,9B <1E64>
1192 DATA DD,66,03,DD,6E,02,22,DE <1E6C>
1193 DATA 9B,C9,06,02,CD,5B,A1,FE <1EB3>
1194 DATA 00,C8,ED,53,E4,9B,DD,7E <1E4B>
1195 DATA 02,32,E6,9B,C9,06,01,CD <1E25>
1196 DATA 5B,A1,FE,00,C8,7B,FE,01 <1E9A>
1197 DATA C2,8F,9B,21,F5,9C,22,EB <1E5F>
1198 DATA 9C,C9,FE,02,C2,9B,9B,21 <1E45>
1199 DATA 14,9D,22,EB,9C,C9,FE,03 <1E93>
1200 DATA C2,A7,9B,21,29,9D,22,EB <1EC9>
1201 DATA 9C,C9,21,48,9D,22,EB,9C <1EA0>
1202 DATA C9,3E,C6,32,7D,9C,3E,0F <1EA3>
1203 DATA 32,79,9C,3E,01,C3,C9,9B <1EA8>
1204 DATA 3E,D6,32,7D,9C,3E,01,32 <1E8A>
1205 DATA 79,9C,3E,0F,32,F7,9B,32 <1EF8>
1206 DATA 86,9C,C9,7B,32,D5,9B,C9 <1ED6>
1207 DATA 00,00,0F,00,00,00,20,22 <1EB7>
1208 DATA 48,02,00,C8,00,4D,22,08 <1E73>
1209 DATA 00,08,00,30,20,27,44,42 <1EFB>
1210 DATA 06,01,CD,5B,A1,FE,00,C8 <1E96>
1211 DATA 00,3E,0F,32,D7,9B,1A,47 <1E49>
1212 DATA 3D,32,DC,9B,3E,00,32,DD <1E5D>
1213 DATA 9B,32,DB,9B,32,D9,9B,3A <1E2D>
1214 DATA C8,B1,FE,00,C2,20,9C,21 <1EBA>
1215 DATA 34,34,22,8D,9C,22,8F,9C <1E5D>
1216 DATA C3,40,9C,FE,01,C2,34,9C <1E29>
1217 DATA 21,34,34,22,8D,9C,21,00 <1E9A>
1218 DATA 00,22,8F,9C,C3,40,9C,21 <1E19>
1219 DATA 34,00,22,8D,9C,21,00,00 <1E59>
1220 DATA 22,8F,9C,13,1A,6F,13,1A <1E9C>
1221 DATA 67,3E,1F,CD,5A,BB,3E,01 <1E43>
1222 DATA CD,5A,BB,3E,19,CD,5A,BB <1E16>
1223 DATA 7E,CD,5A,BB,23,10,F9,2A <1E69>
1224 DATA D7,9B,ED,5B,D9,9B,E5,D5 <1EF2>
1225 DATA CD,F0,BB,FE,00,CA,70,9C <1EC7>
1226 DATA CD,CE,9C,CD,56,9D,D1,E1 <1EFC>
1227 DATA 3A,D7,9B,FE,01,CA,85,9C <1E12>
1228 DATA D6,02,32,D7,9B,C3,5C,9C <1EDB>
1229 DATA 3E,0F,32,D7,9B,21,D9,9B <1ECA>
1230 DATA 34,34,00,00,2A,DE,9B,ED <1EB3>
1231 DATA 5B,E6,9B,19,22,DE,9B,2A <1E91>
1232 DATA E2,9B,22,E0,9B,3A,DB,9B <1E5C>
1233 DATA 3C,32,DB,9B,FE,08,C2,5C <1E50>
1234 DATA 9C,3A,DC,9B,47,3A,DD,9B <1ED3>
1235 DATA B8,C8,3C,32,DD,9B,3E,00 <1E27>
1236 DATA 32,DB,9B,2A,DE,9B,ED,4B <1E73>
1237 DATA D5,9B,09,22,DE,9B,C3,5C <1E8C>
1238 DATA 9C,ED,5B,DE,9B,2A,E0,9B <1E62>
1239 DATA 3A,E6,9B,06,00,4F,CD,63 <1E93>
1240 DATA 9D,3A,E4,9B,4F,09,22,EA <1E3D>
1241 DATA 9B,ED,53,E8,9B,CD,F5,9C <1E78>
1242 DATA C9,2A,EA,9B,ED,5B,E8,9B <1EF7>
1243 DATA E5,D5,06,00,09,C5,CD,C0 <1E2D>
1244 DATA BB,3A,E6,9B,4F,CD,63,9D <1ED6>
1245 DATA CD,F6,BB,C1,D1,E1,0B,0B <1E44>
1246 DATA 79,FE,FE,D0,C3,F5,9C,CD <1EB7>
1247 DATA C0,BB,3A,E6,9B,06,00,4F <1EC6>
1248 DATA CD,63,9D,3A,E4,9B,4F,09 <1EF0>
1249 DATA CD,F6,BB,C9,CD,C0,BB,3A <1E48>
1250 DATA E6,9B,06,00,4F,CD,63,9D <1EBB>
1251 DATA E5,D5,CD,F6,BB,D1,E1,CD <1E45>
1252 DATA C0,BB,3A,E4,9B,4F,09,CD <1E29>
1253 DATA F6,BB,C9,CD,C0,BB,3A,E4 <1E1E>
1254 DATA 9B,06,00,4F,09,CD,F6,BB <1E10>
1255 DATA C9,2A,E0,9B,ED,5B,E4,9B <1EA9>
1256 DATA ED,52,22,E0,9B,C9,E5,62 <1E49>
1257 DATA 6B,09,54,5D,E1,C9,06,03 <1E74>
1258 DATA CD,5B,A1,FE,00,C8,ED,53 <1E8B>
1259 DATA C8,9D,DD,66,05,DD,6E,04 <1EAA>
1260 DATA 22,CD,9D,DD,66,03,DD,6E <1E49>
1261 DATA 02,7E,32,CA,9D,23,7E,5F <1E21>
1262 DATA 23,7E,57,ED,53,CB,9D,7C <1EC9>
1263 DATA FE,C0,C8,C3,A5,9D,2A,CD <1E64>
1264 DATA 9D,23,22,CD,9D,C3,94,9D <1E35>
1265 DATA 3A,CA,9D,47,ED,5B,CB,9D <1E28>
1266 DATA 2A,CD,9D,1A,BE,C2,9B,9D <1E8F>
```

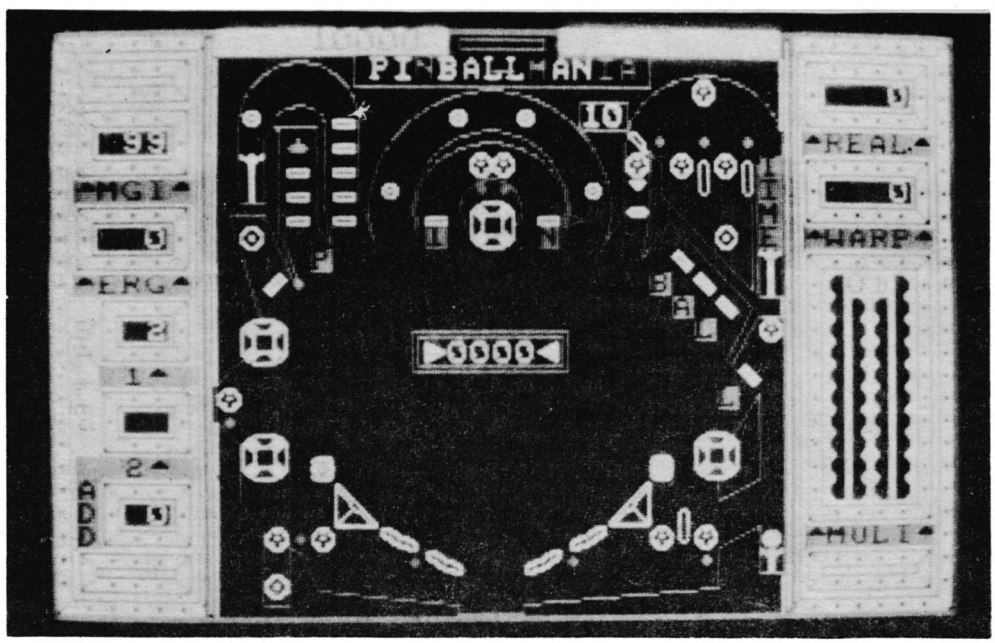
```
1267 DATA 23,13,05,78,FE,00,C2,B0 (1EA4)
1268 DATA 9D,ED,5B,C8,9D,2A,CD,9D (1EB0)
1269 DATA C3,2B,A1,07,02,05,0C,27 (1E41)
1270 DATA 0C,27,06,02,CD,5B,A1,FE (1E04)
1271 DATA 00,C8,DD,66,03,DD,6E,02 (1E10)
1272 DATA 7B,77,23,7A,77,C9,FE,02 (1EAA)
1273 DATA C2,F7,9D,DD,46,03,DD,4E (1E4D)
1274 DATA 02,0A,6F,03,0A,67,CD,2B (1EFB)
1275 DATA A1,C9,13,1A,CD,28,A3,1B (1EA1)
1276 DATA 1A,CD,28,A3,C9,06,03,CD (1ED4)
1277 DATA 5B,A1,FE,00,C8,3E,01,C3 (1ED0)
1278 DATA 19,9E,06,03,CD,5B,A1,FE (1E00)
1279 DATA 00,C8,3E,00,32,48,9E,7B (1EE7)
1280 DATA 32,47,9E,DD,66,03,DD,6E (1E45)
1281 DATA 02,DD,56,05,DD,5E,04,12 (1E73)
1282 DATA 13,1A,4F,13,1A,57,79,5F (1EE7)
1283 DATA 3A,48,9E,FE,00,C2,3E,9E (1EB5)
1284 DATA EB,3A,47,9E,06,00,4F,ED (1E7E)
1285 DATA B0,C9,00,00,3E,C3,C3,50 (1E11)
1286 DATA 9E,3E,C9,32,EE,BD,C9,3E (1EF3)
1287 DATA 00,C3,5B,9E,3E,01,32,E9 (1E88)
1288 DATA 9E,06,08,3E,00,32,EA,9E (1E44)
1289 DATA C5,3A,E9,9E,FE,01,CA,7E (1EB9)
1290 DATA 9E,C3,75,9E,C1,10,F1,C9 (1EB1)
1291 DATA CD,87,9E,CD,D8,9E,C3,71 (1E60)
1292 DATA 9E,CD,D3,9E,CD,87,9E,C3 (1E44)
1293 DATA 71,9E,21,9D,41,3A,EA,9E (1E01)
1294 DATA 57,C6,05,32,EA,9E,0E,09 (1E29)
1295 DATA 3A,37,A2,81,4F,CD,BB,9E (1E4F)
1296 DATA CD,AB,9E,3A,37,A2,3C,B9 (1E7C)
1297 DATA 28,0B,0D,C3,9A,9E,D5,11 (1E98)
1298 DATA 00,02,19,D1,C9,14,3A,EA (1E27)
1299 DATA 9E,BA,C8,C3,93,9E,1E,00 (1EE0)
1300 DATA 3A,E9,9E,FE,01,CA,CA,9E (1EFA)
1301 DATA 3E,3C,C3,CC,9E,3E,3F,32 (1E30)
1302 DATA EB,9E,DF,EB,9E,C9,3E,A1 (1E73)
1303 DATA C3,DA,9E,3E,9E,32,E6,9E (1E3E)
1304 DATA 3E,16,21,9D,41,11,00,5A (1E48)
1305 DATA CD,00,BC,C9,00,00,00,C0 (1E28)
1306 DATA 07,CD,19,BD,C9,DF,F6,9E (1E23)
1307 DATA C9,EA,A2,FC,3E,C9,32,04 (1E29)
1308 DATA AC,C9,3E,F1,32,04,AC,C9 (1E03)
1309 DATA 3A,C8,B1,32,2C,9F,3E,00 (1EAF)
1310 DATA 32,C8,B1,3E,FF,32,D0,B1 (1E31)
1311 DATA C9,3A,2C,9F,32,C8,B1,FE (1E32)
1312 DATA 01,C2,26,9F,3E,44,C3,28 (1E24)
1313 DATA 9F,3E,40,32,D0,B1,C9,00 (1ED4)
1314 DATA 00,06,02,CD,5B,A1,FE,00 (1EB7)
1315 DATA C8,1A,32,0A,A0,13,1A,6F (1E4C)
1316 DATA 13,1A,67,22,FD,9F,DD,7E (1E96)
1317 DATA 02,CD,BE,9F,3A,C8,B1,47 (1EB5)
1318 DATA CD,D2,9F,3E,08,32,FF,9F (1E6E)
1319 DATA CD,19,BD,3A,09,A0,47,3A (1EF5)
1320 DATA FB,9F,B8,CA,90,9F,3C,32 (1E86)
1321 DATA FB,9F,2A,05,A0,22,00,A0 (1EAD)
1322 DATA 2A,00,A0,11,00,08,19,22 (1ED7)
1323 DATA 00,A0,54,5D,23,01,4F,00 (1E4E)
1324 DATA ED,B0,36,00,3A,FF,9F,3D (1EFF)
1325 DATA FE,00,CA,50,9F,32,FF,9F (1E11)
1326 DATA C3,6D,9F,3E,00,32,FB,9F (1EE4)
1327 DATA 3E,1F,CD,5A,BB,3A,07,A0 (1E21)
1328 DATA CD,5A,BB,3A,08,A0,CD,5A (1E27)
1329 DATA BB,3A,0A,A0,FE,00,C8,3D (1EBA)
1330 DATA 32,0A,A0,2A,FD,9F,7E,CD (1E1D)
1331 DATA 5A,BB,23,22,FD,9F,C3,67 (1E62)
1332 DATA 9F,32,08,A0,21,B0,B7,11 (1E3B)
1333 DATA 50,00,19,3D,FE,00,C2,C7 (1E49)
1334 DATA 9F,22,05,A0,C9,3E,13,32 (1E6A)
1335 DATA 07,A0,3E,03,32,09,A0,78 (1E28)
1336 DATA FE,01,C2,ED,9F,3E,27,32 (1E62)
1337 DATA 07,A0,3E,01,32,09,A0,78 (1EA9)
1338 DATA FE,02,C0,3E,4F,32,07,A0 (1EBF)
1339 DATA 3E,00,32,09,A0,C9,44,4C (1E5E)
1340 DATA 4F,4B,42,49,4E,01,00,FF (1E81)
1341 DATA 53,50,49,52,49,54,32,D5 (1EF5)
1342 DATA 21,1C,A0,CD,D4,BC,D0,AF (1E48)
1343 DATA CD,1B,00,D1,C3,24,A0,54 (1EA1)
1344 DATA 41,50,45,2E,4F,55,D4,1A (1E30)
1345 DATA 32,58,AF,13,1A,6F,13,1A (1ECE)
1346 DATA 67,11,00,BF,3A,58,AF,06 (1E33)
1347 DATA 00,4F,ED,B0,21,47,A0,11 (1E31)
1348 DATA 00,AF,01,50,00,ED,B0,C3 (1E7E)
1349 DATA 00,AF,21,00,BF,3A,58,AF (1E4E)
1350 DATA 47,11,00,A7,CD,77,BC,2A (1E9F)
1351 DATA 6D,A7,22,50,AF,2A,6F,A7 (1EFA)
1352 DATA 22,52,AF,3A,67,A7,32,54 (1EFE)
1353 DATA AF,2A,6A,A7,22,56,AF,CD (1E87)
1354 DATA 83,BC,CD,7A,BC,CD,06,BB (1EE5)
1355 DATA 21,00,BF,3A,58,AF,47,11 (1E71)
1356 DATA 00,C0,CD,8C,BC,2A,56,AF (1E9E)
1357 DATA ED,5B,50,AF,ED,4B,52,AF (1E4C)
1358 DATA 3A,54,AF,CD,98,BC,CD,8F (1E81)
1359 DATA BC,C9,06,01,CD,5B,A1,FE (1EBC)
1360 DATA 00,C8,21,00,C0,ED,52,22 (1E3C)
1361 DATA 28,A1,3E,00,32,27,A1,3E (1EEF)
1362 DATA 08,32,26,A1,21,B0,BF,22 (1E7E)
1363 DATA 22,A1,22,20,A1,CD,F6,A0 (1E55)
1364 DATA 54,5D,23,01,4F,00,ED,B0 (1EB3)
1365 DATA 2B,44,4D,2A,22,A1,ED,5B (1E87)
1366 DATA 28,A1,ED,52,3A,27,A1,16 (1E80)
1367 DATA 00,5F,19,7E,02,2A,22,A1 (1E88)
1368 DATA 7C,FE,FF,C2,BA,A0,7D,FE (1EA3)
1369 DATA 80,C2,BA,A0,3A,27,A1,FE (1ED3)
1370 DATA 50,C8,3C,32,27,A1,C3,AC (1E0C)
1371 DATA A0,3A,26,A1,2A,22,A1,FE (1EF7)
1372 DATA 08,C2,14,A1,3E,01,32,26 (1ECD)
1373 DATA A1,11,50,00,2A,20,A1,19 (1EC4)
1374 DATA 22,22,A1,22,20,A1,C9,11 (1E2D)
1375 DATA 00,08,19,22,22,A1,3C,32 (1E3D)
1376 DATA 26,A1,C9,00,00,00,00,00 (1E05)
1377 DATA 00,00,00,00,00,00,3E,04 (1EEE)
1378 DATA 12,13,1A,4F,13,1A,47,7C (1EE2)
1379 DATA 0F,0F,0F,0F,CD,4D,A1,7C (1E8F)
1380 DATA CD,4D,A1,7D,0F,0F,0F,0F (1E0E)
1381 DATA CD,4D,A1,7D,CD,4D,A1,C9 (1E4F)
1382 DATA E6,0F,FE,0A,DA,56,A1,C6 (1E76)
1383 DATA 07,C6,30,02,03,C9,B8,DA (1E2A)
1384 DATA 69,A1,C2,6F,A1,C9,21,B9 (1E17)
1385 DATA A1,C3,72,A1,21,7D,A1,C3 (1E17)
1386 DATA 72,A1,21,9A,A1,7E,FE,00 (1E44)
1387 DATA C8,CD,5A,BB,23,C3,72,A1 (1E12)
1388 DATA 0A,0D,2A,2A,2A,20,50,41 (1EB0)
1389 DATA 52,41,4D,45,54,45,52,20 (1E74)
1390 DATA 4D,49,53,53,49,4E,47,20 (1E51)
```

```
1391 DATA 2A,2A,2A,07,00,0A,0D,2A <1E8F>
1392 DATA 2A,2A,20,54,4F,4F,20,4D <1E0E>
1393 DATA 41,4E,59,20,50,41,52,41 <1EEF>
1394 DATA 4D,45,54,45,52,53,20,2A <1E5B>
1395 DATA 2A,2A,07,00,0A,0D,2A,2A <1E3A>
1396 DATA 2A,20,42,41,44,20,50,41 <1EAD>
1397 DATA 52,41,4D,45,54,45,52,20 <1E84>
1398 DATA 2A,2A,2A,07,00,3E,C0,32 <1EBD>
1399 DATA 37,A2,C9,3E,40,32,37,A2 <1E14>
1400 DATA C9,3E,00,32,37,A2,C9,06 <1E39>
1401 DATA 04,CD,5B,A1,FE,00,C8,3E <1E8F>
1402 DATA 84,C3,FB,A1,06,04,CD,5B <1E91>
1403 DATA A1,FE,00,C8,3E,85,32,33 <1EB4>
1404 DATA A2,ED,53,2D,A2,DD,7E,02 <1E3D>
1405 DATA 21,37,A2,86,32,2B,A2,DD <1E9F>
1406 DATA 7E,04,32,29,A2,DD,7E,06 <1EF2>
1407 DATA 32,27,A2,21,33,A2,CD,D4 <1E5B>
1408 DATA BC,D0,22,34,A2,79,32,36 <1E2A>
1409 DATA A2,1E,00,16,03,0E,41,21 <1E70>
1410 DATA 30,75,DF,34,A2,C9,84,3C <1ED7>
1411 DATA C0,07,40,06,03,CD,5B,A1 <1E00>
1412 DATA FE,00,C8,3E,01,12,13,1A <1E37>
1413 DATA 6F,13,1A,67,3E,1F,CD,5A <1EBF>
1414 DATA BB,DD,7E,04,CD,5A,BB,DD <1E84>
1415 DATA 7E,02,CD,5A,BB,CD,60,BB <1E1F>
1416 DATA 77,C9,3E,01,12,13,1A,6F <1E29>
1417 DATA 13,1A,67,CD,06,BB,77,C9 <1EB4>
1418 DATA 06,05,CD,5B,A1,FE,00,C8 <1EDD>
1419 DATA 3E,00,32,CF,A2,C3,8A,A2 <1E74>
1420 DATA 06,05,CD,5B,A1,FE,00,C8 <1EE1>
1421 DATA 3E,01,32,CF,A2,DD,6E,08 <1E2D>
1422 DATA DD,66,09,DD,7E,06,47,32 <1E8D>
1423 DATA CE,A2,DD,4E,04,DD,56,02 <1E24>
1424 DATA 3E,1F,CD,5A,BB,78,CD,5A <1E2D>
1425 DATA BB,7A,CD,5A,BB,3A,CF,A2 <1E54>
1426 DATA B7,C2,C7,A2,CD,60,BB,77 <1E5D>
1427 DATA 23,78,04,B9,C2,9D,A2,3A <1EDC>
1428 DATA CE,A2,47,7A,14,BB,C2,9D <1E19>
1429 DATA A2,C9,7E,CD,5A,BB,C3,B5 <1E8A>
1430 DATA A2,00,00,7B,FE,A4,DA,DA <1EC5>
1431 DATA A2,CD,63,A1,C9,C6,5C,26 <1ED1>
1432 DATA 00,6F,3E,0A,CD,68,BC,C9 <1EF4>
1433 DATA 7B,32,49,BE,C9,0E,10,D5 <1E79>
1434 DATA 7A,CD,28,A3,7B,CD,28,A3 <1EE1>
1435 DATA CD,47,A3,06,10,1A,CD,28 <1EAA>
1436 DATA A3,3E,20,CD,5A,BB,13,10 <1EC8>
1437 DATA F4,CD,47,A3,D1,06,10,1A <1E1B>
1438 DATA FE,20,DC,25,A3,FE,7F,D4 <1E4B>
1439 DATA 25,A3,CD,5A,BB,13,10,EF <1EA1>
1440 DATA 3E,00,0D,B9,C8,C3,EC,A2 <1E32>
1441 DATA 3E,2E,C9,6F,0F,0F,0F,0F <1EB0>
1442 DATA E6,0F,CD,3A,A3,7B,7D,E6 <1E2C>
1443 DATA 0F,CD,3A,A3,C9,FE,0A,DA <1EF7>
1444 DATA 41,A3,C6,07,C6,30,CD,5A <1E9D>
1445 DATA BB,C9,06,06,3E,20,CD,5A <1E72>
1446 DATA BB,10,FB,C9,06,02,CD,5B <1E8F>
1447 DATA A1,FE,00,C8,DD,66,03,DD <1E41>
1448 DATA 6E,02,22,B6,A3,D5,11,00 <1E01>
1449 DATA C0,06,01,1A,4F,13,1A,B9 <1E2F>
1450 DATA C2,8D,A3,3E,FF,B8,CA,8D <1EA0>
1451 DATA A3,04,13,3E,00,BA,C2,6B <1EA4>
1452 DATA A3,23,3E,00,77,ED,5B,B6 <1E16>
1453 DATA A3,ED,52,D1,CD,2B,A1,C9 <1E29>
1454 DATA 79,77,23,78,77,23,3E,01 <1E7F>
1455 DATA 47,1A,4F,C3,77,A3,62,6B <1EDC>
1456 DATA 11,00,C0,7E,4F,23,7E,47 <1EAA>
1457 DATA 23,FE,00,C8,79,12,13,05 <1E7C>
1458 DATA 78,FE,00,C2,A9,A3,C3,A0 <1E73>
1459 DATA A3,00,00,06,04,CD,5B,A1 <1E68>
1460 DATA FE,00,C8,ED,53,0C,A4,DD <1E35>
1461 DATA 6E,02,DD,66,03,22,0E,A4 <1EC0>
1462 DATA DD,6E,04,DD,66,05,22,10 <1E24>
1463 DATA A4,DD,5E,06,DD,56,07,1A <1E8F>
1464 DATA 47,13,1A,6F,13,1A,67,11 <1E02>
1465 DATA 00,C0,CD,77,BC,CD,7A,BC <1EB7>
1466 DATA 2A,6F,A7,ED,5B,0C,A4,CD <1EF7>
1467 DATA 2B,A1,2A,6D,A7,ED,5B,0E <1ED0>
1468 DATA A4,CD,2B,A1,2A,6A,A7,ED <1E04>
1469 DATA 5B,10,A4,CD,2B,A1,C9,00 <1EA2>
1470 DATA 00,00,00,00,00,06,02,CD <1E5B>
1471 DATA 5B,A1,FE,00,C8,3E,A1,C3 <1E89>
1472 DATA 29,A4,06,02,CD,5B,A1,FE <1ED6>
1473 DATA 00,C8,3E,9E,32,3B,A4,DD <1E1B>
1474 DATA 66,03,DD,6E,02,DD,56,01 <1EC5>
1475 DATA DD,5E,00,3E,10,CD,9E,BC <1E7E>
1476 DATA C9,3E,00,C3,45,A4,3E,01 <1EB3>
1477 DATA 32,BD,A4,21,70,01,22,BB <1E3E>
1478 DATA A4,7E,32,B9,A4,FE,00,C8 <1EE5>
1479 DATA 01,20,4E,11,03,00,19,3A <1E61>
1480 DATA BD,A4,FE,01,CA,A5,A4,09 <1E37>
1481 DATA E5,11,09,00,2A,BB,A4,19 <1E8D>
1482 DATA 09,54,5D,E1,ED,B8,01,04 <1E9C>
1483 DATA 00,2A,BB,A4,09,54,5D,21 <1E51>
1484 DATA B3,A4,01,06,00,ED,B0,2A <1EB2>
1485 DATA BB,A4,ED,5B,B9,A4,3A,BD <1EEB>
1486 DATA A4,FE,01,CA,9C,A4,7B,C6 <1E82>
1487 DATA 06,77,5F,19,C3,4B,A4,7B <1E58>
1488 DATA D6,06,77,5F,19,C3,4B,A4 <1ED0>
1489 DATA 23,54,5D,D5,11,06,00,19 <1E7C>
1490 DATA D1,ED,B0,C3,84,A4,01,00 <1E3B>
1491 DATA 00,2C,2C,2C,00,00,FA,01 <1E25>
1492 DATA 01,3E,01,32,D5,A5,C9,3E <1E1E>
1493 DATA 00,32,D5,A5,C9,06,07,CD <1E05>
1494 DATA 5B,A1,FE,00,C8,7B,32,D2 <1E59>
1495 DATA A5,DD,66,03,DD,6E,02,22 <1E9F>
1496 DATA CA,A5,22,CC,A5,DD,66,05 <1E10>
1497 DATA DD,6E,04,22,CE,A5,DD,7E <1E4D>
1498 DATA 06,32,D1,A5,DD,7E,08,32 <1EAB>
1499 DATA D0,A5,DD,7E,0C,32,D3,A5 <1E54>
1500 DATA 21,D0,A5,86,FE,51,D2,BF <1EA0>
1501 DATA A5,DD,7E,0A,21,D1,A5,86 <1E6A>
1502 DATA FE,C9,D2,BF,A5,21,00,C0 <1EC9>
1503 DATA 22,C5,A5,22,C3,A5,3E,01 <1ED9>
1504 DATA 32,C9,A5,DD,7E,0A,3D,F5 <1ED9>
1505 DATA CD,95,A5,F1,FE,00,C2,23 <1EDD>
1506 DATA A5,3A,D3,A5,5F,16,00,19 <1E24>
1507 DATA 22,C5,A5,2A,C3,A5,19,22 <1E83>
1508 DATA C3,A5,3A,D1,A5,4F,ED,5B <1EB3>
1509 DATA CE,A5,3A,D2,A5,FE,01,C2 <1E8F>
1510 DATA 53,A5,ED,5B,CA,A5,D5,CD <1E46>
1511 DATA 95,A5,D1,3A,D0,A5,47,3A <1E73>
1512 DATA D2,A5,FE,01,CA,90,A5,7E <1EFF>
1513 DATA E5,2A,CC,A5,77,23,22,CC <1E8E>
1514 DATA A5,E1,1A,FE,00,C2,7E,A5 <1E90>
```

```
1515 DATA 3A,D5,A5,FE,01,CA,7F,A5 <1E1C>
1516 DATA 1A,77,23,13,78,FE,00,05 <1EC4>
1517 DATA C2,5C,A5,79,FE,00,C8,0D <1ECE>
1518 DATA C3,53,A5,1A,77,C3,7F,A5 <1E20>
1519 DATA 3A,C9,A5,2A,C5,A5,FE,08 <1E4F>
1520 DATA C2,B3,A5,3E,01,32,C9,A5 <1EC8>
1521 DATA 11,50,00,2A,C3,A5,19,22 <1EAD>
1522 DATA C5,A5,22,C3,A5,C9,11,00 <1ED0>
1523 DATA 08,19,22,C5,A5,3C,32,C9 <1E97>
1524 DATA A5,C9,CD,63,A1,C9,00,00 <1E6D>
1525 DATA 00,00,00,00,00,00,00,00 <1E92>
1526 DATA 00,00,00,00,00,00,00,00 <1E94>
1527 DATA 00,16,00,21,FF,A5,01,04 <1E25>
1528 DATA 00,AF,F5,7A,77,09,F1,3C <1ECA>
1529 DATA FE,09,20,F6,CD,F3,A5,14 <1EAA>
1530 DATA 3E,28,BA,C8,18,E5,0E,C1 <1E14>
1531 DATA 21,FF,A5,DF,FC,A5,C9,52 <1E8A>
1532 DATA C6,07,00,00,C1,02,00,00 <1E3E>
1533 DATA C3,02,00,00,C5,02,00,00 <1EF7>
1534 DATA C7,02,00,00,C9,02,00,00 <1E9B>
1535 DATA C2,02,00,00,C4,02,00,00 <1E27>
1536 DATA C6,02,00,00,C8,02,21,29 <1E24>
1537 DATA 96,01,00,00,3E,04,CD,5A <1E9E>
1538 DATA BB,3E,02,CD,5A,BB,7D,FE <1EA8>
1539 DATA C9,C2,3D,A6,7C,FE,97,C8 <1E32>
1540 DATA 7E,FE,80,D2,4B,A6,CD,5A <1EEA>
1541 DATA BB,04,23,C3,33,A6,D6,80 <1ECD>
1542 DATA CD,5A,BB,3E,12,90,47,3E <1EE9>
1543 DATA 20,CD,5A,BB,10,FB,0C,79 <1E3E>
1544 DATA FE,03,CC,65,A6,C3,47,A6 <1E7C>
1545 DATA 0E,00,3E,0A,CD,5A,BB,3E <1E29>
1546 DATA 0D,CD,5A,BB,C9,00       <1877>
```

PINBALLMANIA

Ein rachsüchtiger Professor hat den jungen Pinball-Meister in die Vergangenheit verbannt. Helfen Sie Tiddy, ins Jahr 2111 zurückzukehren. Ein spannendes Flipper-Spiel für alle CPC-Typen.



In einer kleinen, irgendwo in Mitteleuropa gelegenen Grafschaft namens Zobelonien finden alljährlich die großen Pinball-Meisterschaften statt. So auch im Spätsommer des Jahres 2111. Professor Zorkowitsch ist seit 14 Jahren unbestrittener Rekordhalter.
Dieses Mal aber tritt ein junger Student namens Tiddy Tilt gegen ihn an und besiegt ihn schon in der zweiten Runde. Der Professor schwört wütend Rache.
Er konstruiert eine Zeitmaschine und koppelt diese mit dem Pinball-Automaten. Drei Monate später ist es soweit: Er sperrt Tiddy Tilt in seine Zeitmaschine und startet sie. Unter lautem Zischen verschwindet der Student in der Vergangenheit. Als Tiddy wieder erwacht, findet er in der Zeitmaschine einen Brief des Professors vor:
„Mein lieber junger Feind!
Da Ihr nun Pinball-Meister seid, könnt Ihr gleich unter Beweis stellen, ob Ihr diesen Titel auch wirklich verdient. Ihr befindet Euch in einer meiner Erfindungen, dem Zeitzylinder, mit

dem ich Euch in das Jahr Null zurückbefördert habe.
Die Tür des Zeitzylinders wird sich erst wieder öffnen, wenn Ihr es schafft, ins Jahr 2111 zurückzukommen.
Vor Euch steht ein Flipper. Rechts seht Ihr einige Kontrollanzeigen. Die erste von oben zeigt Euch das Jahr an, in dem Ihr Euch befindet. Darunter ist die Zeitkontrolle, die Euch anzeigt, in welches Jahr Ihr nach dem nächsten Zeitsprung gelangt. Diese ‚Warp'-Zeit ist mit dem Flipper verbunden. Je besser Ihr spielt, desto schneller kommt Ihr in der Zeit voran. Darunter seht Ihr noch die Anzeige für den Zeitmultiplikator.
Auf der linken Seite des Flippers habt Ihr, von oben nach unten betrachtet, zunächst die Anzeige für die Energie, die Ihr für die Magnetfelder zum Auffangen des Balles braucht, bevor dieser durch den rechten oder linken Ausgang verschwindet. Diese Taste aber nicht nur einmal drücken, sondern gedrückt halten!
Darunter seht Ihr die Energie, die Euch für den nächsten Zeitsprung zur Verfügung steht. Außerdem wird die Zahl der Bälle angezeigt.
Ganz unten findet Ihr den Multiplikatorbonus.
Und jetzt, mein lieber junger Feind . . . Gut Flipp und auf Nimmerwiedersehen!
Prof. Zorkowitsch"
Dieses Spiel lohnt die Tipparbeit wirklich, Sie werden nicht enttäuscht sein. Es läuft auf allen CPC-Typen mit Kassetten- oder Diskettenlaufwerk und bietet alles, was ein richtiger Flipper auch hat. Es gilt folgende Tastenbelegung:
Schläger links = Z-Taste;
Schläger rechts = Backslash-Taste, links neben der rechten Shift-Taste;
Magnetfeld an = eckige Klammer zu.

Abtipp-Hilfe

Tippen Sie zunächst das Pinball-Vorprogramm ab und speichern Sie es mit
SAVE "PIN"
Das Hauptprogramm legen Sie auf Ihrer Diskette oder Kassette mit dem Befehl
SAVE "PINBALL"
ab. Gestartet wird das Spiel mit
RUN "PIN"
Wir wünschen Ihnen viel Freude beim Flippern.

ME □

```
1 '*****************************  (24B1)
2 '*    PINBALL-VORPROGRAMM    *  (2440)
3 '*           VON             *  (24D0)
4 '*       ERNST KLASAREK      *  (2491)
5 '*           FUER            *  (2424)
6 '*    SCHNEIDER CPC-WELT     *  (24EF)
7 '*     CPC 464/664/6128   ME*  (2483)
8 '*****************************  (24BF)
10 RANDOMIZE INT(TIME/100):CALL &B
C6E:SYMBOL AFTER 64:FOR pk=(HIMEM+
1)TO(HIMEM+768):POKE pk,PEEK(pk)AN
D(PEEK(pk)*2)-2:NEXT                (53B6)
20 MODE 0:INK 0,0:BORDER 0:DEFINT
a-z:w=26:DEF FNr=INT(RND*(w-1+1)+1
):FOR i=1 TO 15:INK i,FNr,FNr:NEXT
:w=15:FOR s=0 TO 40:RESTORE:FOR i=
5 TO 16:READ a$:LOCATE i,8:PEN FNr
:PRINT a$:NEXT i,s:DATA P,I,N,B,A,
L,L,M,A,N,I,A                       (B74A)
30 SYMBOL AFTER 32:PEN FNr:LOCATE
6,11:PRINT"IS ";:PEN FNR:PRINT"LOA
DING":FOR s=0 TO 5000:NEXT:RUN"pin
ball"                               (49AE)
```

```
1 '*****************************  (24B1)
2 '*   PINBALL-HAUPTPROGRAMM   *  (2408)
3 '*           VON             *  (24D0)
4 '*       ERNST KLASAREK      *  (2491)
5 '*           FUER            *  (2424)
6 '*    SCHNEIDER CPC-WELT     *  (24EF)
7 '*     CPC 464/664/6128   ME*  (2483)
8 '*****************************  (24BF)
10 MODE 1:CALL &BC02:INK 0,6:PAPER
 3:BORDER 6:PEN 2                   (1AD5)
11 SYMBOL AFTER 47                  (098D)
12 FOR p=HIMEM+1 TO HIMEM+768:POKE
 p,PEEK(p)AND(PEEK(p)*2):NEXT       (3463)
13 LOCATE 15,5:PRINT"PINBALLMANIA"
                                    (1A98)
14 PEN 1:LOCATE 11,11:PRINT"TOUCHE
S:"                                 (1A66)
15 LOCATE 11,13:PRINT"(Z) = FLIPPE
R LEFT"                             (21C0)
16 LOCATE 11,14:PRINT"(\) = FLIPPE
R RIGHT"                            (221A)
17 LOCATE 11,15:PRINT"(]) = MAGNET
FIELDS ON"                          (2477)
18 DEFSTR a:DEFINT b-o,q-y:ON ERRO
R GOTO 552                          (19A0)
19 PEN 2:RESTORE 530:FOR i=97 TO 1
45:FOR s=0 TO 7:READ a:b(s)=VAL("&
"+a):NEXT                           (420A)
20 SYMBOL i,b(0),b(1),b(2),b(3),b(
4),b(5),b(6),b(7):NEXT              (4CDE)
21 DIM a(14),ce1(25,20),ce2(25,20)
,o1(25),o2(25)                      (3E73)
22 RESTORE 22:FOR i=0 TO 8:READ kt
(i):NEXT:DATA 1,3,5,1,1,1,1,1,1    (360E)
23 a(0)=CHR$(23)+CHR$(1):a(1)=CHR$
(23)+CHR$(0)                        (2E45)
24 a(2)="n":a(3)="o":a(4)="p":av=C
HR$(8):ay=CHR$(10):am=av+ay         (531D)
25 a(5)=CHR$(22)+CHR$(1):a(6)=CHR$
(22)+CHR$(0):ar=CHR$(141)           (3BC4)
26 a(7)=av+av+CHR$(11)+"sv"+av+am+
"tu":a(8)="l"+am+"m"                (4B2C)
27 a(9)="c"+am+"d":a(10)=CHR$(134)
:a(11)=CHR$(127)+CHR$(126)          (4127)
28 a(12)=CHR$(128)+CHR$(129)        (1B62)
29 a(13)=CHR$(135)+am+CHR$(136)+CH
R$(137)                             (28DF)
30 a(14)="f"+am+CHR$(138)+av+av+CH
R$(139)                             (318D)
31 alf=a(12)+ay+a(12)+CHR$(9)+CHR$
(9)+a(11)+CHR$(11)+a(11)            (485C)
32 add=" "+am+"A"+am+"D"+am+"D"     (2D46)
33 at=a(10)+a(10)+av+am+a(10)+a(10
):an=" ":ax="X 1000"                (53F7)
34 ak(1)="e"+am+"e"+am+"e"+am+"e"+
am+"e":ak(2)="e"+am+"e"+am+"e"      (5A7C)
35 ak(3)="g"+ay+"g"+ay+"g":ak(4)="
i"+am+ay+"{"                        (429F)
36 ak(5)="c"+am+"d"                 (1B9A)
37 aff1=CHR$(131)+CHR$(130)+ay+CHR
$(131)+CHR$(130)                    (2E68)
38 aff2=CHR$(132)+CHR$(133)+CHR$(1
1)+CHR$(132)+CHR$(133)              (2F00)
39 RESTORE 535:GOSUB 504:GOSUB 526
:PRINT a(5)                         (1C1B)
40 DEF FNr=INT(RND*(w-u+1)+u):DEF
FNk1=w1*COS(i):DEF FNk2=w2*SIN(i)   (53B2)
41 q=&BD19:kg(1)=4-lv:kg(2)=kg(1):
wt=0:rt(1)=0:rt(2)=0:bon=0:mt1=0:m
t2=0                                (69E2)
42 pt(1)=0:pt(2)=0:xt1=10:xt2=5:xt
3=5:pl=1:mg=119-(20*lv):ps1=0:ps2=
0:sa=0                              (6CA2)
43 mult=1:ml1=20:mlr=ml1:neu=1:pb(
1)=(1000000*lv):pb(2)=pb(1):bs(7)=
0                                   (64B8)
44 INK 2,20,6:LOCATE 15,15:PRINT"P
LEASE  WAIT":GOSUB 556:EVERY 60,1
GOSUB 551                           (3287)
45 PAPER 2:LOCATE 15,17:PRINT CHR$
(24)+")40 SECONDS("+CHR$(24)        (2C18)
46 GOSUB 160:INK 3,6,20:GOSUB 160:
SPEED INK 10,20:KEY DEF 66,0,0,0    (295C)
47 vn=40:na=0:st=-1:GOSUB 100:MODE
 1:PAPER 0                          (2935)
48 EVERY 80,0 GOSUB 549:LOCATE 26,
8:PRINT"o":PEN 1                    (1C99)
49 ENV 1,1,122,1,122,-1,10:ENV 2,1
,12,1,12,-1,5:ENV 3,1,12,1,120,-5,
1:ENV 4,17,-1,30:ENT 1,100,-20,5:E
NT 2,1,12,1,120,-1,3:ENT 3,11,10,1
,10,-12,1                           (79D1)
50 WINDOW#1,8,17,1,1:WINDOW#2,23,3
2,1,1:WINDOW#3,8,32,1,25            (2D30)
51 FOR i=1 TO 3:PEN#i,3:PAPER#i,2:
NEXT                                (224D)
52 RESTORE 536:READ x,y,ad:WHILE x
<>-1:LOCATE x,y:PRINT ad:READ x,y,
ad:WEND                             (48B7)
53 RESTORE 53:FOR i=1 TO 5:READ x,
y:LOCATE x,y:PRINT ak(i):NEXT:DATA
 13,4,11,6,28,18,10,22,32,10       (567F)
54 PEN 2:LOCATE 17,14:PRINT CHR$(2
46);:PEN 1:PRINT"0000";:PEN 2:PRIN
T CHR$(247):GOSUB 168               (33BA)
55 PEN 1:PRINT a(5):RESTORE 546:FO
R i=1 TO 4:READ x,y:PEN 2:LOCATE x
,y:PRINT at;:PEN 1:PRINT a(7):NEXT
                                    (4EDA)
56 RESTORE 56:FOR i=1 TO 5:READ x,
y,s:LOCATE x,y:PRINT a(s):NEXT:DAT
A 13,20,13,26,20,14,32,22,9,9,6,9,
28,21,8                             (668D)
57 LOCATE 28,6:PRINT"i";a(8):LOCAT
```

```
E 15,22:PRINT a1f:PEN 3:LOCATE 11,
5:PRINT"a"                          (348B)
58 PRINT a(6):PAPER 3:PEN 2:LOCATE
 12,19:PRINT"}":LOCATE 27,19:PRINT
"}"                                 (2BC0)
59 PAPER 0:PRINT a(5)               (10FA)
60 FOR i=0 TO 4 STEP 4              (10C4)
61 LOCATE 34,2+i:PRINT CHR$(144);S
TRING$(4,143);CHR$(142);            (28FC)
62 LOCATE 34,3+i:PRINT ar;"    ";a
r;                                  (2414)
63 LOCATE 34,4+i:PRINT CHR$(145);S
TRING$(4,143);CHR$(140);            (28DE)
64 NEXT                             (0643)
65 FOR i=2 TO 24                    (0FC5)
66 LOCATE 1,i:PRINT ar:LOCATE 7,i:
PRINT ar                            (2299)
67 LOCATE 33,i:PRINT ar:LOCATE 40,
i:PRINT ar;                         (25ED)
68 NEXT                             (064B)
69 RESTORE 538:FOR i=1 TO 20:READ
l1,l2,w,x,y,z                       (3526)
70 LOCATE l1,l2:PRINT CHR$(w);STRI
NG$(x,y);CHR$(z);                   (33D6)
71 NEXT                             (0651)
72 RESTORE 547:FOR i=1 TO 5:READ x
,y,z:LOCATE x,y:PRINT ar;STRING$(z
," ");ar:NEXT                       (4A89)
73 RESTORE 548:PEN 3:FOR i=1 TO 11
:READ x,y:LOCATE x,y:PRINT a(10):N
EXT:LOCATE 1,1:PRINT STRING$(40,20
8);:LOCATE 1,25:PRINT STRING$(40,2
10);:GOSUB 427                      (61D7)
74 FOR i=1 TO 25                    (0FBB)
75 LOCATE 1,i:PRINT CHR$(211):LOCA
TE 40,i:PRINT CHR$(209);            (260E)
76 NEXT:LOCATE 1,1:PRINT a(6)       (1498)
77 PEN 2:FOR i=11 TO 20:LOCATE 34,
i:PRINT ar;"    ";ar;:NEXT          (3380)
78 RESTORE 558:FOR i=0 TO 6:READ x
(i),y(i),ab(i):NEXT                 (3840)
79 LOCATE 26,7:PRINT"q":PEN 0:PAPE
R 2:LOCATE 35,11:PRINT"w01x"        (2433)
80 FOR i=12 TO 20:LOCATE 35,i:PRIN
T"wxwx":NEXT                        (2339)
81 PAPER 0:PEN 2:LOCATE 34,10:PRIN
T CHR$(144);STRING$(4,143);CHR$(14
2);                                 (2A26)
82 LOCATE 34,21:PRINT CHR$(145);ST
RING$(4,143);CHR$(140);             (24B5)
83 RESTORE 557:PEN 0:PAPER 3:FOR i
=1 TO 14:READ x,y,a:LOCATE x,y:PRI
NT a;:NEXT:LOCATE 2,19:PRINT add:I
F bs(7)=1 THEN RETURN               (5A9A)
84 RESTORE 84:PAPER 2:FOR i=12 TO
18:READ a:LOCATE 2,i:PRINT a:NEXT:
ORIGIN 302,282:RESTORE 540:DATA "}
",B,A,L,L,S,")"                     (5095)
85 DEG:WHILE z()1:READ z,x,y,w1,w2
,s:FOR i=x TO y STEP s:PLOT FNk1,F
Nk2,3:i=i+y+ABS(x):PLOT FNk1,FNk2:
i=i-(y+ABS(x)):NEXT:WEND:READ o1,o
2:IF o1=0 THEN 86 ELSE ORIGIN o1,o
2:z=0:GOTO 85                       (CEEA)
86 ORIGIN 1,1:RESTORE 541:x=0       (15AF)
87 READ x,y                         (0F8D)
88 IF x=0 THEN READ x,y:MOVE x,y:G
OTO 87                              (2760)
89 IF x=-1 THEN 92                  (1107)
90 DRAW x,y                         (0FF9)
91 GOTO 87                          (095F)
92 CLS#1:CLS#2:INK 1,24:INK 2,20:I
NK 3,6:z=REMAIN(0):BORDER 0:INK 0,
0                                   (3013)
93 PAPER 0:PEN 1:PRINT a(6):LOCATE
 31,6:PRINT a(8):LOCATE 30,21:PRIN
T" "+am+" "                         (3810)
94 LOCATE 31,20:PRINT" "+am+" ":LO
CATE 32,6:PRINT" ":RESTORE 94:PAPE
R 3:PEN 0:FOR i=6 TO 9:READ x:LOCA
TE 32,i:PRINT CHR$(x):NEXT:PEN 1:P
APER 0:DATA 84,73,77,69             (69F4)
95 MOVE 464,48:DRAW 494,62:MOVE 46
4,80:DRAW 494,94                    (2422)
96 MOVE 494,312:DRAW 494,222:DRAW
512,222:DRAW 512,326                (2649)
97 z=REMAIN(1):vn=0:na=40:st=1:GOS
UB 100:KEY DEF 66,1,252,252:SPEED
INK 1,1                             (40DB)
98 RANDOMIZE INT(TIME/100):PLOT 64
0,0,2:GOTO 135                      (1DD9)
99 :                                (06CA)
100 FOR i=vn TO na STEP st:OUT &BC
00,1:OUT &BD00,i:SOUND 7,i+10*10,6
,15,4,3:NEXT:RETURN                 (4891)
101 :                               (06CE)
102 PEN 2:PAPER 0:FOR i=vn TO na S
TEP st:LOCATE f,i:CALL q:PRINT"j":
GOSUB 150:LOCATE f,i:PRINT" ":NEXT
:RETURN                             (51A7)
103 :                               (06D2)
104 ORIGIN 1,1:TAG:FOR i=vn TO na
STEP st:MOVE f,i:CALL q:PRINT"j";:
MOVE f,i:CALL q:PRINT an;:NEXT:TAG
OFF:an=" ":RETURN                   (63F2)
105 :                               (06D6)
106 ORIGIN 1,1:LOCATE 1,1:PRINT a(
0):TAG                              (19E9)
107 FOR i=vn TO na STEP st:f=f+w1   (2CCA)
108 MOVE f,i:CALL q:PRINT"j";:MOVE
 f,i:CALL q:PRINT"j";:NEXT          (34DF)
109 TAGOFF:PRINT a(1):RETURN        (1169)
110 :                               (06E0)
111 PRINT a(0):ORIGIN o1(u),o2(u):
TAG:t=0                             (2F30)
112 WHILE ce1(u,t)()-1:MOVE ce1(u,
t),ce2(u,t):CALL q:PRINT"j";:MOVE
ce1(u,t),ce2(u,t):CALL q:PRINT"j";
:t=t+1:WEND:TAGOFF:PRINT a(1):ORIG
IN 1,1:RETURN                       (9C77)
113 :                               (06E6)
114 PEN 1:PAPER 3:LOCATE 34,9:PRIN
T"yWARPy"                           (1A8D)
115 PAPER 0:l1=26:l2=7:GOSUB 502:l
2=6:GOSUB 502:LOCATE 26,7:PRINT"q"
:PRINT a(5):PEN 1:LOCATE 26,6:PRIN
T"i";:PEN 2:PRINT av+"j":PRINT a(6
):SOUND 7,0,5,15,4,3,10             (7568)
116 w=100:u=70:m=FNr:t=-m*2:f=2:w=
10:u=1:PEN 2:EVERY 70,0 GOSUB 131   (4FF7)
117 FOR z=m TO 2 STEP-1:t=t+4:pt(p
l)=pt(pl)+100:GOSUB 466             (427C)
118 FOR i=2 TO 4                    (0E40)
119 LOCATE 26,8:PRINT a(i):FOR s=0
 TO t:NEXT                          (253D)
120 NEXT:SOUND 1,ABS(t)*2,2,15,4,3
                                    (1DC6)
121 LOCATE 26,8:PRINT a(3):FOR s=0
 TO t-50:NEXT                       (2547)
122 IF f=2 THEN f=3 ELSE f=2        (1BEA)
123 PEN f                           (0A1F)
124 NEXT z                          (0AA4)
125 z=REMAIN(0):GOSUB 173:sc=50:GO
SUB 459:mg=mg+20:GOSUB 170:PEN 2:L
OCATE 26,8:PRINT"p":PEN 1:kt(8)=1:
LOCATE 31,15:PRINT"g":GOSUB 550     (634B)
126 l1=26:l2=6:GOSUB 375:IF flag)6
 THEN 508                           (293D)
127 PEN 2:LOCATE 26,7:PRINT"j"      (12A9)
128 GOSUB 151                       (098B)
129 LOCATE 26,7:PRINT"q"            (0FE2)
130 RESTORE 130:GOSUB 284:LOCATE 2
6,8:PRINT"o":w1=56:st=2:GOTO 258:D
ATA 250,80,-14,390,-9               (43C9)
131 wt=wt+1:GOSUB 174               (17E4)
132 flag=FNr:LOCATE 26,7:IF flag)6
 AND x(40 THEN PEN 2:PRINT"y":SOUN
D 4,100,2,15,4,3,1 ELSE PEN 3:PRIN
T"q":SOUND 4,10,2,7                 (57B4)
133 LOCATE 26,8:PEN f:RETURN        (12B6)
134 :                               (0611)
135 u=1:w=5:vn=18:na=22:st=1:f=32:
GOSUB 102:bs(7)=1                   (44B9)
136 PAPER 0:LOCATE 32,22:PRINT"j"+
am;:PEN 1:PRINT"c":GOSUB 161        (27E5)
137 IF lo=4 THEN lo=0:GOSUB 329:da
=0:bon=0:GOSUB 151:GOSUB 167:mult=
1:GOSUB 372:GOSUB 151:GOTO 144      (4EFA)
138 IF neu=0 THEN d=6:GOSUB 559:GO
SUB 157:mult=mult+1:sc=20:GOSUB 46
0:sc=15:GOSUB 182:GOSUB 185:GOSUB
372:m=30:mg=mg+25:GOSUB 159:GOSUB
189:GOSUB 186:sc=5:GOSUB 467:GOTO
144                                 (8F2A)
139 PRINT a(1):GOSUB 403            (124C)
140 IF lo=2 OR lo=3 THEN IF mg)0 T
HEN GOSUB 405:GOSUB 406             (2887)
141 PEN 2:LOCATE 14,2:PRINT CHR$(2
4)+" PRESS ENTER "+CHR$(24)         (2CD4)
142 WHILE INKEY$()"":WEND           (0D99)
143 WHILE INKEY$()CHR$(13):WEND     (11B8)
144 PAPER 0:LOCATE 14,2:PRINT SPC(
13):GOSUB 153:GOSUB 414:GOSUB 154:
PEN 1:LOCATE 32,22:PRINT a(9):vn=2
1:na=14:st=-1:f=32:GOSUB 102        (5ED2)
145 l1=32:l2=13:GOSUB 374:GOSUB 16
9:LOCATE 17,16:PRINT SPC(6)         (2DDB)
146 GOSUB 159:IF kt(8)=1 THEN LOCA
TE 31,15:PRINT" ":GOSUB 154:kt(8)=
0:GOSUB 159                         (3639)
147 IF neu=1 THEN neu=0:mg=119-(20
*lv):GOSUB 173:GOSUB 170:GOSUB 176
:GOSUB 179:GOSUB 181:GOSUB 393:GOS
UB 154                              (4E07)
148 l1=32:l2=13:GOSUB 375:GOSUB 56
4                                   (20AB)
149 PEN 1:LOCATE 31,15:PRINT"g":kt
(8)=1:GOSUB 154:GOTO 190            (283C)
150 FOR s=0 TO 80:NEXT:RETURN       (13C2)
151 FOR s=0 TO 150:NEXT:RETURN      (1352)
152 FOR s=100 TO 1500 STEP 50:SOUN
D 7,s,10,7,,,INT(s/100):NEXT:RETUR
N                                   (3374)
153 FOR s=1500 TO 100 STEP-100:SOU
ND 7,s,3,5,,1,INT(s/100):NEXT:RETU
RN                                  (3421)
154 FOR s=300 TO 3000 STEP 300:SOU
ND 7,s,1,7,,,INT(s/100):NEXT:RETUR
N                                   (348F)
155 FOR s=1 TO 29:SOUND 7,0,18,15,
,,s:NEXT:SOUND 7,0,300,15,,,30:SOU
ND 7,0,150,7,,2,30:GOSUB 160:INK 0
,0:BORDER 0:RETURN                  (53B2)
156 FOR i=600 TO 1 STEP-1:SOUND 7,
i,1,15,2,2:CALL &BD20:CALL &BD21:C
ALL &BD22:CALL &BD23:CALL &BD24:CA
LL &BD25:NEXT:RETURN                (4649)
157 w=26:u=1:FOR s=0 TO 6:FOR i=1
TO 3:INK i,26:FOR loop=2 TO 30 STE
P 2:SOUND 7,loop,1,4,,,loop:NEXT l
oop:INK i,0:NEXT i:BORDER FNr,FNr:
NEXT s:BORDER 0:INK 1,24:INK 2,20:
INK 3,6:GOSUB 159:RETURN            (9EFD)
158 FOR s=2 TO 80:SOUND 7,s,3,15,4
,3:NEXT:RETURN                      (2426)
159 FOR s=0 TO 2000:NEXT:RETURN     (1425)
160 FOR s=0 TO 5000:NEXT:RETURN     (14B2)
161 SOUND 7,0,150,7,1,,1:RETURN     (1589)
162 SOUND 7,0,8,12,4,3,20:RETURN    (177E)
163 FOR s=100 TO 5 STEP-5:SOUND 7,
s,1,7:NEXT:RETURN                   (220A)
164 FOR s=10 TO 50:SOUND 7,s*5,10,
7,3,1:NEXT:RETURN                   (27DC)
165 FOR i=0 TO 4:FOR s=80 TO 10 ST
EP-10:SOUND 7,s,2,7,3,1:NEXT s,i:R
ETURN                               (3B8E)
166 FOR i=0 TO 4:FOR s=60 TO 10 ST
EP-10:SOUND 7,s,2,7,3,2:NEXT s,i:R
ETURN                               (3B39)
167 w=10:u=1:FOR s=10 TO 300 STEP
10:SOUND 7,s*FNr,3,5:NEXT:RETURN    (39A7)
168 IF xt1)99 THEN RETURN ELSE PEN
 2:LOCATE 24,4:PRINT USING"##";xt1
:RETURN                             (2B48)
169 anz="yREALy":x=34:y=5:GOSUB 18
7:PEN 1:LOCATE 35,3:PRINT USING"##
##";rt(pl):RETURN                   (490D)
170 anz="yMGTy":x=2:y=7:GOSUB 187   (2615)
171 IF mg)999 THEN mg=999           (1925)
172 PEN 1:LOCATE 3,5:PRINT USING"#
##";mg:RETURN                       (1C83)
173 anz="yWARPy":x=34:y=9:GOSUB 18
7                                   (285F)
174 PEN 1:IF wt)2111 THEN wt=2111:
PEN 3                               (1FFA)
175 LOCATE 35,7:PRINT USING"####";
wt:RETURN                           (1B77)
176 anz="yERGy":x=2:y=11:GOSUB 187
```

```
                                                    (27CB)
177 IF eg)999 THEN PEN 3:LOCATE 3,
9:PRINT"E99":RETURN                                 (2027)
178 PEN 1:LOCATE 3,9:PRINT USING"#
##";eg:RETURN                                       (1C6C)
179 anz=" 1y ":x=3:y=15:GOSUB 187:
PEN 1:LOCATE 4,13:PRINT USING"##";
kg(1)                                               (3F4A)
180 IF sp=1 THEN RETURN ELSE anz="
 2y ":x=3:y=19:GOSUB 187:PEN 1:LOC
ATE 4,17:PRINT USING"##";kg(2):RET
URN                                                 (4D2F)
181 anz=add:x=2:y=19:GOSUB 167:PEN
 1:LOCATE 4,21:PRINT USING"##";da:
RETURN                                              (3EE7)
182 IF da=99 THEN RETURN ELSE PAPE
R 3:PEN 1:LOCATE 2,19:PRINT add                     (2591)
183 FOR i=1 TO sc:da=da+1:SOUND 7,
100,10,7,,2:IF da)99 THEN da=99                     (4173)
184 LOCATE 4,21:PRINT USING"##";da
:NEXT:GOSUB 181:GOSUB 151:GOSUB 15
1:PAPER 0:RETURN                                    (2D4E)
185 BORDER 3,6:PEN 1:PAPER 3:LOCAT
E 34,22:PRINT"yMULTy";:SOUND 7,30,
70,7,1:GOSUB 151:GOSUB 151:PAPER 0
:PEN 2:BORDER 0:RETURN                              (439B)
186 PEN 0:PAPER 3:LOCATE 34,22:PRI
NT"yMULTy";:PAPER 0:PEN 2:RETURN                    (2416)
187 PEN 2:SOUND 4,f+100,15,6,2,2:S
OUND 1,f*7,70,6,2,2:FOR s=0 TO 12:
PAPER 1:LOCATE x,y:PRINT anz:PAPER
 3:LOCATE x,y:PRINT anz:NEXT:PEN 0
:PAPER 3:LOCATE x,y:PRINT anz:PAPE
R 0:PEN 2:GOSUB 151:RETURN                          (8D33)
188 GOSUB 173:GOSUB 170:GOSUB 176:
RETURN                                              (15D7)
189 eg=eg+m:wt=wt+m:GOTO 188                        (2B90)
190 :                                               (0681)
191 l1=29:l2=18:GOSUB 322:w=10:u=1
:ORIGIN 1,1                                         (2F2F)
192 ON FNr GOTO 193,194,195,196,19
7,198,196,197,198,198                               (33B7)
193 v=17:GOSUB 111:GOTO 233                         (162B)
194 v=18:GOSUB 111:GOTO 239                         (166D)
195 v=19:GOSUB 111:f=240:GOTO 256                   (1E74)
196 RESTORE 196:GOTO 199:DATA 140,
252,12,440,-10                                      (234D)
197 RESTORE 197:GOTO 199:DATA 140,
260,11,440,-11                                      (232B)
198 RESTORE 198:DATA 140,266,11,44
0,-13                                               (1E07)
199 READ vn,na,st,f,w1:GOSUB 106                    (27CC)
200 :                                               (0695)
201 IF w1()-10 THEN 207 ELSE l1=19
:l2=8:GOSUB 322:w=6:u=1:ORIGIN 1,1
                                                    (3DC3)
202 ON FNr GOTO 203,204,205,206,20
3,203                                               (23E0)
203 vn=254:na=120:st=-12:f=308:w1=
11:GOSUB 106:GOTO 191                               (3CAD)
204 vn=254:na=106:st=-14:f=308:w1=
10:GOSUB 106:lo=3:GOTO 484                          (4477)
205 vn=254:na=106:st=-16:f=308:w1=
12:GOSUB 106:lo=4:GOTO 484                          (4411)
206 vn=254:na=90:st=-20:f=308:w1=8
:GOSUB 106:GOTO 249                                 (3B66)
207 IF w1()-11 THEN 213                             (133E)
208 l1=19:l2=8:GOSUB 322:w=20                       (2280)
209 IF FNr=2 THEN 211 ELSE w=2:ON
FNr GOTO 210,212                                    (28DC)
210 vn=254:na=70:st=-12:f=294:w1=3
:GOSUB 106:GOTO 285                                 (3B42)
211 vn=250:na=40:st=-18:f=296:an="
j":lo=0:PRINT a(0):GOSUB 104:GOTO
323                                                 (4E04)
212 vn=254:na=70:st=-12:f=294:w1=-
3:GOSUB 106:f=256:GOTO 256                          (450C)
213 w=10:u=1:l1=19:l2=8:GOSUB 322                   (2905)
214 ON FNr GOTO 215,216,217,218,21
9,220,215,217,215,217                               (3396)
215 vn=254:na=200:st=-9:f=296:w1=-
17:GOSUB 106:GOTO 222                               (3C50)
216 vn=254:na=136:st=-10:f=296:w1=
-13:GOSUB 106:GOTO 475                              (3D55)
217 vn=254:na=130:st=-11:f=296:w1=
-11:GOSUB 106:GOTO 233                              (3DB6)
218 vn=254:na=96:st=-15:f=296:w1=-
8:GOSUB 106:GOTO 239                                (3C02)
219 vn=254:na=106:st=-11:f=296:w1=
-9:GOSUB 106:lo=1:GOTO 484                          (449B)
220 vn=254:na=106:st=-11:f=296:w1=
-9:GOSUB 106:lo=2:GOTO 484                          (441E)
221 :                                               (06BF)
222 l1=9:l2=13:GOSUB 322:w=10:u=1:
ORIGIN 1,1                                          (2E8E)
223 ON FNr GOTO 224,225,226,230,22
8,229,231,227,231,227                               (3319)
224 vn=186:na=90:st=-15:f=154:w1=3
0:GOSUB 106:GOTO 249                                (3B33)
225 vn=186:na=106:st=-6:f=154:w1=1
8:GOSUB 106:lo=3:GOTO 484                           (42BE)
226 vn=186:na=110:st=-6:f=154:w1=1
9:GOSUB 106:lo=4:GOTO 484                           (4240)
227 vn=190:na=120:st=-7:f=154:w1=2
8:GOSUB 106:GOTO 191                                (3AD0)
228 vn=190:na=220:st=4:f=154:w1=28
:GOSUB 106:ks=3:GOTO 445                            (415D)
229 vn=196:na=230:st=6:f=154:w1=31
:GOSUB 106:ks=2:GOTO 445                            (41D9)
230 vn=196:na=244:st=5:f=154:w1=20
:GOSUB 106:ks=1:GOTO 445                            (41ED)
231 vn=210:na=270:st=13:f=154:w1=1
9:GOSUB 106:w1=-13:GOTO 213                         (4553)
232 :                                               (06D5)
233 l1=9:l2=18:GOSUB 322:w=8:u=1:O
RIGIN 1,1                                           (2D06)
234 ON FNr GOTO 235,236,237,238,23
5,236,235,236                                       (2B51)
235 vn=150:na=270:st=14:f=154:w1=1
2:GOSUB 106:w1=-13:GOTO 213                         (4576)
236 vn=150:na=260:st=14:f=154:w1=1
4:GOSUB 106:w1=-11:GOTO 207                         (45AB)
237 vn=134:na=280:st=14:f=158:w1=1
6:GOSUB 106:GOTO 355                                (3BC1)
238 vn=134:na=270:st=14:f=154:w1=2
4:GOSUB 106:GOTO 114                                (3B4E)
239 :                                               (06E3)
240 x=13:y=20:s=4:z=13:GOSUB 248:w
=10:u=1                                             (37B0)
241 IF FNr)5 THEN w=5:ON FNr GOTO
243,244,245,246,247                                 (2F70)
242 w=10:v=0:GOSUB 111:GOTO 249                     (1DE0)
243 v=1:GOSUB 111:GOTO 191                          (1586)
244 v=2:GOSUB 111:lo=3:GOTO 484                     (1DCA)
245 v=3:GOSUB 111:lo=4:GOTO 484                     (1D35)
246 v=4:GOSUB 111:f=360:GOTO 286                    (1E4C)
247 v=5:GOSUB 111:f=340:GOTO 286                    (1E72)
248 PEN 3:LOCATE x,y:PRINT a(z):BO
RDER FNr:sc=1:GOSUB 458:BORDER 0:S
OUND s,2000,60,7,3:PEN 1:LOCATE x,
y:PRINT a(z):RETURN                                 (6219)
249 x=26:y=20:s=1:z=14:GOSUB 248:w
=10:u=1                                             (37B3)
250 IF FNr)5 THEN w=4:ON FNr GOTO
252,253,254,255                                     (2B23)
251 w=10:v=6:GOSUB 111:GOTO 240                     (1D2C)
252 v=7:GOSUB 111:lo=1:GOTO 484                     (1D0A)
253 v=8:GOSUB 111:lo=2:GOTO 484                     (1D9F)
254 v=9:GOSUB 111:f=240:GOTO 257                    (1D1B)
255 v=10:GOSUB 111:GOTO 233                         (1638)
256 :                                               (0605)
257 ORIGIN 1,1:w=3:u=2:st=FNr:w1=7
2:IF f)250 THEN w1=62                               (3EED)
258 TAGOFF:LOCATE 1,1:PRINT a(0):T
AG:FOR i=f TO 286 STEP st:w1=w1-1:
MOVE i,w1:PRINT"j";:MOVE i,w1:PRIN
T"j";:IF INKEY(71)=0 THEN 260                       (6B41)
259 SOUND 4,w1,1,4:NEXT:TAGOFF:GOT
O 323                                               (1AB2)
260 TAGOFF:LOCATE 1,1:PRINT a(1):P
EN 1:LOCATE 15,22:PRINT aft1:GOSUB
 162:LOCATE 15,22:PRINT alt                         (3B11)
261 w=6:u=1                                         (1274)
262 IF i(260 THEN ON FNr GOTO 263,
264,265,266,267,264 ELSE 268                        (32AC)
263 RESTORE 263:GOSUB 284:GOTO 348
:DATA 80,280,14,230,0                               (253B)
264 RESTORE 264:GOSUB 284:GOTO 361
:DATA 80,270,14,240,1                               (2580)
265 RESTORE 265:GOSUB 284:w1=-11:G
OTO 207:DATA 80,270,14,240,4                        (2F5E)
266 RESTORE 266:GOSUB 284:GOTO 355
:DATA 80,270,14,240,6                               (25D7)
267 RESTORE 267:GOSUB 284:GOTO 347
:DATA 80,270,14,240,8                               (255B)
268 IF i(270 THEN w=9:ON FNr GOTO
269,270,272,271,272,273,274,274,26
9 ELSE 275                                          (457B)
269 RESTORE 269:GOSUB 284:GOTO 361
:DATA 80,270,14,254,1                               (2594)
270 RESTORE 270:GOSUB 284:w1=-13:G
OTO 213:DATA 80,270,14,254,2                        (2FB2)
271 RESTORE 271:GOSUB 284:w1=-10:G
OTO 201:DATA 80,270,14,254,4                        (2FEE)
272 RESTORE 272:GOSUB 284:GOTO 355
:DATA 80,270,14,254,5                               (25F9)
273 RESTORE 273:GOSUB 284:GOTO 347
:DATA 80,270,14,254,7                               (25A3)
274 RESTORE 274:GOSUB 284:GOTO 114
:DATA 80,270,14,254,10                              (2637)
275 w=10:ON FNr GOTO 277,282,276,2
78,278,276,279,280,281,282                          (3BB6)
276 RESTORE 276:GOSUB 284:w1=-13:G
OTO 213:DATA 70,260,14,270,1                        (2F69)
277 RESTORE 277:GOSUB 284:w1=-11:G
OTO 207:DATA 70,260,14,270,2                        (2F85)
278 RESTORE 278:GOSUB 284:GOTO 347
:DATA 70,270,14,270,6                               (252B)
279 RESTORE 279:GOSUB 284:ks=1:GOT
O 445:DATA 70,250,15,270,12                         (2EA1)
280 RESTORE 280:GOSUB 284:ks=2:GOT
O 445:DATA 70,234,14,270,14                         (2E8C)
281 RESTORE 281:GOSUB 284:ks=3:GOT
O 445:DATA 70,224,14,270,16                         (2E85)
282 RESTORE 282:GOSUB 284:GOTO 380
:DATA 70,180,14,270,22                              (26A5)
283 :                                               (063B)
284 READ vn,na,st,f,w1:GOSUB 105:R
ETURN                                               (2941)
285 :                                               (063F)
286 ORIGIN 1,1:w=3:u=2:st=FNr:w1=6
4:IF f)345 THEN w1=74                               (3F7A)
287 TAGOFF:LOCATE 1,1:PRINT a(0):T
AG:FOR i=f TO 306 STEP-st:w1=w1-1:
MOVE i,w1:PRINT"j";:MOVE i,w1:PRIN
T"j";:IF INKEY(22)=0 THEN 289                       (6CEB)
288 SOUND 1,w1,1,4:NEXT:TAGOFF:GOT
O 323                                               (1A84)
289 TAGOFF:LOCATE 1,1:PRINT a(1):P
EN 1:LOCATE 21,23:PRINT aftr:GOSUB
 162:LOCATE 15,22:PRINT alt                         (3BB0)
290 w=8:u=1                                         (12B2)
291 IF i)340 THEN ON FNr GOTO 292,
293,295,294,295,296,297,295 ELSE 2
98                                                  (3ABD)
292 RESTORE 292:GOSUB 320:GOTO 347
:DATA 80,280,18,360,0                               (25D4)
293 RESTORE 293:GOSUB 320:GOTO 355
:DATA 80,260,14,346,-1                              (2687)
294 RESTORE 294:GOSUB 320:w1=-13:G
OTO 213:DATA 80,270,14,346,-5                       (30D2)
295 RESTORE 295:GOSUB 320:GOTO 361
:DATA 80,280,14,346,-6                              (26A4)
296 RESTORE 296:GOSUB 320:GOTO 348
:DATA 80,280,14,346,-8                              (266F)
297 RESTORE 297:GOSUB 320:GOTO 416
:DATA 80,270,14,346,-11                             (2777)
298 w=11:u=1                                        (13C7)
299 IF i)326 THEN ON FNr GOTO 300,
301,302,303,304,305,307,307,300,30
6,306 ELSE 308                                      (4626)
300 RESTORE 300:GOSUB 320:GOTO 355
:DATA 70,260,14,332,0                               (253B)
301 RESTORE 301:GOSUB 320:w1=-10:G
OTO 201:DATA 70,260,14,332,-2                       (3040)
302 RESTORE 302:GOSUB 320:w1=-11:G
OTO 207:DATA 70,260,14,332,-3                       (3064)
303 RESTORE 303:GOSUB 320:w1=-13:G
OTO 213:DATA 70,260,14,332,-4                       (301A)
304 RESTORE 304:GOSUB 320:GOTO 361
:DATA 70,280,14,332,-5                              (2631)
305 RESTORE 305:GOSUB 320:GOTO 348
:DATA 70,280,14,332,-7                              (2689)
306 RESTORE 306:GOSUB 320:GOTO 416
:DATA 70,270,14,332,-9                              (2695)
307 RESTORE 307:GOSUB 320:GOTO 437
:DATA 70,270,14,332,-11                             (2714)
```

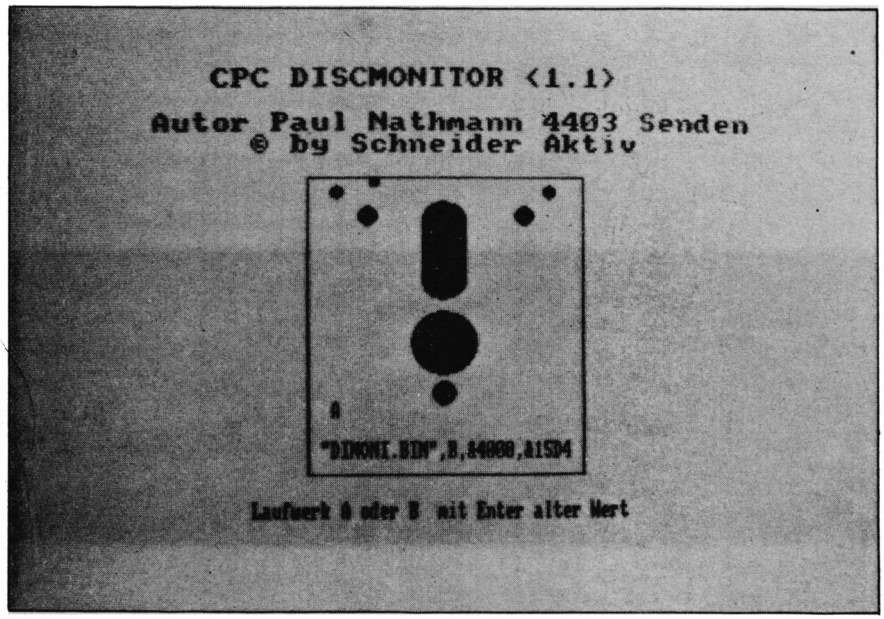
```
308 w=15:u=1                                    (13E3)
309 ON FNr GOTO 310,318,311,312,31
3,314,315,316,317,312,318,314,315,
314,316                                         (47B2)
310 RESTORE 310:GOSUB 320:w1=-10:G
OTO 201:DATA 70,260,14,320,0                    (2F32)
311 RESTORE 311:GOSUB 320:w1=-13:G
OTO 213:DATA 70,260,14,320,-3                   (30E5)
312 RESTORE 312:GOSUB 320:GOTO 361
:DATA 70,280,14,320,-4                          (26C5)
313 RESTORE 313:GOSUB 320:GOTO 348
:DATA 70,280,14,320,-6                          (260B)
314 RESTORE 314:GOSUB 320:GOTO 416
:DATA 70,270,14,320,-8                          (266B)
315 RESTORE 315:GOSUB 320:GOTO 437
:DATA 70,262,14,320,-10                         (27CF)
316 RESTORE 316:GOSUB 320:GOTO 377
:DATA 70,232,14,320,-14                         (275C)
317 RESTORE 317:GOSUB 320:GOTO 221
:DATA 70,196,14,320,-16                         (2756)
318 RESTORE 318:GOSUB 320:GOTO 469
:DATA 70,190,14,320,-20                         (27FB)
319 :                                           (0683)
320 READ vn,na,st,f,w1:GOSUB 106:R
ETURN                                           (29A9)
321 :                                           (0687)
322 PRINT a(5):PEN 3:LOCATE 11,12:
PRINT at;:PEN 2:PRINT a(7):SOUND 4
,f+100,7,15,4,3:SOUND 1,f+10,3,12,
4,1,3:sc=10:GOSUB 465:PEN 2:LOCATE
 11,12:PRINT at;:PEN 1:PRINT a(7):
PRINT a(6):RETURN                               (98CA)
323 :                                           (068B)
324 LOCATE 1,1:PRINT a(1):TAG:FOR
i=50 TO 22 STEP-2:SOUND 7,(1000-(i
*10)),10,15,4,3:MOVE 296,i:CALL q:
PRINT"j";:NEXT:MOVE 296,i:PRINT" "
;:TAGOFF:lo=0:GOSUB 158                          (771A)
325 GOSUB 159:IF sa=1 THEN neu=2:l
o=0:GOSUB 413:sa=0:GOTO 135 ELSE k
g(pl)=kg(pl)-1:xt1=10:xt2=5:xt3=5:
GOSUB 179:mt1=0:mt2=0:GOSUB 329:GO
SUB 168:GOSUB 83:GOSUB 563:GOTO 33
5                                               (9E9B)
326 mult=1:mll=20:ps1=0:ps2=0:mlr=
20:RESTORE 22:FOR i=0 TO 8:READ kt
(i):NEXT                                        (528E)
327 PAPER 0:PEN 1:RESTORE 545:READ
 11,12,ad:WHILE 11<>-1:GOSUB 154:L
OCATE 11,12:PRINT ad:GOSUB 151:REA
D 11,12,ad:WEND:LOCATE 13,4:PRINT
ak(1):LOCATE 11,6:PRINT ak(2):LOCA
TE 28,10:PRINT ak(3)                            (9089)
328 PEN 0:PAPER 2:FOR s=20 TO 12 S
TEP-1:LOCATE 35,s:PRINT"wxwx":NEXT
:LOCATE 35,11:PRINT"w01x":wt=rt(pl
):PAPER 0:bon=0:da=0:eg=0:GOTO 135
                                                (6F67)
329 IF da=0 THEN 333 ELSE IF pl=1
THEN zb=2 ELSE zb=1                             (2CE7)
330 GOSUB 159:PAPER#zb,3:PEN#zb,2:
CLS#zb:anz=add:x=2:y=19:da=da*100:
FOR loop=1 TO mult:SOUND 4,2*loop,
20,15,4,3:LOCATE#zb,1,1:PRINT#zb,l
oop;av+"X";USING"######";da*loop;:
GOSUB 187:GOSUB 151:NEXT:GOSUB 159
:GOSUB 163:CLS#zb                               (D888)
331 z=da*mult:FOR p=1 TO mult:SOUN
D 7,p*20,25,15,4,3:z=z-da:pt(pl)=p
t(pl)+da:GOSUB 466:PRINT#zb,z:FOR
s=0 TO 1000:NEXT s,p:PAPER#zb,2:PE
N#zb,3:CLS#zb:GOSUB 159:da=0:GOSUB
 181                                            (C3B6)
332 IF pt(zb)>0 THEN LOCATE#zb,2,1
:PRINT#zb,USING"########";pt(zb);               (42C8)
333 PAPER 3:PEN 0:LOCATE 2,19:PRIN
T add:PAPER 0:IF bon=0 THEN RETURN
                                                (2705)
334 GOSUB 151:sc=1:GOSUB 159:GOSUB
 427:PEN 1:FOR n=bon TO 1 STEP-1:L
OCATE 18,14:PRINT USING"####";n:GO
SUB 461:SOUND 7,n+10,1,7:FOR s=0 T
O INT(n/2)STEP 2:NEXT s,n:LOCATE 1
8,14:PRINT"0000":RETURN                         (8A21)
335 IF pt(pl)>=pb(pl)THEN pb(pl)=p
b(pl)+1000000:kg(pl)=kg(pl)+1:INK
0,0,20:BORDER 20,0:GOSUB 156:GOSUB
 155:GOSUB 167:GOSUB 408:GOSUB 159
:GOSUB 179:INK 1,24:INK 3,6:INK 0,
0:BORDER 0:GOTO 335                             (A0EC)
336 neu=1:IF sp=2 THEN 338                      (1AAD)
337 IF kg(1)<1 THEN GOSUB 407:GOSU
B 396:GOTO 345 ELSE 326                         (24CF)
338 IF pl=1 AND kg(1)<1 THEN GOSUB
 407:GOSUB 396                                  (2208)
339 IF pl=2 AND kg(2)<1 THEN GOSUB
 407:GOSUB 396                                  (22FE)
340 IF kg(1)<1 AND kg(2)<1 THEN 34
5                                               (1F16)
341 IF pl=1 AND kg(2)>0 THEN pl=2:
GOTO 326                                        (2555)
342 IF pl=1 AND kg(2)<1 THEN 326                (1C20)
343 IF pl=2 AND kg(1)>0 THEN pl=1:
GOTO 326                                        (25BA)
344 IF pl=2 AND kg(1)<1 THEN 326                (1C14)
345 PAPER 3:CLS:PEN 2:LOCATE 16,13
:PRINT"GAME OVER":GOSUB 158:GOSUB
152:GOSUB 160:PRINT STRING$(25,10)
:GOSUB 161:RUN                                  (4140)
346 :                                           (06B9)
347 v=24:GOTO 349                               (1137)
348 v=25                                        (0C9F)
349 GOSUB 549:LOCATE 1,1:PRINT a(0
):TAG:ORIGIN o1(v),o2(v):t=0:u=1:w
=7                                              (4750)
350 WHILE ce1(v,t)<>-1:MOVE ce1(v,
t),ce2(v,t):CALL q:PRINT"j";:SOUND
 FNr,ce1(v,t)+100,1,15,4,3:MOVE ce
1(v,t),ce2(v,t):PRINT"j";:t=t+1:WE
ND:TAGOFF:PRINT a(1)                            (B4B2)
351 ORIGIN 1,1:BORDER 0:sc=xt1:xt1
=xt1+2:TAGOFF:PRINT a(1):GOSUB 168
:GOSUB 465                                      (3E32)
352 IF v=24 THEN vn=260:na=80:st=-
14:f=226:w1=2:GOSUB 105:GOTO 256                (44ED)
353 vn=260:na=80:st=-14:f=360:w1=-
2:GOSUB 105:GOTO 285                            (3DBC)
354 :                                           (06C9)
355 IF kt(4)=1 THEN kt(4)=0:LOCATE
 22,8:GOSUB 359:GOTO 358                        (2B2D)
356 v=20:GOSUB 111:11=20:12=6:GOSU
B 374:d=2:GOSUB 559:GOSUB 185:GOSU
B 165:GOSUB 369:GOSUB 360                       (47F1)
357 GOSUB 154:kt(3)=1:PEN 1:LOCATE
 17,8:PRINT"e":11=20:12=6:GOSUB 37
5:v=22:GOSUB 110                                (4573)
358 RESTORE 358:GOSUB 284:GOTO 256
:DATA 270,80,-14,330,-5                         (27A3)
359 PRINT" ":GOSUB 162:sc=1:GOSUB
461:RETURN                                      (1DAD)
360 GOSUB 186:sc=xt1:GOSUB 459:mg=
mg+5:m=10:GOSUB 189:RETURN                      (3803)
361 IF kt(3)=1 THEN kt(3)=0:LOCATE
 17,8:GOSUB 359:GOTO 364                        (2B36)
362 v=21:GOSUB 111:11=19:12=6:GOSU
B 374:d=1:GOSUB 559:GOSUB 185:GOSU
B 165:GOSUB 366:GOSUB 360                       (478B)
363 GOSUB 154:kt(4)=1:PEN 1:LOCATE
 22,8:PRINT"e":11=19:12=6:GOSUB 37
5:v=23:GOSUB 110                                (4546)
364 RESTORE 364:GOSUB 320:GOTO 285
:DATA 270,70,-14,254,5                          (261D)
365 :                                           (06DF)
366 PAPER 2:PEN 3:IF ps1=0 THEN ps
1=1:LOCATE 35,mll:PRINT"w":PAPER 0
:PEN 2:RETURN                                   (3597)
367 mt1=mt1+1:mult=mult+1:IF mt1=1
0 THEN mll=20:GOTO 368 ELSE IF mt1
>10 THEN ps1=0:mll=20:mult=mult-1:
RETURN                                          (738F)
368 GOSUB 372:ps1=0:LOCATE 36,mll:
PRINT"x":PAPER 0:PEN 2:mll=mll-1:R
ETURN                                           (3AEF)
369 PAPER 2:PEN 3:IF ps2=0 THEN ps
2=1:LOCATE 38,mlr:PRINT"x":RETURN               (2FEC)
370 mt2=mt2+1:mult=mult+1:IF mt2=1
0 THEN mlr=20:GOTO 371 ELSE IF mt2
>10 THEN ps2=0:mlr=20:mult=mult-1:
RETURN                                          (7372)
371 GOSUB 372:ps2=0:LOCATE 37,mlr:
PRINT"w":PAPER 0:PEN 2:mlr=mlr-1:R
ETURN                                           (3A9D)
372 PAPER 2:PEN 3:INK 3,6,26:LOCAT
E 36,11:PRINT USING"##";mult:SOUND
 7,50,100,7,1,3:GOSUB 159:INK 3,6:
RETURN                                          (4309)
373 :                                           (06EF)
374 PRINT a(5):PEN 1:LOCATE 11,12:
PRINT"i";:PEN 2:PRINT av+"j":PRINT
 a(6):GOSUB 161:GOSUB 153:GOSUB 15
9:RETURN                                        (4B86)
375 SOUND 7,0,5,7,,,20:PEN 1:LOCAT
E 11,12:PRINT"i":GOSUB 154:PEN 2:R
ETURN                                           (317F)
376 :                                           (06F5)
377 IF kt(0)=1 THEN kt(0)=0:11=10:
12=11:GOSUB 502:SOUND 7,0,5,7,,,2:
sc=200:GOSUB 163:GOSUB 465:11=10:1
2=12:GOSUB 502:GOTO 222                         (6E39)
378 11=10:12=11:GOSUB 502:11=9:12=
9:GOSUB 381:INK 0,6,0:GOSUB 156:GO
SUB 152:INK 0,0:GOSUB 161:mg=mg+15
:m=50:GOSUB 397:IF sc=0 AND kt(2)<
2 THEN sa=1:GOSUB 413 ELSE kg(pl)=
kg(pl)+1:GOSUB 401:GOSUB 179                    (AFCE)
379 sc=200:GOSUB 459:GOSUB 427:sc=
35:GOSUB 404:GOSUB 467:GOSUB 189:G
OSUB 550:sc=10:GOSUB 182:mult=mult
+2:GOSUB 372:neu=2:lo=0:GOTO 135                (6F14)
380 IF kt(8)=1 THEN kt(8)=0:SOUND
7,0,5,7,,,2:sc=100:11=31:12=15:GOS
UB 502:GOSUB 465:GOSUB 564:GOTO 19
1 ELSE 11=31:12=15:GOSUB 502:11=32
:FOR 12=15 TO 17:GOSUB 502:NEXT:ne
u=0:lo=0:GOTO 135                               (A280)
381 SOUND 7,0,5,7,,,10:TAGOFF:FOR
s=0 TO 100:PEN 1:LOCATE 11,12:PRIN
T"(":SOUND 7,s+1,2,15,4,3:PEN 2:LO
CATE 11,12:PRINT"j":NEXT:PEN 1:LOC
ATE 11,12:PRINT"(":GOSUB 167:RETUR
N                                               (7A9C)
382 :                                           (0602)
383 w=3:u=1:s=311:x=372:y=x:EVERY
10,0 GOSUB 389:FOR i=311 TO 414:OR
IGIN 208,364,s,i,x,y:CLG FNr:s=s-1
:NEXT:z=REMAIN(0):ORIGIN 208,364,2
08,414,380,364:PLOT 300,0,2                     (A254)
384 IF flag=0 THEN 390                          (1311)
385 CLG 3:FOR s=1 TO flag                       (179C)
386 TAG:FOR i=160 TO-LEN(anz)*13 S
TEP-10:SOUND 7,ABS(i+200),2,15,4,3
:MOVE i,14:PRINT anz;" ";:NEXT i                (56D4)
387 TAG:FOR i=-LEN(anz)*13 TO 160
STEP 10:SOUND 7,ABS(i+200),2,15,4,
3:MOVE i,14:PRINT" ";anz;:NEXT i                (55D0)
388 NEXT s:TAGOFF:GOSUB 392:ORIGIN
 1,1,1,639,1,399:flag=0:GOSUB 414:
RETURN                                          (33C7)
389 x=x+1:y=y-1:RETURN                          (1ED2)
390 CLG 3:TAG:FOR i=200 TO-LEN(anz
)*16 STEP-20:MOVE i,14:PRINT anz;"
 ";:NEXT                                        (3E00)
391 TAGOFF:GOSUB 392:ORIGIN 1,1,1,
639,1,399:flag=0:GOSUB 414:RETURN               (2D49)
392 s=372:FOR i=372 TO 380:ORIGIN
208,364,208,414,s,i:CLG 0:s=s-1:SO
UND 7,s,5,10,4,3:NEXT:RETURN                    (58D4)
393 anz="NOW YOU ARE IN YEAR"+STR$
(rt(pl)):GOSUB 151:INK 3,6:flag=0:
GOTO 382                                        (4B0A)
394 anz="TIME-BONUS":GOSUB 163:GOS
UB 151:GOSUB 151:flag=1:GOTO 382                (36DB)
395 anz="BUT THIS IS NOT THE PRESE
NT TIME!":flag=0:GOTO 382                       (3E3F)
396 anz="YOU ARE LOST IN THE TIME!
":GOSUB 164:flag=0:GOTO 382                     (3B6D)
397 anz="VERY GOOD!":flag=1:GOTO 3
82                                              (2735)
398 GOSUB 176:GOSUB 176:anz="SORRY
,YOU HAVE NOT ENOUGH ENERGY FOR TH
E TIME-WARP!":GOSUB 158:flag=0:GOT
O 382                                           (5F3E)
399 anz="!CONGRATULATION!":flag=2:
GOSUB 382                                       (2D6B)
400 anz="NOW YOU ARE IN THE PRESEN
T TIME!":flag=0:GOTO 382                        (3D4A)
401 anz="!EXTRABALL!":GOSUB 163:fl
ag=2:GOTO 382                                   (2DD2)
```

```
402 anz="!TIME-WARP!":flag=2:GOTO
382                                    (28AD)
403 anz="PLAYER NO."+STR$(pl):flag
=0:GOTO 382                            (3173)
404 anz="+"+STR$(sc*1000)+" BONUS"
:GOSUB 163:flag=1:GOTO 382             (3AE4)
405 anz="DON'T FORGET THE MAGNETFI
ELDS!":flag=0:lo=4:GOTO 382            (432E)
406 anz="PRESS (J) IN THE RIGHT MO
MENT TO CATCH THE BALL!":flag=0:GO
TO 382                                 (4DCE)
407 anz="PLAYER NO."+STR$(pl)+"
      ) GAME OVER (":FLAG=0:GOTO
 382                                   (4C5A)
408 anz="!!BALL!!":INK 3,4:INK 1,2
0,15:GOSUB 159:flag=3:GOTO 382         (3823)
409 anz="LEAVE THE TIME-CYLINDER!"
:flag=0:GOTO 382                       (3553)
410 anz="GOOD BYE":flag=2:GOTO 382
                                       (25A4)
411 anz="TIMEWARP-GATE OPEN":GOSUB
 163:flag=0:GOTO 382                   (34D6)
412 anz="SUPERBONUS-GATE OPEN":GOS
UB 163:flag=0:GOTO 382                 (361D)
413 anz="SHOOT AGAIN!":flag=2:GOTO
 382                                   (298E)
414 RESTORE 414:w=3:u=1:TAG:FOR i=
218 TO 408 STEP 16:READ z:PLOT 640
,0,FNr:MOVE i,380:PRINT CHR$(z);:N
EXT:TAGOFF:PLOT 640,0,2:RETURN:DAT
A 80,73,78,66,65,76,76,77,65,78,73
,65                                    (864B)
415 :                                  (0644)
416 vn=270:st=10:f=192:t=2:flag=0:
IF kt(1)=0 THEN 417 ELSE ON kt(1)G
OTO 428,429,430,431,432                (658A)
417 na=340:GOSUB 104:v=11:GOSUB 11
1:11=9:12=5:GOSUB 502:LOCATE 9,6:P
RINT"j":PEN 1:LOCATE 9,7:PRINT"c":
GOSUB 152                              (5196)
418 GOSUB 157:sc=xt2:GOSUB 460:mg=
mg+40:eg=eg+45:wt=wt+30:GOSUB 188:
xt2=xt2+5:sc=10:GOSUB 182:sc=20:GO
SUB 404:GOSUB 467:GOSUB 151            (8303)
419 IF eg<wt-rt(pl)THEN GOSUB 398:
GOTO 424                               (2899)
420 INK 1,10:INK 2,1:INK 3,11:GOSU
B 161:GOSUB 402                        (1F56)
421 w=26:u=1:INK 0,0,26:BORDER 0,2
6:FOR i=310 TO 2 STEP-2:SOUND 1,i,
3,13,4,3:SOUND 4,i,3,7,,,INT(i/10)
:INK 1,FNr:INK 2,FNr:INK 3,FNr:NEX
T:INK 1,26,0:GOSUB 155                 (85B7)
422 BORDER 0:INK 0,0:GOSUB 158:INK
 1,24:eg=eg-(wt-rt(pl)):rt(pl)=wt:
GOSUB 176:INK 2,20:GOSUB 169:GOSUB
 393:INK 3,6:IF rt(pl)<>2111 THEN
GOSUB 395                              (7C94)
423 flag=1:IF rt(pl)=2111 THEN INK
 3,4:INK 2,20:GOSUB 156:GOSUB 399:
GOSUB 409:GOSUB 410:GOSUB 161:FOR
i=1 TO 25:SOUND 7,3000-(i*25),100,
7,3:LOCATE#3,1,25:PRINT#3,CHR$(10)
:NEXT:GOSUB 161:LOCATE 17,12:PRINT
"THE END":GOSUB 160:GOSUB 160:RUN      (A002)
424 GOSUB 160:GOSUB 550:LOCATE 9,6
:PRINT ak(5):11=9:12=5:GOSUB 502:v
=12:GOSUB 111:na=270:vn=340:st=-10
:f=192:GOSUB 104:vn=na:na=80:w1=2:
GOSUB 105                              (8CD6)
425 IF flag=1 THEN kt(2)=5:SOUND 7
,0,5,7,,,20:PEN 1:LOCATE 13,4:PRIN
T ak(1):PEN 2                          (3E8F)
426 GOTO 256                           (094E)
427 PEN 3:LOCATE 17,16:PRINT ax:RE
TURN                                   (17D3)
428 na=340:GOSUB 104:SOUND 7,0,5,7
,,,2:GOSUB 163:GOTO 434                (2AE6)
429 na=330:GOTO 433                    (1329)
430 na=310:GOTO 433                    (130F)
431 na=300:GOTO 433                    (130F)
432 na=280                             (0E48)
433 GOSUB 104:SOUND 7,0,8,7,,,2        (16FE)
434 sc=30-kt(t):GOSUB 465:kt(t)=kt
(t)-1:IF kt(t)<0 THEN kt(t)=0          (551A)
435 st=-10:vn=na:na=270:GOSUB 104:
IF t=1 THEN 443                        (35C3)
436 vn=na:na=270:st=-10:f=192:GOSU
B 104:vn=na:na=80:w1=2:GOSUB 105:G
OTO 256                                (58A1)
437 vn=270:st=10:f=160:t=1:IF kt(t
)=0 THEN 438 ELSE ON kt(t)GOTO 440
,441,442                               (53A4)
438 na=310:GOSUB 104:SOUND 7,100,5
,15,4,3,2:TAGOFF:PEN 3:LOCATE 11,5
:PRINT"b";:PEN 2:LOCATE 11,6:PRINT
"j":sc=xt3:GOSUB 462:xt3=xt3+1         (6597)
439 d=0:GOSUB 559:INK 1,20,0:SOUND
 7,10,40,7,1:FOR s=500 TO 20 STEP-
20:SOUND 7,s,2,15,4,3:NEXT:sc=2:GO
SUB 467:PEN 3:LOCATE 11,5:PRINT"a"
+am+" ":INK 1,24:vn=na:na=270:st=-
10:GOSUB 104:GOTO 443                  (9E7A)
440 na=310:GOSUB 104:SOUND 7,0,5,7
,,,2:GOSUB 163:GOTO 434                (2AEF)
441 na=300:GOTO 433                    (1323)
442 na=280:GOTO 433                    (1311)
443 vn=270:na=80:w1=4:GOSUB 105:GO
TO 256                                 (2981)
444 PEN 2:LOCATE x,y:PRINT"j":GOSU
B 150:LOCATE x,y:PRINT" ":RETURN       (2E2C)
445 TAGOFF:SOUND 4,100,8,15,4,3,1      (178B)
446 IF ks<>1 THEN 449                  (1151)
447 x=28:y=10:GOSUB 444:IF kt(5)=1
 THEN kt(5)=0:d=3:GOSUB 559:GOSUB
454                                    (415E)
448 vn=244:na=80:st=-8:w1=-3:f=430
:GOSUB 105:GOTO 286                    (3BBA)
449 IF ks<>2 THEN 452                  (11AF)
450 x=29:y=11:GOSUB 444:IF kt(6)=1
 THEN kt(6)=0:d=4:GOSUB 559:GOSUB
454                                    (41AA)
451 vn=224:na=80:st=-8:w1=-4:f=440
:GOSUB 105:GOTO 286                    (3BB3)
452 x=30:y=12:GOSUB 444:IF kt(7)=1
 THEN kt(7)=0:d=5:GOSUB 559:GOSUB
454                                    (411A)
453 vn=210:na=80:st=-8:w1=-5:f=450
:GOSUB 105:GOTO 286                    (3BAC)
454 SOUND 7,100,15,15,2,3:sc=2:GOS
UB 461:FOR i=5 TO 7:IF kt(i)=1 THE
N RETURN ELSE NEXT                     (3F4A)
455 INK 2,0:INK 3,3,6:INK 1,20,0:G
OSUB 549:GOSUB 167:GOSUB 161:INK 2
,14:GOSUB 394:wt=wt+40:GOSUB 173:s
c=5:GOSUB 467                          (53A5)
456 BORDER 0:sc=INT(wt/10):GOSUB 4
59:eg=eg+45:GOSUB 176:mg=mg+30:GOS
UB 170:INK 3,6:INK 1,24:INK 2,20:k
t(5)=1:kt(6)=1:kt(7)=1:PEN 1:PAPER
 0:LOCATE 28,10:PRINT ak(3):SOUND
7,0,8,15,4,3,20:RETURN                 (A356)
457 :                                  (0698)
458 sum=10:GOTO 463                    (1335)
459 sum=100:GOTO 463                   (1364)
460 sum=500:GOTO 463                   (14B5)
461 sum=1000:GOTO 463                  (14CE)
462 sum=5000                           (0FED)
463 TAGOFF:FOR i=1 TO sc:pt(pl)=pt
(pl)+sum:SOUND 1,100+(i*2),3,12,4,
3:GOSUB 466:NEXT:RETURN                (5638)
464 :                                  (06A6)
465 TAGOFF:pt(pl)=pt(pl)+(100*sc)      (2B76)
466 LOCATE#pl,2,1:PRINT#pl,USING"#
#######";pt(pl):RETURN                 (333D)
467 FOR i=1 TO sc:bon=bon+1:PEN 3:
LOCATE 17,14:PRINT STRING$(6,207):
PEN 2:SOUND 7,bon+2,3,15,4,3:LOCAT
E 17,14:PRINT CHR$(246);:PEN 1:PRI
NT USING"####";bon;:PEN 2:PRINT CH
R$(247):NEXT:PEN 1:RETURN              (84EA)
468 :                                  (06AE)
469 w=10:u=1:ORIGIN 1,1:11=9:12=13
:GOSUB 322:IF FNr>2 THEN 473           (3B8A)
470 w=2:u=1:ON FNr GOTO 471,472        (21CE)
471 vn=174:na=80:st=-12:w1=14:f=13
6:GOSUB 106:GOTO 256                   (3B95)
472 TAGOFF:PEN 2:11=9:12=15:GOSUB
502:GOSUB 481:IF FNr=1 THEN 482 EL
SE PEN 2:LOCATE 9,15:PRINT"j":GOSU
B 151:LOCATE 9,15:PRINT" ":GOTO 46
9                                      (55F0)
473 TAG:FOR i=174 TO 134 STEP-30:M
OVE 136,i:CALL q:PRINT"j";:MOVE 13
6,i:CALL q:PRINT" ";:NEXT:TAGOFF       (4473)
474 :                                  (06BA)
475 w=10:u=1:ORIGIN 1,1:11=9:12=18
:GOSUB 322:IF FNr>2 THEN 479           (3B17)
476 w=2:u=1:ON FNr GOTO 477,478        (2104)
477 v=13:GOSUB 111:f=350:GOTO 286      (1F9E)
478 TAGOFF:PEN 2:11=9:12=17:GOSUB
502:GOSUB 481:IF FNr=1 THEN 482 EL
SE PEN 2:LOCATE 9,17:PRINT"j":GOSU
B 151:LOCATE 9,17:PRINT" ":GOTO 47
5                                      (552B)
479 TAG:FOR i=144 TO 184 STEP 30:M
OVE 136,i:CALL q:PRINT"j";:MOVE 13
6,i:CALL q:PRINT" ";:NEXT:TAGOFF:G
OTO 469                                (48FD)
480 :                                  (06C6)
481 11=8:12=16:GOSUB 374:GOSUB 153
:sc=10:GOSUB 460:mg=mg+5:m=15:GOSU
B 189:sc=5:GOSUB 182:GOSUB 550:sc=
1:GOSUB 467:11=8:12=16:GOSUB 375:R
ETURN                                  (7FD7)
482 vn=150:na=100:st=-5:w1=20:f=13
6:GOSUB 106:GOTO 249                   (3A21)
483 :                                  (06CC)
484 ORIGIN 1,1:LOCATE 1,1:PRINT a(
0):IF lo<3 THEN f=170:w1=190:flag=
1:GOTO 486                             (40CC)
485 f=410:w1=420:flag=0                (216B)
486 WHILE INKEY(19)=0 AND mg>0         (16E3)
487 TAG:MOVE f,110:PRINT"j";:MOVE
f,110:CALL q:PRINT"j";                 (2A17)
488 IF flag=1 THEN f=f-10:IF f<w1-
30 THEN flag=0                         (35D3)
489 IF flag=0 THEN f=f+10:IF f>w1
THEN flag=1                            (321F)
490 TAGOFF:LOCATE 1,1:PRINT a(1):m
g=mg-1:PEN 1:LOCATE 3,5:PRINT USIN
G"###";mg:SOUND 7,mg+f-100,5,5,3,1
:PRINT a(0)                            (5A18)
491 WEND:TAGOFF:PRINT a(1):IF mg=0
 THEN SOUND 7,100,70,7,3               (27E3)
492 IF lo<3 THEN IF f>170 THEN 11=
14:12=20:GOSUB 502:GOTO 240 ELSE 4
94                                     (378B)
493 IF f<410 THEN 11=25:12=20:GOSU
B 502:GOTO 249                         (2AF4)
494 IF lo<3 THEN 11=11:12=21:GOSUB
 502:GOTO 495 ELSE 498                 (2EA7)
495 :                                  (06E4)
496 IF lo=1 THEN 11=10:12=21:GOSUB
 502:12=22:GOSUB 374:sc=5:GOSUB 46
0:GOSUB 182:11=10:12=22:GOSUB 375:
12=23:GOSUB 502:12=24:GOSUB 381:mg
=mg+5:m=10:GOSUB 189:lo=0:neu=2:GO
TO 135                                 (ABA8)
497 IF lo=2 THEN 11=12:12=21:GOSUB
 502:12=22:GOSUB 374:sc=10:GOSUB 4
60:GOSUB 182:11=12:12=22:GOSUB 375
:11=13:12=23:GOSUB 502:vn=232:na=3
00:st=2:GOTO 503                       (93DD)
498 LOCATE 28,20:PRINT"j":GOSUB 15
1:LOCATE 28,20:PRINT" "                (21FC)
499 IF lo=3 THEN 11=27:12=21:GOSUB
 502:12=22:GOSUB 374:sc=10:GOSUB 4
60:GOSUB 182:11=27:12=22:GOSUB 375
:11=26:12=23:GOSUB 502:vn=378:na=2
92:st=-2:GOTO 503                      (9583)
500 11=29:12=21:GOSUB 502:12=22:GO
SUB 374:sc=5:GOSUB 460:GOSUB 182:1
1=29:12=22:GOSUB 375:11=30:12=21:G
OSUB 502:11=31:GOSUB 502:11=32:GOS
UB 502:neu=2:GOTO 136                  (9362)
501 :                                  (06F0)
502 PEN 2:LOCATE 11,12:PRINT"j":GO
SUB 151:LOCATE 11,12:PRINT" ":RETU
RN                                     (320F)
503 TAG:FOR i=vn TO na STEP st:SOU
ND 7,INT(i/3),1,12,4,3:MOVE i,30:P
RINT"j";:NEXT:MOVE i,30:PRINT" ";:
TAGOFF:GOSUB 164:GOTO 325              (6188)
504 PLOT 640,0,2:11=30:TAG             (1809)
505 READ a$:x=VAL("&"+a$):IF x=&FF
 THEN TAGOFF:RETURN                    (2A13)
506 MOVE 11,280:PRINT CHR$(x);:11=
```

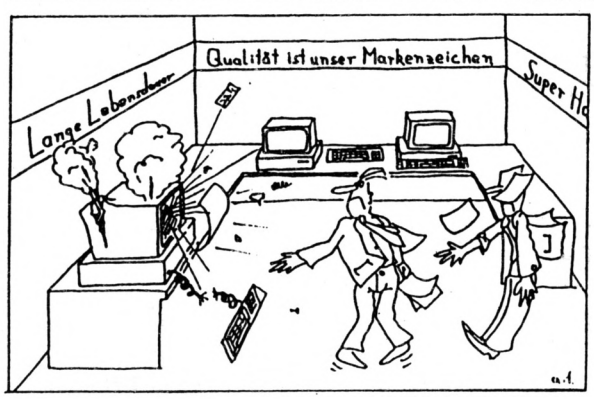
```
11+14:GOTO 505                                    (2EB3)
507 :                                             (06FC)
508 v=14:GOSUB 111:GOSUB 427                      (166C)
509 11=29:12=3:GOSUB 374:PEN 2:LOC
ATE 26,8:PRINT"o":LOCATE 26,7:PRIN
T"q":PEN 1:kt(8)=1:LOCATE 31,15:PR
INT"g":GOSUB 154:sc=3:GOSUB 467:w=
6:u=1:sc=10:GOSUB 460:IF FNr>3 AND
 wt<>2111 THEN 516 ELSE w=3:ON FNr
 GOTO 510,513,513                                 (ADD4)
510 PEN 2:LOCATE 28,7:PRINT"y":GOS
UB 166:GOSUB 375:11=28:GOSUB 522                  (2A19)
511 PAPER 0:IF kt(1)>0 THEN FOR i=
8 TO 6 STEP-1:GOSUB 151:LOCATE 11,
i:PRINT" ":GOSUB 154:NEXT:kt(1)=0:
GOSUB 412                                         (4A8B)
512 11=28:12=6:GOSUB 375:12=7:GOSU
B 502:11=29:12=8:GOSUB 502:GOTO 52
3                                                 (4278)
513 PEN 2:LOCATE 30,7:PRINT"y":GOS
UB 166:GOSUB 375:11=30:GOSUB 522                  (2AFC)
514 PAPER 0:IF kt(2)>0 THEN FOR i=
8 TO 4 STEP-1:GOSUB 151:LOCATE 13,
i:PRINT" ":GOSUB 154:NEXT:kt(2)=0:
GOSUB 411                                         (4A35)
515 11=30:12=6:GOSUB 375:12=7:GOSU
B 502:12=8:GOSUB 502:GOTO 523                     (39F4)
516 GOSUB 375:v=15:GOSUB 111:11=32
:PAPER 3:RESTORE 94:FOR 12=6 TO 9:
READ x:LOCATE 11,12:PRINT"j":GOSUB
 150:PEN 1:LOCATE 11,12:PRINT CHR$
(x):PEN 2:NEXT:PAPER 0                            (71E8)
517 LOCATE 11,12:PRINT"j";:PEN 1:P
RINT am+"c":GOSUB 152:GOSUB 165                   (2F83)
518 sc=xt2:GOSUB 460:eg=eg+40:GOSU
B 176:PAPER 3:LOCATE 14,2:PRINT CH
R$(24)+" PRESS SPACE "+CHR$(24):GO
SUB 167:w=wt+50:u=wt+20                           (733A)
519 WHILE INKEY(47)<>0:RESTORE 94:
FOR s=6 TO 9:READ x:LOCATE 32,s:PR
INT CHR$(x):SOUND 7,500,2,5,,2:PEN
 0:LOCATE 32,s:PRINT CHR$(x):PEN 1
:NEXT s:wt=FNr:GOSUB 175:WEND:GOSU
B 166:GOSUB 173:PAPER 0:LOCATE 14,
2:PRINT SPC(13):GOSUB 414                         (956F)
520 GOSUB 154:PEN 1:LOCATE 32,10:P
RINT ak(5):11=32:RESTORE 520:PAPER
 3:FOR 12=9 TO 6 STEP-1:PEN 2:LOCA
TE 11,12:PRINT"j":PEN 0:READ x:LOC
ATE 11,12:PRINT CHR$(x):NEXT:PEN 1
:DATA 69,77,73,84                                 (84F1)
521 v=16:GOSUB 111:PAPER 0:GOTO 50
9                                                 (19DA)
522 12=4:GOSUB 502:12=5:GOSUB 502:
12=6:GOSUB 374:GOSUB 185:mult=mult
+1:GOSUB 372:sc=20:GOSUB 460:GOSUB
 186:RETURN                                       (5CD0)
523 :                                             (061C)
524 11=30:12=9:GOSUB 381:GOSUB 167
:GOSUB 159:neu=2:lo=0:GOTO 135                    (3A4C)
525 :                                             (0620)
526 a="":LOCATE 13,18:PRINT"(1 OR
2) PLAYERS ":GOSUB 528:sp=VAL(a):I
F sp<1 OR sp>2 THEN 526                           (514B)
527 CLS:a="":LOCATE 7,18:PRINT"LEV
EL 1-3 (1=EASY/3=STRONG)":GOSUB 52
8:lv=VAL(a):IF lv<1 OR lv>3 THEN 5
27 ELSE CLS:RETURN                                (617C)
528 LOCATE 20,20:PRINT CHR$(7)+"_"
:WHILE INKEY$<>"":WEND:a="":WHILE
a="":a=INKEY$:WEND:LOCATE 20,20:PR
INT UPPER$(a):GOSUB 159:RETURN                    (52DA)
529 :                                             (0628)
530 DATA 0,0,0,18,18,7E,FF,7E,0,0,
0,7E,FF,7E,18,18,99,FF,3C,18,18,18
,18,18,18,18,18,18,18,18,18,18,0,0
,0,0,0,FF,81,FF,0,3,5,B,15,29,51,A
1,20,70,D8,6C,36,1B,E,4,4,E,1B,36,
6C,D8,70,20,3C,5A,C3,99,81,E7,66,3
C,0,0,3C,6E,76,7E,3C,0,0,0,3C,66,5
A,66,3C,0                                         (F521)
531 DATA 0,0,18,24,24,24,24,24,24,
24,24,24,24,18,0,0,0,0,0,0,FF,0,0,0,
0,0,0,7E,C3,7E,0,0,0,0,7E,56,C3,6A
,7E,0,0,0,7E,3C,18,0,0,0,0,0,0,0,0
,0,0,0,0,F,30,40,C0,87,88,88,88,88
,88,88,87,C0,60,30,F,11,11,11,E1,3
,6,1C,F0,F0,1C,6                                  (DA56)
532 DATA 3,E1,11,11,11,1E,3E,7E,7E
,7E,7E,3E,1E,78,7C,7E,7E,7E,7E,7C,
78,0,18,3C,7E,0,0,0,0,0,18,30,70,7
0,30,10,0,3C,66,C3,99,99,C3,66,3C,
80,F0,48,24,12,A,6,3,7E,81,66,DB,D
B,66,81,7E,F,39,63,CE,18,70,C0,0,0
,0,0,3,E,18,13,1E,F0,9C                           (E177)
533 DATA C6,73,18,E,3,0,0,0,0,C0,7
0,18,C8,78,FE,3,3,FE,0,0,0,0,F,10,
10,F,0,0,0,0,7F,C0,C0,7F,0,0,0,0,F
0,8,8,F0,0,0,0,0,0,0,18,3C,3C,18,0
,0,0,C0,A0,D0,A8,94,8A,85,84,82,85
,88,90,A0,C0,FF,80,40,A0,50,28,14,
A,FE,21,41,A1,11,9,5,3,FF                         (E3B5)
534 DATA 1,2,5,A,14,28,50,7F,BD,3D
,DD,ED,F5,F9,1,FF,BD,BD,BD,A5,A5,B
D,BD,BD,FE,1,F9,F5,ED,DD,3D,BD,FF,
0,FF,E7,E7,FF,0,FF,7F,80,9F,AF,B7,
BB,BC,BD,BD,BC,BB,B7,AF,9F,80,7F                  (A6A7)
535 DATA 47,41,4D,45,20,44,45,53,4
9,47,4E,20,26,20,50,52,4F,47,52,41
,4D,4D,49,4E,47,20,42,59,20,45,2E,
4A,2E,4B,4C,41,53,41,52,45,4B,FF                  (84C9)
536 DATA 19,6,ii,17,8,e,22,8,e,18,
4,k,21,4,k,26,5,"/",9,9,f,32,13,i,
8,16,i,10,11,h,27,22,i i,26,6,i,12
,22,i,31,15,g,30,9,f                              (789B)
537 DATA 30,6,i,29,3,i,15,7,k,24,7
,k,35,3,TIME,35,7,TIME,9,4,k,-1,-1
,""                                               (452D)
538 DATA 1,1,144,5,143,142,33,1,14
4,6,143,142,1,25,145,5,143,140,33,
25,145,6,143,140,2,2,144,3,143,142
,2,3,145,3,143,140,2,23,144,3,143,
142,2,24,145,3,143,140,2,4,144,3,1
43,142,2,6,145,3,143,140,2,8,144,3
,143,142                                          (D262)
539 DATA 2,10,145,3,143,140,3,12,1
44,2,143,142,3,14,145,2,143,140,3,
16,144,2,143,142,3,18,145,2,143,14
0,3,20,144,2,143,142,3,22,145,2,14
3,140,34,23,144,4,143,142,34,24,14
5,4,143,140                                       (B30E)
540 DATA 0,-20,90,56,52,2,0,-16,90
,86,78,1,0,-16,90,54,44,2,1,-30,94
,22,20,3,456,322,1,-6,90,58,48,2,1
68,338,0,-10,95,18,12,4,1,-4,92,42
,40,2,0,0                                         (8FC0)
541 DATA ,,220,260,210,286,210,338
,,,186,338,186,286,190,258,,,126,3
38,126,286,144,286,144,328,150,336
,,,190,258,180,272,180,336,154,336
,154,268,168,232,164,232,144,278,1
20,278,142,206,,,398,320,398,278,3
82,258,,,416,320,424,312,424,292,4
90,226                                            (F2AC)
542 DATA 490,312,504,316,,,490,214
,430,274,410,248,416,278,416,318,,
,490,214,470,178,474,174,494,208,5
14,208,514,30,494,30,494,154,474,1
24,,,128,178,110,160,110,136,126,1
20,126,80,142,60,142,10,160,10,160
,50,168,46,180,20,278,10,278,4                    (E809)
543 DATA 324,4,324,10,464,20,464,9
4,,,104,396,104,1,518,1,518,396,10
2,396,102,2,520,2,520,398,104,398,
,,108,394,108,382,206,382,206,362,
416,362,416,382,512,382,512,394,,,
280,392,280,386,340,386,340,392,28
0,392,,,514,208,514,30,,,512,208,5
12,30                                             (F1AA)
544 DATA ,,250,194,352,194,352,174
,250,174,250,194,,,246,198,356,198
,356,170,246,170,246,198,,,366,352
,366,334,400,334,400,352,366,352,-
1,-1                                              (8A58)
545 DATA 10,11,h,17,8,e,22,8,e,31,
15,g,-1,-1                                        (2A53)
546 DATA 19,8,9,13,29,18,9,18                     (1BFA)
547 DATA 2,5,3,2,9,3,3,13,2,3,17,2
,3,21,2                                           (27B0)
548 DATA 11,11,11,22,16,23,23,23,2
7,5,29,5,31,5,8,17,19,7,20,7,29,23                (42FB)
549 mk=w:w=25:BORDER FNr,FNr:w=mk:
RETURN                                            (31D6)
550 kt(0)=1:GOSUB 154:PAPER 0:PEN
1:LOCATE 10,11:PRINT"h":RETURN                    (28F4)
551 SOUND 7,80,3,5:RETURN                         (10F4)
552 RESUME NEXT                                   (0757)
553 DATA 296,60,160,20,-10,90,90,3
34,66,160,30,-8,120,120,314,66,160
,20,-8,110,120,314,66,162,20,-10,1
10,120,290,70,160,10,-8,70,100,280
,70,150,10,-10,64,70,296,60,20,160
,10,90,90,280,40,40,150,8,130,110,
280,40,40,150,8,130,110                           (E153)
554 DATA 300,40,30,160,8,80,120,25
0,60,30,130,10,130,110,160,340,40,
150,12,32,30,160,340,150,40,-12,32
,30,240,110,150,-20,-10,120,80,450
,322,140,100,-10,50,40,450,322,60,
20,-10,50,40,450,322,20,80,10,50,4
2                                                 (CB2E)
555 DATA 304,110,40,160,8,160,60,3
24,110,20,190,10,110,100,330,110,3
0,190,10,90,90,294,286,-10,60,12,4
0,34,294,286,190,120,-12,40,34,294
,286,50,-20,-16,40,34,294,286,130,
200,16,40,34,294,286,7,190,13,70,6
8,294,286,180,-10,-13,70,68                       (E54D)
556 DEG:RESTORE 553:t=0:FOR s=0 TO
 25:READ o1(s),o2(s),vn,na,st,w1,w
2:ORIGIN o1,o2:FOR i=vn TO na STEP
 st:ce1(s,t)=FNk1:ce2(s,t)=FNk2:t=
t+1:NEXT i:ce1(s,t)=-1:ce2(s,t)=-1
:t=0:NEXT s:RETURN                                (F6C7)
557 DATA 34,22,yMULTy,2,7,yMGTy,34
,5,yREALy,34,9,yWARPy,2,11,yERGy,3
,15," ly ",3,19," 2y "                            (5854)
558 DATA 12,10,P,17,9,I,22,9,N,27,
11,B,28,12,A,29,13,L,30,16,L                      (3C25)
559 IF bs(d)=1 THEN RETURN                        (1527)
560 PAPER 3:PEN 1:LOCATE x(d),y(d)
:PRINT ab(d):PAPER 0:bs(d)=1:FOR i
=0 TO 6:IF bs(i)=0 THEN RETURN ELS
E NEXT                                            (5D66)
561 FOR i=1 TO 25:GOSUB 163:NEXT:s
c=3:FOR i=40 TO 4 STEP-1:GOSUB 549
:FOR s=0 TO 6:PAPER 2:PEN 3:LOCATE
 x(s),y(s):PRINT ab(s):SOUND 7,s*i
,3,15,4,3:GOSUB 465:PAPER 3:PEN 2:
LOCATE x(s),y(s):PRINT ab(s):NEXT
s,i                                               (B532)
562 m=50:mg=mg+30:GOSUB 189:sc=25:
mult=mult+2:GOSUB 404:GOSUB 467:GO
SUB 372:BORDER 0:GOSUB 83:PAPER 0                 (5530)
563 FOR i=0 TO 6:bs(i)=0:NEXT:RETU
RN                                                (2079)
564 IF bs(6)=1 THEN t=1 ELSE t=0                  (1F75)
565 t=31:FOR i=15 TO 17:LOCATE t,i
:CALL q:PRINT"j":GOSUB 150:PAPER 3
:PEN t:LOCATE 30,16:PRINT ab(6):PA
PER 0:PEN 2:LOCATE t,i:PRINT" ":t=
t-1:NEXT:RETURN                                   (7380)
```

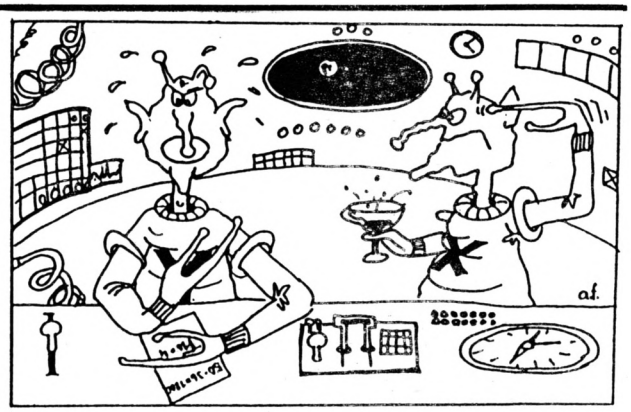
Diskettenmonitor

Der Diskettenmonitor wurde von unserem Autor Paul Nathmann zu 100 Prozent in Maschinensprache geschrieben. Die extrem hohe Arbeitsgeschwindigkeit macht dies deutlich. Lassen Sie sich von den gewaltigen DATA Zeilen nicht abschrecken, denn mit Diskmonitor erhalten Sie ein äußerst komfortables Programm zur Bearbeitung Ihrer Disketten.



Unser Disketten-Monitor bietet neben den „üblichen" Funktionen wie Sektor laden oder Sektor ändern weitere nützliche Routinen, etwa Drukkerausgabe, Berechnung von Spur und Sektor und Informationen zum Directory-Bereich.
Nach Programmstart erscheint zunächst ein Einschaltbild. Hier können Sie ein Laufwerk auswählen. Mit der ENTER-Taste wird die aktuelle Floppy übernommen. Jetzt erst kommen Sie in den Diskettenmonitor. Der Bildschirmaufbau sieht folgendermaßen aus: In der ersten Zeile werden der aktuelle Track, Sektor und Blocknummer ausgegeben. Auch die jeweiligen Recordteile und das Format der Diskette (DATA oder CPM) werden genannt. Im unteren Viertel des Monitors haben Sie ständig alle Funktionen von Diskmonitor im Blick. Den Rest des Bildschirms nimmt der aktuelle Sektor in Anspruch; er wird in Hex-Zahlen und ASCII-Codes ausgegeben.
Mit CTRL-E wird der Monitor verlassen, der CPC springt in den Ready-Modus. Mit der Taste C können Sie das Laufwerk ändern oder sich schlicht und einfach ein Directory ausgeben lassen. Taste B steht für Spur und Sektor berechnen. Dies wird nach Eingabe der Blocknummer erledigt und das Ergebnis ausgegeben. Mit P wird ein ganzer Sektor gedruckt. Ist der Drucker nicht bereit, wird nach fünf Sekunden ins Programm zurückgesprungen. Durch Betätigen der Taste I wird der Aufbau des Directories angezeigt – eine äußerst nützliche Sache, wenn Sie mit Diskmonitor im Directory herumPOKEn wollen.
Taste L steht für Sektor laden, Taste S für Sektor auf Diskette schreiben. Da ein vollständiger Sektor nicht komplett auf den





Monitor paßt, wird er vom Programm in zwei Seiten aufgeteilt. Mit CTRL-CURSOR LINKS wird eine halbe Sektoranzeige zurück-, mit CTRL-CURSOR RECHTS eine halbe Sektoranzeige vorgeblättert. Mit dieser Funktion können Sie die ganze Diskette durchforsten.
Mit Taste A kommen Sie ins Herz von Diskmonitor, nämlich in den Menüpunkt Verändern des Diskettenspeichers. Es erscheint ein neues Menü. Mit den Cursorpfeilen geht es in alle Richtungen. Beim Drücken von COPY wechselt der Cursor vom Hex- ins ASCII-Feld oder zurück. Im Hex-Feld können nur Hexadezimal-Werte eingegeben werden, im ASCII-Feld die ASCII-Werte null bis 127. Mit dem Pfund-Zeichen läßt sich in beiden Feldern das Füllbyte (&E5) eingeben. Halten Sie die Taste gedrückt, können Sie einen Sektor schnell überschreiben beziehungsweise reinigen.
Mit CTRL-TAB wird das siebte Bit des jeweiligen Bytes gewechselt. Dies eignet sich gut zum Ändern von Byte neun und zehn im Directory:

File R/O (Read Only): Bit sieben.
File R/W (Read Write): Bit sieben gleich Null
File SYS (unsichtbar): Bit sieben gleich Eins
File DIR (sichtbar): Bit sieben gleich Null

Mit ESC wird der Änderungsmodus wieder verlassen, nachdem gefragt wurde, ob abgespeichert werden soll oder nicht.
Die Druckerausgabe ist auf Entwurfsdruck eingestellt. Wer lieber eine saubere NLQ-Ausgabe wünscht, dem kann mit POKE &4ACC,&C9 geholfen werden. POKE &4ACC,&3E wechselt wieder auf Entwurfsdruck. Vor dem Start kann mit CALL &4989: POKE &4B58,1 der blinkende Cursor ausgeschaltet werden. CALL &4948 bringt ihn wieder zum Blinken.
Tippen Sie den BASIC-Lader "DISMONI.BAS" ein und speichern Sie ihn auf Diskette ab. Dieses Programm enthält eine spezielle Laderoutine für den CPC 6128, mit der eine bestimmte Bank gewählt werden kann. Für den CPC 464 und 664 wird ganz normal das Binärprogramm geladen. Als nächstes muß der Datalader "DISMONI.LAD" eingegeben und vorsichtshalber abgespeichert werden. Danach wird er mit RUN gestartet. Dadurch wird ein Binärfile abgespeichert, und der CPC springt in den Ready-Modus. Jetzt tippen Sie "GOTO 662" (ENTER) ein. Das Binärfile wird wieder eingeladen, an einem anderen Speicherplatz (&4000) abgelegt und wieder abgespeichert. Beachten Sie, daß nur die zweite Version lauffähig ist! *TB/Paul Nathmann* □

```
1 '******************************  <24B1>
2 '*        DISKMONITOR         *  <2420>
3 '*             VON            *  <24D0>
4 '*       PAUL NATHMANN        *  <24B7>
5 '*             FUER           *  <2424>
6 '*    SCHNEIDER CPC-WELT      *  <24EF>
7 '*     CPC 464/664/6128    TB*  <24DB>
8 '******************************  <24BF>
10 '                               <071F>
20 MODE 2                          <07D5>
30 IF PEEK(&BB4F)=&74 THEN 70:' CP
C 6128                             <21CD>
40 GOSUB 170                       <0973>
50 PRINT"Starten mit CALL &4000"   <1E05>
60 END                             <06DA>
70 INPUT"In welche Bank 4-7 sonst
normal ";b$                        <2D27>
80 b=VAL(b$):IF b<4 OR b>7 THEN 40
                                   <25AD>
90 e=b+&C0                         <12F3>
100 OUT &7F00,e:' bei CPC 6128 Ban
k einschalten                      <2F20>
110 h=HIMEM:GOSUB 170              <115F>
120 CALL &BD5B:' wieder normale Ba
nk, call ohne Parameter AKKU = 0   <3E27>
130 MEMORY h:PRINT"Starten mit"    <194E>
140 t$="MEMORY &3FFF:call &BD5B"+M
ID$(",1,2,3,4,5,6,7",1,b*2)+":CALL
 &4000"                            <4EB1>
150 PRINT t$                       <0A4D>
160 END                            <06A3>
170 MEMORY &3FFF                   <095D>
180 LOAD"DISMONI.BIN",&4000        <17A4>
190 RETURN                         <06A2>
```

```
100 '****************************  <2234>
110 '* DATALADER ZU DISKMONITOR*   <2253>
120 '*   ERZEUGT <DIMONI.BIN>  *   <22C3>
130 '*           VON           *   <2275>
140 '*      PAUL NATHMANN      *   <22C0>
150 '*           FUER          *   <2228>
160 '*   SCHNEIDER CPC-WELT    *   <2247>
170 '*    CPC 464/664/6128  TB*   <22DE>
180 '****************************  <22D4>
200 MEMORY &7FFF                   <0998>
650 a=&8000:e=&95D4:zb=1000:e=e+1  <2CB9>
660 FOR i=a TO e:IF i=e THEN SAVE"
DISMONI.BIN",B,&8000,&15D4:PRINT"J
etzt eingeben: goto 662":DELETE 67
0-                                 <5D2A>
661 IF i<e THEN GOTO 670           <1497>
662 MEMORY &3FFF:LOAD"DISMONI.BIN"
,&4000:SAVE"DISMONI.BIN",B,&4000,&
15D4:END                           <3A6F>
```

```
670 READ d$:POKE i,VAL("&"+d$)     <1D5A>
730 IF i<e THEN NEXT i             <1526>
1001 DATA ED,73,6E,4B,CD,00,BB,0E  <1E43>
1002 DATA 00,CD,15,B9,21,70,4B,CD  <1E07>
1003 DATA D4,BC,D2,17,43,3E,FF,CD  <1E57>
1004 DATA 1B,00,21,71,4B,CD,D4,BC  <1E1A>
1005 DATA D2,17,43,22,5F,4B,79,32  <1E68>
1006 DATA 61,4B,21,72,4B,CD,D4,BC  <1E4F>
1007 DATA D2,17,43,22,62,4B,79,32  <1EFB>
1008 DATA 64,4B,11,7B,4B,CD,D3,45  <1E02>
1009 DATA 3A,58,4B,B7,20,03,CD,48  <1E70>
1010 DATA 49,ED,7B,6E,4B,AF,32,9F  <1E53>
1011 DATA A8,32,DF,A8,32,5D,4B,32  <1E4B>
1012 DATA 6B,4B,32,01,A7,CD,57,41  <1EA1>
1013 DATA CD,76,41,3E,15,CD,5A,BB  <1E5B>
1014 DATA CD,50,41,F5,3E,06,CD,5A  <1E7B>
1015 DATA BB,F1,CA,2C,43,CD,41,49  <1EA4>
1016 DATA 06,02,28,19,06,00,3E,C1  <1EA6>
1017 DATA BE,28,12,11,D9,4C,CD,D3  <1E33>
1018 DATA 45,11,6B,4C,CD,D3,45,CD  <1EBE>
1019 DATA C8,45,C3,49,40,78,32,5C  <1E78>
1020 DATA 4B,32,5D,4B,3E,01,32,5E  <1EA7>
1021 DATA 4B,CD,BA,43,CD,CF,43,11  <1E4C>
1022 DATA 26,4E,CD,D3,45,CD,84,BB  <1EA6>
1023 DATA CD,C8,45,FE,FA,28,2D,FE  <1E2E>
1024 DATA FB,28,2F,FE,05,28,64,E6  <1E39>
1025 DATA DF,FE,4C,28,2B,FE,42,28  <1E76>
1026 DATA 30,FE,49,28,32,FE,41,28  <1E38>
1027 DATA 34,FE,50,28,42,FE,53,28  <1E14>
1028 DATA 35,FE,43,28,40,00,00,00  <1ED8>
1029 DATA 00,00,18,C9,CD,F2,45,C3  <1E9F>
1030 DATA AD,40,CD,2A,46,C3,AD,40  <1EDB>
1031 DATA CD,61,46,D2,49,40,C3,A7  <1E10>
1032 DATA 40,CD,86,46,C3,A7,40,CD  <1E1A>
1033 DATA 60,4A,C3,A7,40,CD,DF,46  <1E35>
1034 DATA D2,49,40,C3,A7,40,CD,25  <1EC8>
1035 DATA 49,D2,49,40,C3,A7,40,CD  <1E8B>
1036 DATA 96,49,C3,A7,40,CD,34,41  <1EA8>
1037 DATA C3,75,40,CD,6C,BB,21,70  <1E66>
1038 DATA 4B,CD,D4,BC,D2,17,43,AF  <1E15>
1039 DATA CD,1B,00,C9,CD,57,41,CD  <1EFA>
1040 DATA 76,41,21,05,01,CD,8B,44  <1E91>
1041 DATA CD,50,41,CA,2C,43,11,BE  <1E6D>
1042 DATA 4C,CD,D3,45,CD,C8,45,C9  <1ED5>
1043 DATA 11,00,60,CD,9B,BC,C9,11  <1E55>
1044 DATA 83,4B,CD,D3,45,21,0F,C0  <1E99>
1045 DATA 3E,15,CD,F0,42,21,A9,C0  <1E47>
1046 DATA 3E,1F,CD,F0,42,21,03,C1  <1E9E>
1047 DATA 3E,14,CD,F0,42,C9,21,54  <1E87>
1048 DATA 00,11,C5,00,D5,CD,EA,BB  <1E46>
1049 DATA D1,21,3E,01,E5,CD,F6,BB  <1ECB>
1050 DATA E1,11,A9,01,D5,CD,F6,BB  <1E67>
1051 DATA D1,21,54,00,E5,CD,F6,BB  <1EC0>
1052 DATA E1,11,C6,00,D5,CD,F6,BB  <1EED>
1053 DATA D1,21,3E,01,E5,CD,F6,BB  <1ED3>
1054 DATA E1,11,AA,01,D5,CD,EA,BB  <1E5A>
1055 DATA D1,21,54,00,CD,F6,BB,0E  <1E08>
1056 DATA 0D,41,21,D7,00,16,01,AF  <1E90>
1057 DATA DD,21,82,55,CD,C8,42,3C  <1E46>
1058 DATA 41,DD,21,9A,55,CD,C8,42  <1E43>
1059 DATA 0E,05,41,21,9F,00,AF,DD  <1E21>
1060 DATA 21,9C,55,CD,C8,42,41,DD  <1E10>
1061 DATA 21,A4,55,3C,CD,C8,42,0E  <1EAC>
1062 DATA 04,41,21,28,01,16,00,AF  <1EA4>
1063 DATA DD,21,A6,55,CD,C8,42,41  <1ED1>
1064 DATA DD,21,AC,55,3C,CD,C8,42  <1E97>
1065 DATA 41,21,28,01,16,01,AF,DD  <1E68>
1066 DATA 21,AE,55,CD,C8,42,41,DD  <1E1E>
1067 DATA 21,B4,55,3C,CD,C8,42,0E  <1ED8>
1068 DATA 03,41,21,38,01,AF,DD,21  <1E1F>
1069 DATA B6,55,CD,C8,42,41,DD,21  <1EBE>
1070 DATA BA,55,3C,CD,C8,42,41,21  <1E23>
1071 DATA 38,01,16,00,AF,DD,21,BC  <1EBE>
1072 DATA 55,CD,C8,42,41,DD,21,C0  <1EAE>
1073 DATA 55,3C,CD,C8,42,0E,08,41  <1ED6>
1074 DATA 21,2D,01,16,01,AF,DD,21  <1EC5>
1075 DATA C2,55,CD,C8,42,41,DD,21  <1ED5>
1076 DATA D0,55,21,EF,00,3C,CD,C8  <1EB3>
1077 DATA 42,11,47,01,06,25,C5,D5  <1E86>
1078 DATA 21,F0,00,CD,EA,BB,D1,D5  <1E03>
1079 DATA 21,1C,01,CD,F6,BB,D1,1B  <1E43>
1080 DATA C1,10,EB,11,0E,4C,CD,D3  <1E6A>
1081 DATA 45,3E,83,CD,D9,45,CD,58  <1E28>
1082 DATA 43,CD,C8,45,FE,0D,28,1D  <1EC2>
1083 DATA E6,5F,FE,41,28,04,FE,42  <1ED2>
1084 DATA 20,EF,CB,FF,21,5B,4B,77  <1EE4>
1085 DATA CD,D4,BC,D2,17,43,AF,CD  <1E9B>
1086 DATA 1B,00,DA,2C,43,11,5D,4C  <1EDE>
1087 DATA CD,D3,45,3A,02,A7,C6,41  <1EE0>
1088 DATA CD,5A,BB,11,6B,4C,CD,D3  <1E44>
1089 DATA 45,CD,C8,45,CD,84,BB,C9  <1EBE>
1090 DATA C5,F5,E5,DD,5E,00,D5,CD  <1ECD>
1091 DATA C0,BB,D1,DD,5E,01,D5,CD  <1ED1>
1092 DATA F6,BB,D1,E1,2D,2D,F1,B7  <1E35>
1093 DATA 20,08,DD,23,DD,23,C1,10  <1EB7>
1094 DATA DF,C9,DD,2B,DD,2B,18,F6  <1EE1>
1095 DATA 11,00,08,E5,F5,4A,7E,23  <1E9C>
1096 DATA 06,04,1F,CB,1E,CB,2E,10  <1EF0>
1097 DATA F9,2B,06,04,1F,CB,1E,CB  <1EF5>
1098 DATA 2E,10,F9,19,0D,20,E7,F1  <1EB9>
1099 DATA E1,23,23,3D,20,DD,C9,CD  <1E66>
1100 DATA 57,41,11,FA,4C,CD,D3,45  <1E56>
1101 DATA CD,58,43,CD,C8,45,CD,23  <1E36>
1102 DATA 41,C3,00,00,F5,CD,57,41  <1EF4>
1103 DATA CD,58,43,F1,CB,77,28,0E  <1E83>
1104 DATA 11,24,55,CB,4F,20,0A,11  <1E5F>
1105 DATA 53,55,CB,47,20,03,11,ED  <1E40>
1106 DATA 54,CD,D3,45,11,BE,4C,CD  <1E9D>
1107 DATA D3,45,CD,C8,45,C3,49,40  <1EAC>
1108 DATA 3E,01,01,00,00,CD,32,BC  <1E29>
1109 DATA C9,01,02,02,21,59,4B,CD  <1EE5>
1110 DATA D4,44,20,08,3A,5D,4B,CD  <1E80>
1111 DATA 8C,45,B7,C9,11,00,00,CD  <1EAC>
1112 DATA 1B,45,FE,28,30,05,32,5D  <1E9D>
1113 DATA 4B,B7,C9,CD,70,44,37,C9  <1EFB>
1114 DATA 01,01,01,21,59,4B,CD,D4  <1EC3>
1115 DATA 44,20,06,3A,5E,4B,C3,8C  <1E96>
1116 DATA 45,2B,CD,A5,44,7E,D6,30  <1EAD>
1117 DATA 28,05,32,5E,4B,B7,C9,CD  <1E69>
1118 DATA 70,44,37,C9,11,83,4D,CD  <1E55>
1119 DATA D3,45,CD,C8,45,E6,5F,FE  <1EA8>
1120 DATA 4E,C9,AF,32,65,4B,06,01  <1EE0>
1121 DATA C3,38,45,3A,02,A7,21,9F  <1EFF>
1122 DATA A8,B7,C8,21,DF,A8,C9,11  <1EB1>
```

```
1123 DATA 73,4B,CD,D3,45,CD,41,49 (1EC2)
1124 DATA 20,08,11,AC,4D,CD,D3,45 (1E15)
1125 DATA 18,06,11,B1,4D,CD,D3,45 (1E73)
1126 DATA 11,B6,4D,CD,D3,45,3A,5D (1E1B)
1127 DATA 4B,CD,8C,45,CD,DC,45,3A (1E65)
1128 DATA 5D,4B,CD,60,45,CD,DC,45 (1E23)
1129 DATA 3A,5E,4B,CD,8C,45,CD,DC (1E69)
1130 DATA 45,3A,65,4B,3C,CD,8C,45 (1E72)
1131 DATA D5,CD,E3,4A,D1,CD,DC,45 (1E2A)
1132 DATA 11,00,5D,3A,65,4B,B7,28 (1ECE)
1133 DATA 03,11,00,5E,0E,00,21,04 (1E8E)
1134 DATA 01,E5,CD,75,BB,E1,06,10 (1EE6)
1135 DATA 3A,65,4B,CD,60,45,79,CD (1E69)
1136 DATA 60,45,D5,3E,20,CD,5A,BB (1E62)
1137 DATA 3E,20,CD,5A,BB,1A,CD,60 (1E44)
1138 DATA 45,13,10,F4,3E,20,CD,5A (1EE9)
1139 DATA BB,CD,5A,BB,D1,06,10,1A (1EBC)
1140 DATA E5,D5,C5,CD,5D,BB,C1,D1 (1E92)
1141 DATA E1,13,10,F3,3E,10,81,4F (1EF7)
1142 DATA 3E,14,2C,BD,20,BB,37,C9 (1EA6)
1143 DATA CD,84,BB,21,18,30,11,9C (1E60)
1144 DATA 4D,CD,D0,45,11,BE,4C,CD (1ED6)
1145 DATA D3,45,CD,C8,45,C3,81,BB (1EB1)
1146 DATA 21,18,01,CD,75,BB,3E,14 (1E5C)
1147 DATA C3,5A,BB,21,18,01,11,74 (1EDE)
1148 DATA 4D,CD,D0,45,3A,5E,4B,CD (1E41)
1149 DATA 8C,45,C3,DC,45,E5,E5,3E (1ECA)
1150 DATA 08,CD,5A,BB,3E,20,CD,5A (1E2D)
1151 DATA BB,E1,7E,CD,5A,BB,E1,C9 (1E6C)
1152 DATA 11,66,4D,21,17,01,CD,D0 (1E08)
1153 DATA 45,3A,5D,4B,18,D9,11,34 (1E9A)
1154 DATA 4D,21,16,01,CD,D0,45,3A (1E42)
1155 DATA 5C,4B,18,CB,CD,81,BB,CD (1ED2)
1156 DATA C8,45,FE,7F,28,25,FE,0D (1E12)
1157 DATA 28,33,FE,30,38,F1,FE,3A (1EB0)
1158 DATA 30,ED,77,CD,5A,BB,23,10 (1E57)
1159 DATA E6,CD,C8,45,FE,7F,28,0F (1ED7)
1160 DATA FE,0D,28,19,3E,07,CD,5A (1E10)
1161 DATA BB,18,EE,78,B9,28,D0,3E (1E09)
1162 DATA 08,CD,5A,BB,3E,10,CD,5A (1E4F)
1163 DATA BB,04,2B,18,C2,CD,84,BB (1EBE)
1164 DATA 78,B9,C9,21,59,4B,FE,01 (1EC0)
1165 DATA 28,0F,7E,D6,30,87,4F,87 (1E82)
1166 DATA 87,81,57,23,7E,D6,30,82 (1E06)
1167 DATA C9,CD,A5,44,16,00,18,F4 (1E83)
1168 DATA 3A,5E,4B,4F,0D,CD,C3,43 (1E32)
1169 DATA 7E,81,4F,3A,02,A7,5F,3A (1E45)
1170 DATA 5D,4B,57,21,00,5D,05,20 (1E11)
1171 DATA 09,F3,DF,5F,4B,FB,D2,2C (1E5D)
1172 DATA 43,C9,F3,DF,62,4B,18,F5 (1E03)
1173 DATA F5,C5,4F,0F,0F,0F,0F,E6 (1E20)
1174 DATA 0F,CD,75,45,79,E6,0F,CD (1EC1)
1175 DATA 75,45,C1,F1,C9,FE,0A,38 (1E45)
1176 DATA 02,C6,07,C6,30,47,3A,6B (1E9A)
1177 DATA 4B,B7,20,04,78,C3,5A,BB (1E18)
1178 DATA 78,C3,DB,4A,D5,C5,F5,FE (1E5B)
1179 DATA 0A,38,2C,4F,06,04,58,D6 (1E41)
1180 DATA 0A,38,02,10,FA,7B,90,47 (1E00)
1181 DATA 79,C6,06,10,FC,5F,CB,3F (1E9D)
1182 DATA CB,3F,CB,3F,CB,3F,C6,30 (1E11)
1183 DATA CD,7D,45,7B,E6,0F,C6,30 (1EBB)
1184 DATA CD,7D,45,F1,C1,D1,C9,F5 (1EC0)
1185 DATA 3E,20,CD,7D,45,F1,18,EE (1EF5)
1186 DATA CD,09,BB,38,FB,C3,06,BB (1E8B)
1187 DATA CD,75,BB,1A,FE,80,F2,DF (1E59)
1188 DATA 45,CD,5A,BB,13,18,F4,FE (1E44)
1189 DATA FF,C8,FE,90,30,F3,D6,80 (1EDE)
1190 DATA 47,3E,20,CD,5A,BB,10,FB (1E7B)
1191 DATA 18,EA,3A,65,4B,B7,28,07 (1E9E)
1192 DATA 97,32,65,4B,C3,CF,43,3C (1E3D)
1193 DATA 32,65,4B,3A,5E,4B,FE,01 (1E6D)
1194 DATA 28,0B,3D,32,5E,4B,CD,BA (1EB8)
1195 DATA 43,3E,01,18,E4,3E,09,32 (1EA9)
1196 DATA 5E,4B,3A,5D,4B,B7,28,06 (1E95)
1197 DATA 3D,32,5D,4B,18,E8,3E,27 (1ED7)
1198 DATA 18,F7,3A,65,4B,B7,20,07 (1E33)
1199 DATA 3C,32,65,4B,C3,CF,43,3A (1E43)
1200 DATA 5E,4B,FE,09,28,0A,3C,32 (1E0D)
1201 DATA 5E,4B,CD,BA,43,C3,CF,43 (1E3D)
1202 DATA 3E,01,32,5E,4B,3A,5D,4B (1EF8)
1203 DATA FE,27,28,0A,3C,32,5D,4B (1EE3)
1204 DATA CD,BA,43,C3,CF,43,97,18 (1EEC)
1205 DATA F4,21,15,01,CD,8B,44,CD (1EF5)
1206 DATA C6,44,CD,B8,44,CD,61,43 (1E0E)
1207 DATA 38,F8,CD,93,44,CD,88,43 (1E4F)
1208 DATA 38,F8,CD,AC,43,37,C8,CD (1E6B)
1209 DATA BA,43,D0,C3,CF,43,11,2E (1E41)
1210 DATA 4F,CD,D3,45,CD,81,BB,CD (1E79)
1211 DATA D4,48,D0,F5,21,17,01,11 (1E3C)
1212 DATA 48,4F,CD,D0,45,F1,D5,06 (1EC0)
1213 DATA 00,60,6F,29,11,12,00,19 (1E6D)
1214 DATA 1E,09,B7,04,ED,52,30,FB (1E28)
1215 DATA 05,B7,19,4D,CD,41,49,28 (1EB8)
1216 DATA 02,05,05,78,D1,CD,8C,45 (1EA9)
1217 DATA F5,CD,DC,45,F1,CD,60,45 (1EAB)
1218 DATA CD,DC,45,79,3C,CD,8C,45 (1E93)
1219 DATA CD,84,BB,11,BE,4C,CD,D3 (1EC5)
1220 DATA 45,CD,C8,45,C3,88,44,11 (1E38)
1221 DATA 72,4F,CD,D3,45,21,04,07 (1E98)
1222 DATA 22,66,4B,7C,32,6D,4B,CD (1E46)
1223 DATA 75,BB,CD,81,BB,3E,38,32 (1EAE)
1224 DATA 6C,4B,21,00,5D,3A,65,4B (1E26)
1225 DATA B7,28,03,21,00,5E,22,68 (1EF2)
1226 DATA 4B,CD,C8,45,FE,F0,28,4C (1EF5)
1227 DATA FE,F1,28,4D,FE,F2,28,4E (1E3B)
1228 DATA FE,F3,28,31,FE,E0,28,53 (1EA5)
1229 DATA FE,E1,28,61,FE,A3,CA,C5 (1EDC)
1230 DATA 48,FE,FC,CA,DA,47,FE,67 (1ED7)
1231 DATA 30,D7,FE,30,38,D3,CD,DA (1EDD)
1232 DATA 48,F5,2A,68,4B,77,3A,6C (1E34)
1233 DATA 4B,CD,6F,BB,F1,CD,5D,BB (1E45)
1234 DATA CD,96,48,18,06,CD,96,48 (1E9C)
1235 DATA D2,09,47,3A,6D,4B,CD,C0 (1E85)
1236 DATA 47,C3,09,47,CD,95,47,18 (1E98)
1237 DATA A8,CD,A6,47,18,A3,CD,67 (1E2B)
1238 DATA 48,38,9E,3A,6D,4B,CD,C0 (1E13)
1239 DATA 47,18,96,3A,6C,4B,CD,8E (1EB9)
1240 DATA 47,C3,F8,47,3A,6D,4B,CD (1EE5)
1241 DATA 8E,47,C3,09,47,CD,CA,47 (1E61)
1242 DATA F5,CD,60,45,18,B0,32,67 (1EE6)
1243 DATA 4B,CD,6F,BB,C9,3A,66,4B (1EA4)
1244 DATA FE,05,D8,3D,CD,B3,47,B7 (1E1B)
1245 DATA ED,52,22,68,4B,C9,3A,66 (1E89)
1246 DATA 4B,FE,13,D0,3C,CD,B3,47 (1E03)
```

```
1247 DATA 19,18,EF,32,66,4B,CD,72 <1EE0>
1248 DATA BB,2A,68,4B,11,10,00,C9 <1EAD>
1249 DATA 32,67,4B,2A,66,4B,CD,75 <1E3E>
1250 DATA BB,C9,2A,68,4B,7E,CB,7F <1EDB>
1251 DATA 28,04,CB,BF,18,02,CB,FF <1E77>
1252 DATA 77,C9,21,19,1E,11,19,50 <1EF1>
1253 DATA CD,D0,45,CD,F0,47,37,C0 <1EAC>
1254 DATA CD,84,BB,06,FF,C3,38,45 <1ED4>
1255 DATA CD,C8,45,E6,5F,FE,4A,C9 <1E0B>
1256 DATA CD,C8,45,FE,F0,28,47,FE <1EB9>
1257 DATA F1,28,48,FE,F2,28,49,FE <1EF1>
1258 DATA F3,28,2E,FE,E0,CA,7C,47 <1ECA>
1259 DATA FE,E1,28,4D,FE,A3,28,45 <1E02>
1260 DATA FE,FC,CA,DA,47,FE,81,D2 <1EE6>
1261 DATA F8,47,F5,2A,68,4B,77,CD <1EBB>
1262 DATA 5D,BB,3A,6D,4B,CD,6F,BB <1E21>
1263 DATA F1,CD,60,45,CD,96,48,18 <1E4B>
1264 DATA 05,CD,96,48,30,BA,3A,6C <1EB8>
1265 DATA 4B,CD,C0,47,18,B2,CD,95 <1E40>
1266 DATA 47,18,AD,CD,A6,47,18,A8 <1E25>
1267 DATA CD,67,48,38,A3,3A,6C,4B <1EC1>
1268 DATA CD,C0,47,18,9B,3E,E5,18 <1EFF>
1269 DATA C1,CD,CA,47,F5,18,C0,3A <1E50>
1270 DATA 6C,4B,FE,39,38,15,3D,32 <1E43>
1271 DATA 6C,4B,3A,6D,4B,D6,03,32 <1E95>
1272 DATA 6D,4B,2A,68,4B,2B,22,68 <1E90>
1273 DATA 4B,B7,C9,3A,66,4B,FE,05 <1E71>
1274 DATA D8,3D,32,66,4B,3E,47,32 <1EA0>
1275 DATA 6C,4B,3E,34,18,E1,3A,6C <1E87>
1276 DATA 4B,FE,47,30,15,3C,32,6C <1E3D>
1277 DATA 4B,3A,6D,4B,C6,03,32,6D <1E39>
1278 DATA 4B,2A,68,4B,23,22,68,4B <1E3B>
1279 DATA 37,C9,3A,66,4B,FE,13,D0 <1E03>
1280 DATA 3C,32,66,4B,3E,38,32,6C <1E82>
1281 DATA 4B,3E,07,18,E1,3E,45,CD <1E6D>
1282 DATA 5A,BB,3E,35,CD,5A,BB,3E <1E40>
1283 DATA E5,C3,39,47,CD,C8,45,FE <1E6D>
1284 DATA 0D,C8,06,02,21,59,4B,18 <1EB3>
1285 DATA 08,3E,07,CD,5A,BB,CD,C8 <1E31>
1286 DATA 45,FE,30,38,F4,FE,47,38 <1ED0>
1287 DATA 0C,FE,61,38,EC,FE,67,30 <1EDC>
1288 DATA E8,D6,20,18,08,FE,3A,38 <1ECD>
1289 DATA 06,FE,41,38,DC,D6,07,77 <1E6F>
1290 DATA FE,3A,38,02,C6,07,CD,5A <1E95>
1291 DATA BB,23,10,D2,2B,2B,7E,D6 <1E16>
1292 DATA 30,87,87,87,87,5F,23,7E <1E81>
1293 DATA D6,30,83,37,C9,21,15,01 <1E0E>
1294 DATA CD,8B,44,CD,C6,44,CD,B8 <1EBD>
1295 DATA 44,CD,61,43,38,F8,CD,93 <1EF2>
1296 DATA 44,CD,88,43,38,F8,C3,DA <1E5D>
1297 DATA 47,CD,C3,43,3E,41,BE,C9 <1E5A>
1298 DATA 21,51,4B,06,80,0E,00,11 <1E6A>
1299 DATA 67,49,CD,EF,BC,21,4B,4B <1E64>
1300 DATA 11,12,00,01,12,00,CD,E9 <1EA2>
1301 DATA BC,3E,07,32,58,4B,C9,F3 <1EC7>
1302 DATA F5,C5,D5,E5,3A,4A,4B,B7 <1EC7>
1303 DATA 28,09,CD,7E,BB,97,32,4A <1E4D>
1304 DATA 4B,18,08,CD,7B,BB,3E,FF <1EF4>
1305 DATA 32,4A,4B,E1,D1,C1,F1,FB <1E67>
1306 DATA C9,F3,21,4B,4B,CD,EC,BC <1E19>
1307 DATA AF,32,58,4B,FB,C9,11,13 <1E1E>
1308 DATA 4F,CD,D3,45,CD,F0,47,C0 <1E5F>
1309 DATA 3E,0D,4F,06,10,79,CD,2B <1E5A>
1310 DATA BD,38,03,10,F8,C9,3E,01 <1E94>
1311 DATA 32,6B,4B,CD,CC,4A,CD,41 <1E7F>
1312 DATA 49,11,B1,4D,20,03,11,AC <1E33>
1313 DATA 4D,CD,C2,4A,11,B6,4D,CD <1E9F>
1314 DATA C2,4A,3A,5D,4B,F5,CD,8C <1E45>
1315 DATA 45,CD,C9,4A,F1,CD,60,45 <1E6A>
1316 DATA CD,C9,4A,3A,5E,4B,CD,8C <1EB9>
1317 DATA 45,CD,56,4A,11,DE,4D,CD <1EA7>
1318 DATA C2,4A,CD,51,4A,11,00,5D <1EBB>
1319 DATA 0E,00,21,00,00,CD,0B,4A <1EBB>
1320 DATA 0E,00,21,00,01,CD,0B,4A <1EE5>
1321 DATA 11,2F,50,CD,C2,4A,AF,32 <1EB8>
1322 DATA 6B,4B,C9,06,10,7C,CD,60 <1E6A>
1323 DATA 45,79,CD,60,45,D5,3E,20 <1E6F>
1324 DATA CD,89,45,3E,20,CD,89,45 <1EAE>
1325 DATA 1A,CD,60,45,13,10,F4,3E <1E7E>
1326 DATA 20,CD,89,45,CD,89,45,D1 <1E34>
1327 DATA 06,10,1A,FE,20,38,04,FE <1EE3>
1328 DATA 81,38,02,3E,2E,CD,89,45 <1E63>
1329 DATA 13,10,EF,3E,10,81,4F,CD <1EC4>
1330 DATA 56,4A,3E,10,2C,BD,20,BB <1E1B>
1331 DATA C9,3E,0A,CD,DB,4A,3E,0D <1EAF>
1332 DATA CD,DB,4A,3E,0A,C3,DB,4A <1E4D>
1333 DATA CD,41,49,20,10,3A,5D,4B <1E6B>
1334 DATA FE,03,30,1E,3A,5E,4B,FE <1E2E>
1335 DATA 05,30,17,18,0D,3A,5D,4B <1EA6>
1336 DATA B7,20,0F,3A,5E,4B,FE,05 <1E0E>
1337 DATA 30,08,11,58,50,CD,D3,45 <1E3E>
1338 DATA 18,26,11,DB,52,CD,D3,45 <1ED0>
1339 DATA D5,21,00,5D,E5,21,00,00 <1E58>
1340 DATA 54,06,43,E3,7E,23,E3,5F <1E10>
1341 DATA 19,10,F8,D1,7C,CD,60,45 <1E44>
1342 DATA 7D,CD,60,45,D1,CD,DC,45 <1E8E>
1343 DATA 11,BE,4C,CD,D3,45,CD,C8 <1EF6>
1344 DATA 45,21,04,01,CD,8B,44,C3 <1E76>
1345 DATA 18,44,1A,FE,FF,C8,CD,DB <1E50>
1346 DATA 4A,13,18,F6,3E,1B,CD,DB <1E75>
1347 DATA 4A,3E,78,CD,DB,4A,3E,30 <1E71>
1348 DATA C3,DB,4A,CD,2E,BD,38,FB <1E0E>
1349 DATA C3,31,BD,3A,5D,4B,87,87 <1E6A>
1350 DATA 5F,6F,26,00,54,29,29,29 <1E23>
1351 DATA 19,3A,5E,4B,87,87,5F,19 <1E25>
1352 DATA 0E,00,1E,04,B7,ED,52,28 <1E97>
1353 DATA 07,ED,52,28,17,0C,18,F5 <1EE0>
1354 DATA CD,2B,4B,D0,CD,3B,4B,3A <1E85>
1355 DATA 65,4B,B7,CA,D3,45,CD,36 <1E0C>
1356 DATA 4B,C3,D3,45,CD,2B,4B,D0 <1E7D>
1357 DATA CD,3B,4B,CD,36,4B,CD,36 <1EC0>
1358 DATA 4B,18,E4,CD,41,49,C0,79 <1EB6>
1359 DATA D6,09,3F,D0,4F,C9,13,13 <1E44>
1360 DATA 13,13,C9,11,C6,54,CD,D3 <1E7A>
1361 DATA 45,79,CD,60,45,CD,DC,45 <1ECC>
1362 DATA 13,C9,00,00,00,00,00,00 <1ECF>
1363 DATA 00,00,00,00,00,00,00,00 <1E4D>
1364 DATA 00,00,00,00,00,00,00,00 <1E4F>
1365 DATA 00,00,00,00,00,00,00,00 <1E51>
1366 DATA 00,00,00,00,00,00,00,00 <1E53>
1367 DATA 81,84,85,1F,4F,14,13,1F <1EFE>
1368 DATA 01,01,FF,1D,16,16,1C,00 <1E09>
1369 DATA 16,16,FF,04,02,1C,01,16 <1ECA>
1370 DATA 16,1F,10,01,43,20,50,20 <1E92>
```

```
1371 DATA 43,83,44,20,49,20,53,20 (1EE1)
1372 DATA 43,20,4D,20,4F,20,4E,20 (1EBC)
1373 DATA 49,20,54,20,4F,20,52,83 (1E3C)
1374 DATA 3C,20,31,20,2E,20,31,20 (1ECA)
1375 DATA 3E,1F,0A,03,41,20,75,20 (1E35)
1376 DATA 74,20,6F,20,72,83,50,20 (1E4F)
1377 DATA 61,20,75,20,6C,83,4E,20 (1EE4)
1378 DATA 61,20,74,20,68,20,6D,20 (1E22)
1379 DATA 61,20,6E,20,6E,83,34,20 (1E4A)
1380 DATA 34,20,30,20,33,83,53,20 (1EDE)
1381 DATA 65,20,6E,20,64,20,65,20 (1E97)
1382 DATA 6E,1F,14,04,A4,83,62,20 (1EEC)
1383 DATA 79,83,53,20,63,20,68,20 (1EBF)
1384 DATA 6E,20,65,20,69,20,64,20 (1E48)
1385 DATA 65,20,72,83,41,20,6B,20 (1E47)
1386 DATA 74,20,69,20,76,FF,1F,20 (1EB3)
1387 DATA 06,FF,1F,1C,11,41,1F,1B (1EA0)
1388 DATA 13,22,44,49,4D,4F,4E,49 (1EAC)
1389 DATA 2E,42,49,4E,22,2C,42,2C (1E3B)
1390 DATA 26,34,30,30,30,2C,26,31 (1EC2)
1391 DATA 35,44,34,1F,14,16,4C,61 (1EDF)
1392 DATA 75,66,77,65,72,6B,20,41 (1E48)
1393 DATA 20,6F,64,65,72,20,42,82 (1EA2)
1394 DATA 6D,69,74,20,45,6E,74,65 (1EB7)
1395 DATA 72,20,61,6C,74,65,72,20 (1E19)
1396 DATA 57,65,72,74,FF,11,1F,23 (1E33)
1397 DATA 16,4C,61,75,66,77,65,72 (1E17)
1398 DATA 6B,20,FF,1F,14,18,42,69 (1E1F)
1399 DATA 74,74,65,20,64,69,65,20 (1E19)
1400 DATA 75,6E,74,65,72,73,75,63 (1EBD)
1401 DATA 68,65,6E,64,65,20,44,69 (1EA7)
1402 DATA 73,6B,65,74,74,65,20,65 (1E46)
1403 DATA 69,6E,6C,65,67,65,6E,2C (1E31)
1404 DATA 1F,18,19,64,61,6E,61,63 (1E31)
1405 DATA 68,20,69,72,67,65,6E,64 (1E8D)
1406 DATA 20,65,69,6E,65,20,54,61 (1E48)
1407 DATA 73,74,65,20,64,72,75,65 (1E14)
1408 DATA 63,6B,65,6E,2E,FF,1F,32 (1E15)
1409 DATA 19,4D,69,74,20,54,61,73 (1E9B)
1410 DATA 74,65,6E,64,72,75,63,6B (1EEF)
1411 DATA 20,77,65,69,74,65,72,2E (1EF1)
1412 DATA FF,1F,1A,0C,2A,20,2A,20 (1E5A)
1413 DATA 2A,20,46,61,6C,73,63,68 (1E3E)
1414 DATA 65,20,44,69,73,6B,65,74 (1E2F)
1415 DATA 74,65,20,2A,20,2A,20,2A (1E2E)
1416 DATA 07,FF,1F,0C,0C,2A,20,2A (1EB3)
1417 DATA 20,2A,20,42,65,66,65,68 (1EC3)
1418 DATA 6C,20,6E,69,63,68,74,20 (1E5E)
1419 DATA 67,65,66,75,6E,64,65,6E (1E75)
1420 DATA 20,2D,20,42,69,74,74,65 (1E09)
1421 DATA 20,6E,65,75,20,73,74,61 (1EA5)
1422 DATA 72,74,65,6E,20,2A,20,2A (1E8B)
1423 DATA 20,2A,07,FF,44,69,72,65 (1E53)
1424 DATA 63,74,6F,72,79,20,61,75 (1E7D)
1425 DATA 66,20,54,72,61,63,6B,20 (1EB0)
1426 DATA FF,83,53,65,6B,74,6F,72 (1E28)
1427 DATA 20,28,6F,68,6E,65,20,4F (1EB2)
1428 DATA 66,66,73,65,74,29,20,31 (1EEF)
1429 DATA 20,2D,20,34,21,FF,14,54 (1EC6)
1430 DATA 72,61,63,6B,85,28,FF,29 (1E57)
1431 DATA 84,3F,20,FF,14,53,65,6B (1E42)
1432 DATA 74,6F,72,84,28,FF,29,84 (1E17)
1433 DATA 3F,20,FF,1F,32,19,41,6C (1E7F)
1434 DATA 6C,65,73,20,72,69,63,68 (1E8C)
1435 DATA 74,69,67,82,28,20,2F,6E (1EFE)
1436 DATA 29,20,3F,FF,46,61,6C,73 (1E80)
1437 DATA 63,68,65,72,20,57,65,72 (1E62)
1438 DATA 74,21,07,FF,43,50,2F,4D (1EC3)
1439 DATA FF,44,41,54,41,FF,20,46 (1EAE)
1440 DATA 6F,72,6D,61,74,20,54,72 (1EC9)
1441 DATA 61,63,6B,20,FF,20,20,26 (1EC7)
1442 DATA FF,20,20,53,65,6B,74,6F (1E47)
1443 DATA 72,20,FF,20,20,53,65,69 (1E9F)
1444 DATA 74,65,20,FF,0D,0A,41,64 (1E2A)
1445 DATA 72,2E,20,20,20,30,20,20 (1EFD)
1446 DATA 31,20,20,32,20,20,33,20 (1E56)
1447 DATA 20,34,20,20,35,20,20,36 (1EB0)
1448 DATA 20,20,37,20,20,38,20,20 (1EEE)
1449 DATA 39,20,20,41,20,20,42,20 (1EB0)
1450 DATA 20,43,20,20,44,20,20,45 (1E54)
1451 DATA 20,20,46,20,20,30,31,32 (1E69)
1452 DATA 33,34,35,36,37,38,39,41 (1E22)
1453 DATA 42,43,44,45,46,FF,1F,01 (1E7D)
1454 DATA 14,14,0A,18,20,43,54,52 (1E47)
1455 DATA 4C,20,2B,20,F2,20,F3,20 (1E3E)
1456 DATA 18,20,6E,61,65,63,68,73 (1E68)
1457 DATA 74,65,20,53,65,6B,74,6F (1E0B)
1458 DATA 72,68,61,65,6C,66,74,65 (1E67)
1459 DATA 85,18,20,42,20,18,20,54 (1E19)
1460 DATA 72,61,63,6B,20,75,2E,20 (1EDB)
1461 DATA 53,65,6B,74,6F,72,20,65 (1E9A)
1462 DATA 72,72,65,63,68,6E,65,6E (1E76)
1463 DATA 0D,0A,18,20,4C,20,18,20 (1E5E)
1464 DATA 4C,65,73,65,6E,20,6D,69 (1E4A)
1465 DATA 74,20,53,65,6B,74,6F,72 (1E5B)
1466 DATA 65,6E,20,65,69,6E,67,61 (1E86)
1467 DATA 62,65,8A,18,20,53,20,18 (1EE4)
1468 DATA 20,53,63,68,72,65,69,62 (1E3F)
1469 DATA 65,6E,20,61,75,66,20,44 (1E87)
1470 DATA 69,73,6B,65,74,74,65,0D (1E97)
1471 DATA 0A,18,20,43,20,18,20,43 (1ED8)
1472 DATA 61,74,20,6F,64,65,72,20 (1E71)
1473 DATA 44,69,73,6B,65,74,74,65 (1EF5)
1474 DATA 6E,20,77,65,63,68,73,65 (1EEE)
1475 DATA 6C,8A,18,20,49,20,18,20 (1EBF)
1476 DATA 49,6E,66,6F,72,6D,61,74 (1E66)
1477 DATA 69,6F,6E,0D,0A,18,20,41 (1ED2)
1478 DATA 20,18,20,41,65,6E,64,65 (1E06)
1479 DATA 72,6E,89,18,20,50,20,18 (1EFD)
1480 DATA 20,44,72,75,63,6B,65,6E (1E2A)
1481 DATA 89,18,20,43,54,52,4C,20 (1E80)
1482 DATA 26,20,45,20,18,20,45,6E (1EE7)
1483 DATA 64,65,FF,1F,01,14,14,1F (1ECD)
1484 DATA 1C,18,18,20,44,72,75,63 (1E55)
1485 DATA 6B,65,6E,82,28,6A,2F,20 (1E23)
1486 DATA 29,20,3F,20,18,FF,1F,01 (1E99)
1487 DATA 14,14,0A,42,6C,6F,63,6B (1ECD)
1488 DATA 6E,72,2E,20,65,69,6E,67 (1E58)
1489 DATA 65,62,65,6E,20,3F,20,FF (1EB2)
1490 DATA 54,72,61,63,6B,6E,72,2E (1EF6)
1491 DATA 20,3D,20,FF,83,26,FF,86 (1E5A)
1492 DATA 53,65,6B,74,6F,72,6E,72 (1E6E)
1493 DATA 2E,20,3D,20,FF,49,6E,66 (1ED5)
1494 DATA 6F,72,6D,61,74,69,6F,6E (1EA9)
```

```
1495 DATA 20,FF,1F,01,14,14,1F,1C <1E58>
1496 DATA 15,41,65,6E,64,65,72,75 <1E58>
1497 DATA 6E,67,73,20,2D,20,4D,6F <1EC7>
1498 DATA 64,75,73,1F,06,16,F2,0B <1E38>
1499 DATA F0,08,0A,0A,F1,0B,F3,83 <1EF6>
1500 DATA 45,69,6E,67,61,62,65,20 <1E02>
1501 DATA 61,75,66,20,43,75,72,73 <1E12>
1502 DATA 6F,72,20,20,2D,20,20,43 <1E33>
1503 DATA 54,52,4C,20,2B,20,54,41 <1E31>
1504 DATA 42,20,20,42,69,74,20,37 <1ECB>
1505 DATA 20,73,65,74,7A,65,6E,20 <1E31>
1506 DATA 6F,64,65,72,20,6C,6F,65 <1EF5>
1507 DATA 73,63,68,65,6E,1F,16,17 <1EBF>
1508 DATA 6D,69,74,20,A3,20,4C,6F <1E61>
1509 DATA 65,73,63,68,62,79,74,65 <1E8A>
1510 DATA 20,26,45,35,20,2D,20,43 <1E76>
1511 DATA 4F,50,59,20,77,65,63,68 <1E09>
1512 DATA 73,65,6C,6E,1F,1B,18,45 <1ED6>
1513 DATA 69,6E,67,61,62,65,61,62 <1EBF>
1514 DATA 73,63,68,6C,75,73,73,20 <1EE8>
1515 DATA 6D,69,74,20,20,45,53,43 <1E55>
1516 DATA FF,18,20,53,70,65,69,63 <1E22>
1517 DATA 68,65,72,6E,83,28,6A,2F <1E24>
1518 DATA 20,29,20,3F,20,18,FF,0A <1E01>
1519 DATA 20,20,20,20,20,20,2A,20 <1EAD>
1520 DATA 2A,20,2A,20,20,41,75,73 <1E40>
1521 DATA 64,72,75,63,6B,20,62,65 <1E9C>
1522 DATA 65,6E,64,65,74,2E,20,20 <1ECF>
1523 DATA 2A,20,2A,20,2A,0D,0A,FF <1E37>
1524 DATA 1F,01,06,14,0A,45,69,6E <1EAC>
1525 DATA 20,44,69,72,65,63,74,6F <1E86>
1526 DATA 72,79,20,69,73,74,20,26 <1E84>
1527 DATA 32,30,20,42,79,74,65,73 <1E87>
1528 DATA 20,6C,61,6E,67,2E,0D,0A <1E45>
1529 DATA 42,79,74,65,20,30,30,20 <1EE5>
1530 DATA 3A,86,55,73,65,72,6E,75 <1E5D>
1531 DATA 6D,6D,65,72,20,30,2D,31 <1EED>
1532 DATA 35,20,28,20,45,35,20,67 <1E71>
1533 DATA 65,6C,6F,65,73,63,68,74 <1E6A>
1534 DATA 20,6F,64,65,72,20,6E,69 <1E9A>
1535 DATA 63,68,74,20,62,65,6C,65 <1E65>
1536 DATA 67,74,20,29,2E,0D,0A,42 <1E23>
1537 DATA 79,74,65,20,30,31,20,2D <1E87>
1538 DATA 20,30,38,20,3A,20,46,69 <1E0E>
1539 DATA 6C,65,6E,61,6D,65,20,69 <1E7C>
1540 DATA 6E,20,47,72,6F,73,73,62 <1EFC>
1541 DATA 75,63,68,73,74,61,62,65 <1E22>
1542 DATA 6E,2E,0D,0A,42,79,74,65 <1E90>
1543 DATA 20,30,39,20,2D,20,30,42 <1EC8>
1544 DATA 20,3A,20,46,69,6C,65,6E <1EC7>
1545 DATA 61,6D,65,20,2D,20,45,78 <1E4A>
1546 DATA 74,65,6E,73,69,6F,6E,20 <1E6E>
1547 DATA 69,6E,20,47,72,6F,73,73 <1E03>
1548 DATA 62,75,63,68,73,74,61,62 <1EDB>
1549 DATA 65,6E,2E,0D,0A,42,79,74 <1EC7>
1550 DATA 65,20,30,39,20,3A,86,3E <1E21>
1551 DATA 20,26,38,30,2C,20,42,69 <1EEA>
1552 DATA 74,20,37,3D,31,20,44,61 <1ED3>
1553 DATA 74,65,69,20,69,73,74,20 <1E3B>
1554 DATA 73,63,68,72,65,69,62,67 <1E0A>
1555 DATA 65,73,63,68,75,65,74,7A <1E3A>
1556 DATA 74,2E,0D,0A,42,79,74,65 <1E31>
1557 DATA 20,30,41,20,3A,86,3E,20 <1E47>
1558 DATA 26,38,30,2C,20,42,69,74 <1E24>
1559 DATA 20,37,3D,31,20,44,61,74 <1E4A>
1560 DATA 65,69,20,77,69,72,64,20 <1E47>
1561 DATA 62,65,69,20,44,49,52,20 <1ECD>
1562 DATA 6E,69,63,68,74,20,61,6E <1EF6>
1563 DATA 67,65,7A,65,69,67,74,2E <1E69>
1564 DATA 0D,0A,42,79,74,65,20,30 <1EE0>
1565 DATA 43,20,3A,86,45,78,74,65 <1E2C>
1566 DATA 6E,64,6E,72,2C,20,77,69 <1E7F>
1567 DATA 72,64,20,75,6D,20,31,20 <1E06>
1568 DATA 65,72,68,6F,65,74,20,73 <1E99>
1569 DATA 6F,62,61,6C,64,20,31,36 <1E37>
1570 DATA 20,4B,20,75,65,62,65,72 <1EB0>
1571 DATA 73,63,68,72,69,74,74,65 <1EF2>
1572 DATA 6E,20,77,65,72,64,65,6E <1E07>
1573 DATA 2C,0D,0A,8F,64,61,62,65 <1EA3>
1574 DATA 69,20,65,72,66,6F,6C,67 <1EBC>
1575 DATA 74,20,6A,65,77,65,69,6C <1EDB>
1576 DATA 73,20,65,69,6E,65,20,6E <1E45>
1577 DATA 65,75,65,20,44,69,72,65 <1EC1>
1578 DATA 63,74,6F,72,79,65,69,6E <1EF9>
1579 DATA 74,72,61,67,75,6E,67,2E <1E8E>
1580 DATA 0D,0A,42,79,74,65,20,30 <1E00>
1581 DATA 44,20,2D,20,30,45,20,3A <1E7A>
1582 DATA 20,6E,69,63,68,74,20,62 <1EFC>
1583 DATA 65,6E,75,74,7A,74,2E,0D <1E1A>
1584 DATA 0A,42,79,74,65,20,30,46 <1EC1>
1585 DATA 20,3A,86,41,6E,7A,61,68 <1EE1>
1586 DATA 6C,20,64,65,72,20,52,65 <1E59>
1587 DATA 63,6F,72,64,73,20,28,20 <1E72>
1588 DATA 31,20,52,65,63,6F,72,64 <1E2C>
1589 DATA 20,3D,20,31,32,38,20,42 <1E71>
1590 DATA 79,74,65,73,20,29,2E,0D <1E3D>
1591 DATA 0A,42,79,74,65,20,31,30 <1ECF>
1592 DATA 20,2D,20,31,46,20,3A,20 <1EAA>
1593 DATA 42,6C,6F,63,6B,6E,75,6D <1E92>
1594 DATA 6D,65,72,20,28,20,31,20 <1E7B>
1595 DATA 42,6C,6F,63,6B,20,3D,20 <1E50>
1596 DATA 32,20,53,65,6B,74,6F,72 <1E66>
1597 DATA 65,6E,20,3D,20,38,20,52 <1EC8>
1598 DATA 65,63,6F,72,64,73,20,29 <1EA4>
1599 DATA 2E,0D,0A,42,79,74,65,20 <1E45>
1600 DATA 31,30,20,3A,86,42,6C,6F <1EDB>
1601 DATA 63,6B,61,64,72,65,73,73 <1E7E>
1602 DATA 65,20,64,65,73,20,46,69 <1EFF>
1603 DATA 6C,65,68,65,61,64,65,72 <1E40>
1604 DATA 2E,02,FF,1F,01,09,14,0A <1EE6>
1605 DATA 44,61,74,65,6E,20,64,65 <1E8E>
1606 DATA 73,20,46,69,6C,65,68,65 <1E3D>
1607 DATA 61,64,65,72,73,2E,0D,0A <1E2F>
1608 DATA 42,79,74,65,20,30,30,20 <1E85>
1609 DATA 2D,20,30,42,20,3A,20,57 <1E25>
1610 DATA 69,65,20,62,65,69,20,44 <1E5A>
1611 DATA 69,72,65,63,74,6F,72,79 <1E23>
1612 DATA 65,69,6E,74,72,61,67,75 <1E35>
1613 DATA 6E,67,2E,0D,0A,42,79,74 <1EEF>
1614 DATA 65,20,31,32,86,3A,20,30 <1EE9>
1615 DATA 30,20,3D,20,42,41,53,49 <1EAA>
1616 DATA 43,20,50,72,6F,67,72,61 <1ED8>
1617 DATA 6D,6D,1F,10,0D,30,31,20 <1EFA>
1618 DATA 3D,20,67,65,73,63,68,75 <1E0C>
```

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.

SCHNEIDER CPC-WELT
ANZEIGENABTLG.
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

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

PROGRAMMSERVICE

Achtung! Preis-Senkung!!!

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

☐ Diskette (30 DM)
Name _____
Vorname _____
Straße/Hausnr. _____
PLZ/Ort _____
Ich bezahle:
☐ per beiliegendem Verrechnungsscheck / Bargeld
☐ bargeldlos per Bankeinzug von meinem Konto (nur möglich in der Bundesrepublik!)
bei (Bank) und Ort _____
Kontonummer _____
Bankleitzahl _____
(steht auf jedem Kontoauszug)
Verkaufsbedingungen: Lieferung nur gegen Vorkasse oder Bankabbuchung. Keine Nachnahme.
Umtausch bei Nichtfunktionieren.

Unterschrift _____ SPI
Bitte ausschneiden und einsenden an

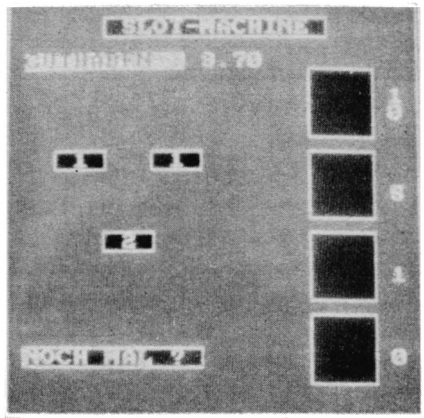
CPC-WELT
SOFTWARE-SERVICE SP
POSTFACH 1161
D-8044 UNTERSCHLEISSHEIM

Jede Diskette nur noch 30 DM! Sie sparen 10 DM

```
1619 DATA 65,74,7A,74,65,73,20,42 <1E6A>
1620 DATA 41,53,49,43,20,50,72,6F <1ED2>
1621 DATA 67,72,61,6D,6D,1F,10,0E <1E9B>
1622 DATA 30,32,20,3D,20,42,69,6E <1EAF>
1623 DATA 61,65,72,20,50,72,6F,67 <1ECB>
1624 DATA 72,61,6D,6D,0D,0A,42,79 <1EDE>
1625 DATA 74,65,20,31,36,20,2B,20 <1E25>
1626 DATA 31,35,20,3A,20,4C,61,64 <1E00>
1627 DATA 65,61,64,72,65,73,73,65 <1E85>
1628 DATA 0D,0A,42,79,74,65,20,31 <1E65>
1629 DATA 39,20,2B,20,31,38,20,3A <1E24>
1630 DATA 20,4C,61,65,6E,67,65,0D <1E3B>
1631 DATA 0A,42,79,74,65,20,30,42 <1E0E>
1632 DATA 20,2B,20,30,41,20,3A,20 <1E9E>
1633 DATA 53,74,61,72,74,61,64,72 <1E82>
1634 DATA 65,73,73,65,20,77,65,6E <1E77>
1635 DATA 6E,20,42,79,74,65,20,31 <1E00>
1636 DATA 32,20,3D,20,30,32,0D,0A <1ECB>
1637 DATA 42,79,74,65,20,34,31,20 <1ED3>
1638 DATA 2B,20,34,30,20,3A,20,4C <1EA5>
1639 DATA 61,65,6E,67,65,0D,0A,42 <1E3F>
1640 DATA 79,74,65,20,34,34,20,2B <1EE8>
1641 DATA 20,34,33,20,3A,20,50,72 <1E4C>
1642 DATA 75,65,66,73,75,6D,6D,65 <1EB2>
1643 DATA 20,64,65,72,20,42,79,74 <1ED0>
1644 DATA 65,73,20,30,30,20,2D,20 <1E19>
1645 DATA 34,32,0D,0A,42,79,74,65 <1E06>
1646 DATA 20,34,35,20,2D,20,37,46 <1E9A>
1647 DATA 20,3A,20,55,6E,77,69,63 <1E98>
1648 DATA 68,74,69,67,2E,0D,0A,0A <1E48>
1649 DATA 41,6B,74,75,65,6C,6C,65 <1E8F>
1650 DATA 20,50,72,75,65,66,73,75 <1EAF>
1651 DATA 6D,6D,65,20,64,65,72,20 <1E61>
1652 DATA 42,79,74,65,73,20,30,30 <1EAA>
1653 DATA 20,2D,20,34,32,20,3D,20 <1E3A>
1654 DATA 18,20,FF,20,18,0D,0A,77 <1E49>
1655 DATA 65,6E,6E,20,67,6C,65,69 <1ED2>
1656 DATA 63,68,20,64,61,6E,6E,20 <1E8E>
1657 DATA 42,41,53,49,43,2D,20,6F <1EB9>
1658 DATA 64,65,72,20,42,49,4E,41 <1E66>
1659 DATA 45,52,20,50,72,6F,67,72 <1EE9>
1660 DATA 61,6D,6D,0D,0A,73,6F,6E <1E37>
1661 DATA 73,74,20,41,53,43,49,49 <1EDE>
1662 DATA 2D,2C,20,43,50,2F,4D,20 <1E4E>
1663 DATA 50,72,6F,67,72,61,6D,6D <1EF9>
1664 DATA 20,6F,64,65,72,20,44,41 <1EDD>
1665 DATA 54,45,49,2E,02,FF,82,42 <1E3B>
1666 DATA 4C,4F,43,4B,20,26,FF,82 <1E4D>
1667 DATA 52,65,63,6F,72,64,74,65 <1EB4>
1668 DATA 69,6C,65,20,FF,31,2B,32 <1E2F>
1669 DATA FF,33,2B,34,FF,35,2B,36 <1E63>
1670 DATA FF,37,2B,38,FF,1F,0F,0C <1EB5>
1671 DATA 2A,20,2A,20,2A,20,44,69 <1E1E>
1672 DATA 73,6B,65,74,74,65,20,66 <1E67>
1673 DATA 65,68,6C,74,20,2F,20,4C <1ED4>
1674 DATA 61,75,66,77,65,72,6B,20 <1EEB>
1675 DATA 6E,69,63,68,74,20,62,65 <1EB9>
1676 DATA 72,65,69,74,20,2A,20,2A <1EBC>
1677 DATA 20,2A,07,FF,1F,13,0C,2A <1E0C>
1678 DATA 20,2A,20,2A,20,44,69,73 <1E5C>
1679 DATA 6B,65,74,74,65,20,69,73 <1E4D>
1680 DATA 74,20,73,63,68,72,65,69 <1E0B>
1681 DATA 62,67,65,73,63,68,75,65 <1E7A>
1682 DATA 74,7A,74,20,2A,20,2A,20 <1E96>
1683 DATA 2A,07,FF,1F,13,0C,2A,20 <1EF2>
1684 DATA 2A,20,2A,20,44,69,73,6B <1EBE>
1685 DATA 65,74,74,65,20,69,73,74 <1EEB>
1686 DATA 20,6E,69,63,68,74,20,66 <1EDD>
1687 DATA 6F,72,6D,61,74,69,65,72 <1ECB>
1688 DATA 74,20,2A,20,2A,20,2A,07 <1EA6>
1689 DATA FF,FF,38,30,40,29,44,26 <1E1A>
1690 DATA 46,23,49,21,4A,1F,4C,1E <1EA7>
1691 DATA 4D,1D,4E,1C,4F,1B,4F,1B <1EF5>
1692 DATA 50,1A,50,1A,37,32,3A,2F <1EE0>
1693 DATA 3C,2D,3D,2C,3E,2B,F8,F4 <1E65>
1694 DATA FB,F1,FD,EF,FE,EE,7A,76 <1E73>
1695 DATA 7D,73,7F,71,80,70,8E,8C <1E7A>
1696 DATA 91,89,92,88,DE,DC,E1,D9 <1EDF>
1697 DATA E2,D8,37,31,3E,2B,41,28 <1E1A>
1698 DATA 43,26,45,25,46,24,47,23 <1E1C>
1699 DATA 47,23,00,20,00          <155B>
```

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("?JjNn",INKEY$)+1 GOT
O 250,250,160,160,260,260          (2ECE)
260 MODE 1:PEN 1:PRINT"Ihr letztes
 Guthaben :";USING"##.##";geld:CAL
L &BB06:/BASIC                     (4221)
270 MODE 1:PEN 1:PRINT"Ihr Guthabe
n sank unter 30 Pf => E N D E":PRI
NT:PRINT"Moechten Sie ein neues Sp
iel beginnen ?"                    (63EA)
280 ON INSTR("?JjNn",INKEY$)+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.

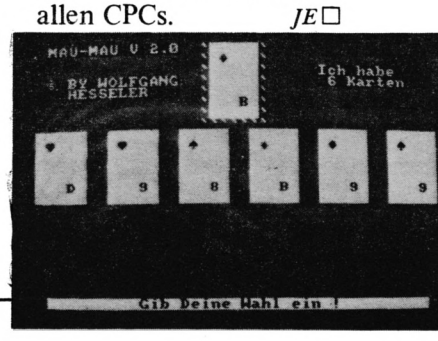
Bei der abzulegenden Karte wird zuerst die Farbe eingegeben:

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*□



```
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)
```

```
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>
```

Leg die Leitung

Dieses Programm unseres Autors Udo Ziese ist ein Spiel für zwei Spieler und läuft auf allen CPC's.

Beide Spieler müssen versuchen, so schnell wie möglich eine Wasserleitung zwischen den beiden Rohrenden auf der jeweiligen Bildschirmhälfte zu verlegen.

Dazu haben Sie verschiedene Rohrteile zur Verfügung: Die schwarzen Rohrteile sind die normalen. Weiße Rohrteile sind unzerstörbar. Wählen Sie jedoch ein rotes Teil aus und Ihr Gegenspieler hat das gleiche Stück als letztes verlegt, so wird dieses wieder zerstört, falls es schwarz war. So können Sie immer wieder die letzten Teile Ihres Gegners zerstören.

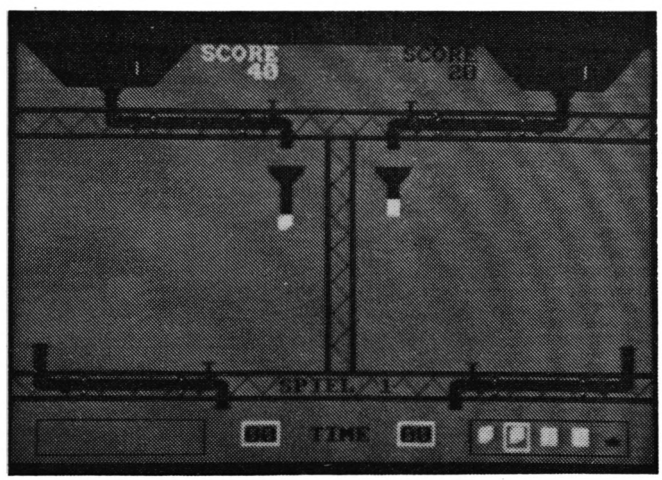
Die zur Verfügung stehenden Rohrteile werden am unteren Bildschirmrand eingeblendet. Falls ein Rohrteil nicht paßt, wird es durch ein anderes ersetzt. Haben Sie ein T-Stück angelegt, so müssen Sie, bevor Sie weiter Ihre Leitung verlegen, erst das überflüssige Rohrende mit einer Kappe verschließen (die Stelle ist durch einen Punkt gekennzeichnet).

Haben Sie schließlich den Tank mit dem Wasserhahn verbunden, so wird das Wasser laut rauschend fließen.

Um das Spiel zu erschweren, können Sie die Rohrauswahl mit einem Zeitlimit versehen. Ist die Zeit abgelaufen und Sie haben nicht gezogen, so ist der Gegenspieler an der Reihe. Durch Drücken der Taste "Z" wird die Uhr aktiviert und Time leuchtet weiß. Nun können Sie die Uhr mit der Taste "S" stellen. Mit Enter beenden Sie die Zeiteingabe. Sie können jederzeit durch "Z" das Zeitspiel wieder verlassen.

Sie bekommen für jede Gerade 10, für eine Biegung 20, ein T-Stück 30 und eine Kappe 50 Punkte. Der Sieger erhält einen Bonus von 200 Punkten. Für jedes Stück, das zerstört wird, werden dem Gegenspieler 20 Punkte abgezogen.

Die Steuerung erfolgt über 2 Joysticks oder über die Tastatur. Als Tasten stehen dem Spieler 1 (linke Hälfte) die obere Zahlenreihe zur Verfügung. Der zweite Spieler (rechte Hälfte) bedient den Zifernblock. Dabei gilt jeweils 1=links, 2=rechts, 5=Rohrteil setzen. *(JE)*




```
100 '*****************************  (234A)
110 '*     LEG DIE LEITUNG       *  (2360)
120 '*           VON             *  (2328)
130 '*        UDO ZIESE          *  (236F)
140 '*           FUER            *  (23F9)
150 '*    SCHNEIDER CPC-WELT     *  (23F0)
160 '*     CPC 464/664/6128   JE*  (2391)
170 '*****************************  (23D6)
180 '                                (0772)
190 '*****************************  (23FE)
200 '*    INITIALISIERUNG        *  (23B9)
210 '*****************************  (2327)
220 '                                (07C2)
230 CLEAR:MODE 1:INK 0,18:INK 1,26
:INK 2,6:INK 3,0:RANDOMIZE TIME    (2366)
240 PEN 1:PRINT CHR$(7);:LOCATE 13
,11:PRINT">> BITTE WARTEN <<":PEN
3:LOCATE 15,13:PRINT"ca. 6 Sekunde
n":FOR pau=1 TO 1500:NEXT          (5843)
250 DEFINT a-z:POKE 47219,2:DIM ap
o(540),bpo(540)                    (2DEB)
260 SYMBOL AFTER 220               (0933)
270 SYMBOL 232,126,125,127,127,127
```

```
,63,15,0:SYMBOL 231,126,255,126,12
6,126,126,126,126:SYMBOL 230,0,15,
63,127,127,127,125,126:SYMBOL 229,
126,126,126,126,126,126,126,255:SY
MBOL 228,126,126,126,126,126,126,1
26,126                                  <8EE8>
280 SYMBOL 238,63,31,31,31,15,7,3,
1:SYMBOL 237,252,248,248,248,240,2
24,192,128:SYMBOL 236,255,126,126,
126,126,126,255,126:SYMBOL 235,255
,126,126,126,126,126,126,126:SYMBO
L 234,255,255,255,255,255,255,255,
255:SYMBOL 233,128,255,255,255,255
,255,255,128                            <A9DA>
290 SYMBOL 239,24,24,0,0,0,0,0,0:S
YMBOL 240,24,60,60,24,0,0,0,0:SYMB
OL 241,24,60,60,60,24,0,0,0:SYMBOL
 242,24,60,126,60,24,0,0,0:SYMBOL
243,24,24,60,126,60,24,0,0:SYMBOL
244,24,24,60,126,126,60,24,0:SYMBO
L 245,24,24,60,126,255,255,126,60 <B5BF>
300 SYMBOL 246,24,24,60,126,255,25
5,255,60:SYMBOL 247,24,24,60,126,2
55,255,255,126:SYMBOL 254,64,36,18
,10,37,11,33,255                        <584A>
310 SYMBOL 224,0,255,255,255,255,2
55,255,0:SYMBOL 225,2,255,255,255,
255,255,255,2:SYMBOL 252,158,255,2
55,255,255,255,255,128:SYMBOL 253,
0,0,0,0,24,255,126,126:SYMBOL 227,
126,126,126,126,126,126,255,126:SY
MBOL 220,2,36,72,80,164,208,132,25
5                                       <A3C9>
320 SYMBOL 255,0,63,12,12,12,12,12
,12:SYMBOL 226,0,240,252,254,254,2
54,190,126:SYMBOL 249,126,190,254,
254,254,252,240,0:SYMBOL 223,126,1
26,255,24,0,0,0,0:SYMBOL 222,32,22
4,224,240,240,224,224,32:SYMBOL 22
1,4,7,7,15,15,7,7,4                     <9F03>
330 GOSUB 340:GOTO 350                  <0E2F>
340 FOR i=25 TO 0 STEP-1:SOUND 1,0
,4,13,0,0,10:CALL &BD19:OUT &BC00,
6:OUT &BD00,i:FOR pau=1 TO 30:NEXT
:NEXT:OUT &BC00,1:OUT &BD00,0:RETU
RN                                      <5A5E>
350 CLS:FOR y=336 TO 340 STEP 2:MO
VE 0,y:DRAW 638,y,2:MOVE 0,y-24:DR
AW 638,y-24:NEXT:FOR x=0 TO 640 ST
EP 32:MOVE x,334:DRAW x+16,318:MOV
E x+16,318:DRAW x+32,334:NEXT           <8621>
360 FOR y=80 TO 84 STEP 2:MOVE 0,y
:DRAW 638,y,2:MOVE 0,y-24:DRAW 638
,y-24:NEXT:FOR x=0 TO 640 STEP 32:
MOVE x,78:DRAW x+16,62:MOVE x+16,6
2:DRAW x+32,78:NEXT                     <7E07>
370 FOR x=304 TO 308:MOVE x,310:DR
AW x,86,2:MOVE x+26,310:DRAW x+26,
86:NEXT:y1=22:FOR y=110 TO 308 STE
P 32:MOVE 310,y:DRAW 334,y-24:NEXT
:FOR y=110 TO 270 STEP 32:MOVE 310
,y-2:DRAW 334,y+22:NEXT                 <9F5B>
380 PRINT CHR$(22)CHR$(1)               <1131>
390 PEN 3:LOCATE 6,4:PRINT CHR$(23
1):LOCATE 6,5:PRINT CHR$(232)CHR$(
225)CHR$(233);STRING$(7,CHR$(224))
CHR$(252)CHR$(226):LOCATE 17,6:PRI
NT CHR$(227)                            <5442>
400 PEN 3:LOCATE 2,19:PRINT CHR$(2
31):LOCATE 2,20:PRINT CHR$(228):LO
CATE 2,21:PRINT CHR$(232)CHR$(225)
CHR$(233);STRING$(7,CHR$(224))CHR$
(252)CHR$(226):LOCATE 13,22:PRINT
CHR$(227)                               <6593>
410 PEN 3:LOCATE 29,20:PRINT CHR$(
255):LOCATE 28,22:PRINT CHR$(227):
LOCATE 28,21:PRINT CHR$(230)CHR$(2
52)CHR$(233);STRING$(6,CHR$(224))C
HR$(225)CHR$(233)CHR$(249):LOCATE
39,20:PRINT CHR$(228):LOCATE 39,19
:PRINT CHR$(231)                        <7D5C>
420 PEN 3:LOCATE 25,4:PRINT CHR$(2
55):LOCATE 24,6:PRINT CHR$(227):LO
CATE 24,5:PRINT CHR$(230)CHR$(252)
CHR$(233);STRING$(6,CHR$(224))CHR$
(225)CHR$(233)CHR$(249):LOCATE 35,
4:PRINT CHR$(231)                       <6AE8>
430 LOCATE 16,4:PRINT CHR$(255):LO
CATE 12,20:PRINT CHR$(255):PEN 3:L
OCATE 16,8:PRINT CHR$(238)CHR$(143
)CHR$(237):LOCATE 17,9:PRINT CHR$(
236):LOCATE 23,8:PRINT CHR$(238)CH
R$(234)CHR$(237):LOCATE 24,9:PRINT
 CHR$(236)                              <741D>
440 i=0:FOR y=360 TO 398 STEP 2:i=
i+2:MOVE 510-i,y:DRAW 590+i,y,2:NE
XT:i=0:FOR y=358 TO 352 STEP-2:i=i
+2:MOVE 538+i,y:DRAW 564-i,y,3:NEX
T                                       <8F0D>
450 i=0:FOR y=360 TO 398 STEP 2:i=
i+2:MOVE 46-i,y:DRAW 126+i,y,2:NEX
T:i=0:FOR y=358 TO 352 STEP-2:i=i+
2:MOVE 74+i,y:DRAW 100-i,y,3:NEXT       <8B69>
460 PEN 1:LOCATE 12,1:PRINT"SCORE"
:PEN 3:LOCATE 25,1:PRINT"SCORE":PL
OT 366,64,0:PLOT 370,64,0:PLOT-2,0
,3                                      <4484>
470 TAG:MOVE 256,76:PRINT"SPIEL";:
MOVE 360,76:PRINT"1";:TAGOFF            <280F>
480 WINDOW#1,15,16,24,24:PAPER#1,0
:PEN#1,3:WINDOW#2,25,26,24,24:PAPE
R#2,0:PEN#2,3                           <3C40>
490 CLS#1:CLS#2:MOVE 220,34:DRAWR
36,0,1:DRAWR 0,-20:DRAWR-36,0:DRAW
R 0,20:MOVE 380,34:DRAWR 36,0:DRAW
R 0,-20:DRAWR-36,0:DRAWR 0,20           <512F>
```

```
500 PRINT#1,"00";:PRINT#2,"00";:LO
CATE 19,24:PRINT"TIME"                (275E)
510 FOR x=126 TO 130:MOVE x,334:DR
AW x,318,2:MOVE x+90,334:DRAW x+90
,318:MOVE x+352,78:DRAW x+352,62:M
OVE x+442,78:DRAW x+442,62:MOVE x+
294,334:DRAW x+294,318:MOVE x+382,
334:DRAW x+382,318:MOVE x-64,78:DR
AW x-64,62:MOVE x+30,78:DRAW x+30,
62:NEXT                               (DEC5)
520 MOVE 110,370:DRAW 110,388,3:DR
AW 114,388:DRAW 114,370:DRAW 110,3
70:MOVE 112,372:DRAW 112,382,1:PLO
T 112,384,3:PLOT 112,386              (5266)
530 PLOT 128,332,1:PLOT 128,320:PL
OT 218,332:PLOT 218,320:PLOT 480,7
6:PLOT 480,64:PLOT 570,76:PLOT 570
,64:PLOT 422,332:PLOT 422,320:PLOT
 510,332:PLOT 510,320:PLOT 64,76:P
LOT 64,64:PLOT 158,76:PLOT 158,64 (860D)
540 MOVE 572,370:DRAW 572,388,3:DR
AW 576,388:DRAW 576,370:DRAW 572,3
70:MOVE 574,372:DRAW 574,386:MOVE
574,372:DRAW 574,382,1:PRINT CHR$(
22)+CHR$(0);                          (6867)
550 x=3:y=24:FOR z=1 TO 5:GOSUB 14
10:a$(z)=a$:fz(z)=f:PEN fz(z):LOCA
TE x,y:PRINT a$(z);:x=x+2:NEXT        (76FB)
560 x=30:y=24:FOR z1=1 TO 5:GOSUB
1410:b$(z1)=a$:ffz(z1)=f:PEN ffz(z
1):LOCATE x,y:PRINT b$(z1);:x=x+2:
NEXT                                  (7E9B)
570 MOVE 22,38:DRAWR 160,0,2:DRAWR
 0,-32:DRAWR-160,0:DRAWR 0,32:MOVE
 454,38:DRAWR 160,0:DRAWR 0,-32:DR
AWR-160,0:DRAWR 0,32                  (4917)
580 ascore=PEEK(47220)+PEEK(47221)
*256:bscore=PEEK(47222)+PEEK(47223
)*256:PEN 1:LOCATE 12,2:PRINT USIN
G"#####";ascore:PEN 3:LOCATE 25,2:
PRINT USING"#####";bscore             (861D)
590 OUT &BC00,1:OUT &BD00,40:FOR i
=0 TO 25:SOUND 1,0,4,13,0,0,10:CAL
L &BD19:OUT &BC00,6:OUT &BD00,i:FO
R pau=1 TO 30:NEXT:NEXT               (5644)
600 '                                 (07BD)
610 '*****************************   (2348)
620 '*  SPIELER LINKES FELD      *   (237B)
630 '*****************************   (2370)
640 '                                 (070C)
650 x=26:y=34:xa=17:ya=10:z=1:ll=1
:l1=1:bb=1:b=1:bb1=0:b1=0:bkap=0:a
kap=0:x1=458:xb=24:yb=10:z1=1:lls=
0:ls=0:lr=0:bbu=0:bu=0:bbo=0:bo=0:
z2=0:z3=0:lu=1:lt=1                   (F094)
660 x3=458:o=0:o1=0:azeit=0:bzeit=
0:zeit=0:zeit1=0:zeit2=0:ti=0:GOSU
B 830                                 (601E)
670 IF JOY(0)=8 OR INKEY(65)=0 THE
N x=x+32:z=z+1:GOSUB 800:GOSUB 830
                                      (39A9)
680 IF JOY(0)=4 OR INKEY(64)=0 THE
N x=x-32:z=z-1:GOSUB 800:GOSUB 830
                                      (3911)
690 IF INKEY(71)=0 THEN GOSUB 1310
                                      (131D)
700 IF JOY(0)=16 AND ti=1 THEN zei
t1=zeit:PRINT#1,USING"##";zeit;:az
eit=0:SOUND 1,80,5,7,3,1:GOTO 760
ELSE IF JOY(0)=16 AND ti=0 THEN SO
UND 1,80,5,7,3,1:GOTO 760             (7E85)
710 IF INKEY(49)=0 AND ti=1 THEN z
eit1=zeit:PRINT#1,USING"##";zeit;:
azeit=0:SOUND 1,80,5,7,3,1:GOTO 76
0 ELSE IF INKEY(49)=0 AND ti=0 THE
N SOUND 1,80,5,7,3,1:GOTO 760         (7EFF)
720 IF azeit>20 THEN GOTO 1340        (16C8)
730 IF ti=1 THEN azeit=azeit+1:GOT
O 670 ELSE GOTO 670                   (2CD9)
740 IF xa=2 AND ya=19 AND akap=0 T
HEN PEN 3:LOCATE 2,19:PRINT CHR$(2
31):SOUND 1,119,10,7:SOUND 1,113,1
0,7:SOUND 1,100,20,7:GOTO 2540        (57EF)
750 LOCATE 3,24:PRINT STRING$(9,"
");:GOSUB 820:g=0:FOR i=1 TO 5:PEN
 ffz(i):LOCATE 30+g,24:PRINT b$(i)
;:g=g+2:NEXT:GOSUB 2070:GOTO 1910 (6BA1)
760 IF fz(z)=2 THEN GOTO 1700         (18BA)
770 IF a$(z)=CHR$(akap)THEN GOSUB
1080:GOTO 740                         (26B8)
780 IF akap=0 THEN GOSUB 840:GOTO
740                                   (19D1)
790 IF akap<>0 THEN GOSUB 1250:GOT
O 740                                 (195F)
800 IF ti=1 THEN azeit=azeit+7        (2127)
810 IF x<26 THEN SOUND 1,80,5,7,3,
1:x=26:z=1 ELSE IF x>154 THEN SOUN
D 1,80,5,7,3,1:x=154:z=5              (5128)
820 MOVE x2,y:DRAWR 24,0,0:DRAWR 0
,-24:DRAWR-24,0:DRAWR 0,24:FOR pau
=1 TO 100:NEXT:RETURN                 (3DD9)
830 MOVE x,y:DRAWR 24,0,1:DRAWR 0,
-24:DRAWR-24,0:DRAWR 0,24:FOR pau=
1 TO 100:NEXT:x2=x:RETURN             (473F)
840 IF ya=18 THEN bu=1 ELSE bu=0      (1F57)
850 IF ya=8 THEN bo=1 ELSE bo=0       (1E0F)
860 IF xa=2 THEN ls=1:bo=1:bu=0 EL
SE ls=0                               (2E16)
870 IF xa>15 AND ya<11 THEN bo=1:b
h=1                                   (27CC)
880 IF TEST((xa*16)-4,(16*(26-ya)-
24))<>0 AND xa>2 AND ya<18 THEN bu
=1                                    (449A)
890 IF TEST((xa*16)-4,(16*(26-ya)+
8))<>0 THEN bo=1                      (3246)
900 IF bu=0 AND b1=0 AND a$(z)=CHR
```

```
$(228)THEN xn=xa:yn=ya+1:ra1=l1:ra
2=b:ra3=b1:l1=1:b=1:as=10:GOTO 122
0'gerade nach unten                    (97E2)
910 IF bo=0 AND b=0 AND a$(z)=CHR$
(228)THEN xn=xa:yn=ya-1:ra1=l1:ra2
=b:ra3=b1:l1=1:b1=1:as=10:GOTO 122
0'gerade nach oben                     (9642)
920 IF ls=0 AND l1=0 AND a$(z)=CHR
$(224)THEN xn=xa-1:yn=ya:ra1=l1:ra
2=b:ra3=b1:b=1:b1=1:as=10:GOTO 122
0'gerade nach links                    (97C5)
930 IF bu=0 AND l1=0 AND a$(z)=CHR
$(230)THEN xn=xa:yn=ya+1:ra1=l1:ra
2=b:ra3=b1:b1=0:b=1:l1=1:as=20:GOT
O 1220'winkel von rechts nach unte
n                                      (AA9D)
940 IF bo=0 AND l1=0 AND a$(z)=CHR
$(232)THEN xn=xa:yn=ya-1:ra1=l1:ra
2=b:ra3=b1:b=0:b1=1:l1=1:as=20:GOT
O 1220'winkel von rechts nach oben
                                       (A9C9)
950 IF b=0 AND a$(z)=CHR$(226)THEN
 xn=xa-1:yn=ya:ra1=l1:ra2=b:ra3=b1
:l1=0:b=1:b1=1:as=20:GOTO 1220'win
kel von unten nach links               (A0DD)
960 IF ls=0 AND b1=0 AND a$(z)=CHR
$(249)THEN xn=xa-1:yn=ya:ra1=l1:ra
2=b:ra3=b1:l1=0:b=1:b1=1:as=20:GOT
O 1220'winkel von oben nach links      (A851)
970 IF a$(z)=CHR$(143)THEN s=INT(R
ND*6):z2=z2+1 ELSE GOTO 1250           (39F9)
980 IF s=0 AND ls=0 AND l1=0 THEN
yn=ya-1:xn=xa:ra1=l1:ra2=b:ra3=b1:
b=1:b1=1:l1=0:akap=253:PLOT(xa*16)
-8,16*(26-ya)+8,1:as=30:GOTO 1220      (AC0C)
990 IF s=1 AND ls=0 AND l1=0 THEN
yn=ya+1:xn=xa:ra1=l1:ra2=b:ra3=b1:
b=1:b1=1:l1=0:akap=223:PLOT(xa*16)
-8,16*(26-ya)-24,1:as=30:GOTO 1220
                                       (AD5D)
1000 IF ls=0 AND b1=0 THEN xn=xa+1
:yn=ya:akap=222:as=30:GOTO 1060        (48DD)
1010 IF s=2 AND xa>3 AND bo=0 AND
l1=0 THEN xn=xa-1:yn=ya:ra1=l1:ra2
=b:ra3=b1:b=0:b1=1:l1=1:akap=221:P
LOT(xa*16)-24,16*(26-ya)-8,1:as=30
:GOTO 1220                             (B598)
1020 IF ls=0 AND b=0 THEN xn=xa+1:
yn=ya:akap=222:as=30:GOTO 1060         (47FE)
1030 IF bu=0 AND b1=0 THEN xn=xa+1
:yn=ya:akap=222:lu=0:GOTO 1060         (47B4)
1040 IF s=3 AND bu=0 AND l1=0 THEN
 xn=xa-1:yn=ya:ra1=l1:ra2=b:ra3=b1
:b1=0:b=1:l1=1:akap=221:PLOT(xa*16
)-24,16*(26-ya)-8,1:as=30:GOTO 122
0                                      (ADD5)
1050 IF akap=0 AND z2<11 THEN GOTO
 970 ELSE z2=0:GOTO 1250               (2BC0)
1060 t8=0:FOR i=(xa*16)+2 TO((xa+1
)*16)-4:t5=TEST(i,16*(26-ya)-8):t6
=TEST(i,16*(26-ya)-2):t7=TEST(i,16
*(26-ya)-14):IF t5<>0 OR t6<>0 OR
t7<>0 THEN t8=1                        (AFC5)
1070 NEXT:IF t8=1 THEN akap=0:GOTO
 1250 ELSE PLOT(xa*16)+8,16*(26-ya
)-8,1:ra1=l1:ra2=b:ra3=b1:l1=0:b=1
:b1=1:GOTO 1220                        (8197)
1080 IF l1=0 AND akap=253 THEN xn=
xa-1:yn=ya+1:kapa=akap:ra1=l1:ra2=
b:ra3=b1:GOSUB 1150:akap=0:as=50:G
OTO 1220                               (8736)
1090 IF l1=0 AND akap=223 THEN xn=
xa-1:yn=ya-1:kapa=akap:ra1=l1:ra2=
b:ra3=b1:GOSUB 1160:akap=0:as=50:G
OTO 1220                               (873D)
1100 IF lu=0 AND akap=222 THEN xn=
xa-1:yn=ya+1:ra1=l1:ra2=b:ra3=b1:l
1=1:b=1:b1=0:lu=1:kapa=akap:GOSUB
1210:akap=0:as=50:GOTO 1220            (A604)
1110 IF l1=0 AND akap=222 THEN xn=
xa-2:yn=ya:kapa=akap:ra1=l1:ra2=b:
ra3=b1:GOSUB 1170:akap=0:as=50:GOT
O 1220                                 (8597)
1120 IF b=0 AND akap=221 THEN xn=x
a+1:yn=ya-1:kapa=akap:ra1=l1:ra2=b
:ra3=b1:GOSUB 1180:akap=0:as=50:GO
TO 1220                                (8607)
1130 IF b1=0 AND akap=221 THEN xn=
xa+1:yn=ya+1:kapa=akap:ra1=l1:ra2=
b:ra3=b1:GOSUB 1200:akap=0:as=50:G
OTO 1220                               (8747)
1140 GOTO 1250                         (0901)
1150 MOVE((xa-1)*16)-10,16*(26-(ya
+1))-12:DRAWR-4,4,1:DRAWR 4,4:DRAW
R-4,-4:DRAWR 12,0:RETURN'Pfeil nac
h links                                (5A8A)
1160 MOVE((xa-1)*16)-10,16*(26-ya)
+4:DRAWR-4,4,1:DRAWR 4,4:DRAWR-4,-
4:DRAWR 12,0:RETURN'Pfeil nach lin
ks                                     (555D)
1170 MOVE((xa-2)*16)-10,16*(26-ya)
-12:DRAWR-4,4,1:DRAWR 4,4:DRAWR-4,
-4:DRAWR 12,0:RETURN'Pfeil nach li
nks                                    (56B0)
1180 MOVE(xa*16)+2,16*(26-ya)+6:DR
AWR 4,4,1:DRAWR 4,-4:DRAWR-4,4:DRA
WR 0,-10:RETURN'Pfeil nach oben        (4FD5)
1190 MOVE((xa-2)*16)-10,16*(26-ya)
-8:DRAWR-4,4,1:DRAWR 4,4:DRAWR-4,-
4:DRAWR 10,0:RETURN'Pfeil nach lin
ks                                     (557C)
1200 MOVE(xa*16)+10,16*(26-(ya+1))
-8:DRAWR-4,-4,1:DRAWR-4,4:DRAWR 4,
-4:DRAWR 0,10:RETURN'Pfeil nach un
ten                                    (5622)
1210 MOVE((xa-1)*16)-6,16*(26-(ya+
```

```
1))-8:DRAWR-4,-4,1:DRAWR-4,4:DRAWR
 4,-4:DRAWR 0,10:RETURN'Pfeil nach
 unten                                  (5933)
1220 IF fz(z)=1 THEN as=as+10           (2221)
1230 PEN fz(z):LOCATE xa,ya:PRINT
a$(z);:xak=xa:yak=ya:amerk$=a$(z):
fmer=fz(z):ap=apo(o):apo(o)=xa:apo
(o+1)=ya:apo(o+2)=ASC(a$(z)):o=o+3
:xa=xn:ya=yn                            (EA59)
1240 FOR pau=1 TO 200:NEXT:GOSUB 1
410:a$(z)=a$:fz(z)=f:PEN fz(z):LOC
ATE(x+22)/16,24:PRINT a$(z);:ascor
e=ascore+as:SOUND 1,120,12,7:PEN 1
:LOCATE 12,2:PRINT USING"#####";as
core:RETURN                             (A86A)
1250 FOR pau=1 TO 200:NEXT:GOSUB 1
410:a$(z)=a$:fz(z)=f:PEN fz(z):LOC
ATE(x+22)/16,24:PRINT a$(z);:RETUR
N                                       (6669)
1260 '                                  (07E6)
1270 '*****************************
                                        (2373)
1280 '*    ZEITEINSTELLUNG        *
                                        (237D)
1290 '*****************************
                                        (239C)
1300 '                                  (0738)
1310 IF ti=0 THEN PEN 1:LOCATE 19,
24:PRINT"TIME":ti=1:zeit=0:GOTO 13
20 ELSE PEN 3:LOCATE 19,24:PRINT"T
IME":PRINT#1,"00";:PRINT#2,"00";:t
i=0:zeit=0:zeit1=0:zeit2=0:azeit=0
:bzeit=0:FOR pau=1 TO 200:NEXT:RET
URN                                     (ACE7)
1320 IF INKEY(60)=0 THEN zeit=zeit
+1:SOUND 129,30,5,15:PRINT#1,USING
"##";zeit;:PRINT#2,USING"##";zeit;
:zeit1=zeit:zeit2=zeit:FOR pau=1 T
O 200:NEXT                              (83DD)
1330 IF INKEY(18)=0 THEN RETURN EL
SE GOTO 1320                            (1605)
1340 zeit1=zeit1-1:IF zeit1<0 THEN
 FOR pau=500 TO 200 STEP-40:SOUND
1,pau,5,15:NEXT:zeit1=zeit:PRINT#1
,USING"##";zeit;:GOTO 750 ELSE SOU
ND 129,30,5,13:PRINT#1,USING"##";z
eit1;:azeit=0:GOTO 670                  (A2F3)
1350 zeit2=zeit2-1:IF zeit2<0 THEN
 FOR pau=500 TO 200 STEP-40:SOUND
1,pau,5,15:NEXT:zeit2=zeit:PRINT#2
,USING"##";zeit;:GOTO 1990 ELSE SO
UND 129,30,5,13:PRINT#2,USING"##";
zeit2;:bzeit=0:GOTO 1910                (A26B)
1360 '                                  (07B0)
1370 '*****************************
                                        (233B)
1380 '* ROHRTEILE PER ZUFALL     *
                                        (235B)
1390 '*****************************
                                        (2363)
1400 '                                  (0700)
1410 s4=INT(RND*22)+1:f=INT(RND*10
)                                       (2567)
1420 IF f=3 OR f=6 THEN f=2:GOTO 1
440                                     (1FD2)
1430 IF f=1 OR f=10 THEN f=1 ELSE
f=3                                     (23EC)
1440 ON s4 GOSUB 1460,1470,1480,14
90,1500,1510,1520,1530,1540,1550,1
560,1570,1580,1590,1600,1610,1620,
1630,1640,1650,1660,1670,1680           (675E)
1450 IF a$=CHR$(143)THEN f=1:RETUR
N ELSE RETURN                           (1D84)
1460 a$=CHR$(228):RETURN                (128D)
1470 a$=CHR$(224):RETURN                (1221)
1480 a$=CHR$(230):RETURN                (12F5)
1490 a$=CHR$(253):RETURN                (12EE)
1500 a$=CHR$(228):RETURN                (12DD)
1510 a$=CHR$(230):RETURN                (1232)
1520 a$=CHR$(232):RETURN                (1286)
1530 a$=CHR$(143):RETURN                (128F)
1540 a$=CHR$(224):RETURN                (12AE)
1550 a$=CHR$(222):RETURN                (1282)
1560 a$=CHR$(226):RETURN                (1217)
1570 a$=CHR$(249):RETURN                (1210)
1580 a$=CHR$(228):RETURN                (127F)
1590 a$=CHR$(224):RETURN                (1213)
1600 a$=CHR$(226):RETURN                (1267)
1610 a$=CHR$(223):RETURN                (121B)
1620 a$=CHR$(224):RETURN                (124F)
1630 a$=CHR$(232):RETURN                (1262)
1640 a$=CHR$(228):RETURN                (12F7)
1650 a$=CHR$(221):RETURN                (122B)
1660 a$=CHR$(228):RETURN                (121E)
1670 a$=CHR$(249):RETURN                (12D8)
1680 a$=CHR$(224):RETURN                (12C7)
1690 '                                  (0744)
1700 '*****************************
                                        (23D0)
1710 '*  ROHRTEIL ZERSTOEREN     *
                                        (2326)
1720 '*****************************
                                        (23F8)
1730 '                                  (0794)
1740 IF a$(z)<>bmerk$OR fmerk=1 TH
EN GOSUB 1250:LOCATE 3,24:PRINT ST
RING$(9," ");:GOSUB 820:g=0:FOR i=
1 TO 5:PEN ffz(i):LOCATE 30+g,24:P
RINT b$(i);:g=g+2:NEXT:GOSUB 2070:
GOTO 1910                               (901E)
1750 IF bmerk$=CHR$(221)OR bmerk$=
CHR$(222)OR bmerk$=CHR$(223)OR bme
rk$=CHR$(253)THEN bkap=ASC(bmerk$)
                                        (5A9C)
1760 IF bmerk$=CHR$(226)OR bmerk$=
CHR$(249)THEN rb=0:rb2=1:rb3=1          (3FD3)
```

```
1770 IF bemerk$=CHR$(230)THEN rb=1
:rb2=0:rb3=1                          (301F)
1780 IF bmerk$=CHR$(232)THEN rb=1:
rb2=1:rb3=0                           (2FBE)
1790 PEN 3:LOCATE xbk,ybk:PRINT"?"
;:xb=xbk:yb=ybk:bpo=bp:ll=rb:bb=rb
2:bb1=rb3:GOSUB 1250:LOCATE 3,24:P
RINT STRING$(9," ");:GOSUB 820:g=0
:FOR i=1 TO 5:PEN ffz(i):LOCATE 30
+g,24:PRINT b$(i);:g=g+2:NEXT:GOSU
B 2070:bscore=bscore-20:GOTO 1910 (ED78)
1800 IF b$(z1)<>amerk$OR fmer=1 TH
EN GOSUB 2480:LOCATE 30,24:PRINT S
TRING$(9," ");:GOSUB 2060:g=0:FOR
i=1 TO 5:PEN fz(i):LOCATE 3+g,24:P
RINT a$(i);:g=g+2:NEXT:GOSUB 830:G
OTO 670                               (8F69)
1810 IF amerk$=CHR$(221)OR amerk$=
CHR$(222)OR amerk$=CHR$(223)OR ame
rk$=CHR$(253)THEN akap=ASC(amerk$)
                                      (5A1D)
1820 IF amerk$=CHR$(230)OR amerk$=
CHR$(232)THEN ra1=0:ra2=1:ra3=1       (408E)
1830 IF amerk$=CHR$(249)THEN ra1=1
:ra2=1:ra3=0                          (30A7)
1840 IF amerk$=CHR$(226)THEN ra1=1
:ra2=0:ra3=1                          (300F)
1850 PEN 3:LOCATE xak,yak:PRINT"?"
;:xa=xak:ya=yak:apo=ap:ll=ra1:b=ra
2:b1=ra3:GOSUB 2480:LOCATE 30,24:P
RINT STRING$(9," ");:GOSUB 2060:g=
0:FOR i=1 TO 5:PEN fz(i):LOCATE 3+
g,24:PRINT a$(i);:g=g+2:NEXT:GOSUB
 830:ascore=ascore-20:GOTO 670        (EB7C)
1860 '                                (079A)
1870 '****************************
                                      (2325)
1880 '* SPIELER RECHTES FELD      *
                                      (2339)
1890 '****************************
                                      (234D)
1900 '                                (07EA)
1910 IF JOY(1)=8 OR INKEY(14)=0 TH
EN x1=x1+32:z1=z1+1:GOSUB 2040:GOS
UB 2070                               (3DF8)
1920 IF JOY(1)=4 OR INKEY(13)=0 TH
EN x1=x1-32:z1=z1-1:GOSUB 2040:GOS
UB 2070                               (3DEE)
1930 IF INKEY(71)=0 THEN GOSUB 131
0                                     (13D1)
1940 IF JOY(1)=16 AND ti=1 THEN ze
it2=zeit:PRINT#2,USING"##";zeit;:b
zeit=0:SOUND 1,200,5,7,3,1:GOTO 20
00 ELSE IF JOY(1)=16 AND ti=0 THEN
 SOUND 1,200,5,7,3,1:GOTO 2000       (7EEA)
1950 IF INKEY(12)=0 AND ti=1 THEN
zeit2=zeit:PRINT#2,USING"##";zeit;
:bzeit=0:SOUND 1,200,5,7,3,1:GOTO
2000 ELSE IF INKEY(12)=0 AND ti=0
THEN SOUND 1,200,5,7,3,1:GOTO 2000
                                      (7E3E)
1960 IF bzeit>20 THEN GOTO 1350       (160E)
1970 IF ti=1 THEN bzeit=bzeit+1:GO
TO 1910 ELSE GOTO 1910                (2C52)
1980 IF xb=39 AND yb=19 AND bkap=0
 THEN PEN 3:LOCATE 39,19:PRINT CHR
$(231);:SOUND 1,119,10,7:SOUND 1,1
13,10,7:SOUND 1,100,20,7:GOTO 2550
                                      (5A5B)
1990 LOCATE 30,24:PRINT STRING$(9,
" ");:GOSUB 2060:g=0:FOR i=1 TO 5:
PEN fz(i):LOCATE 3+g,24:PRINT a$(i
):g=g+2:NEXT:GOSUB 830:GOTO 670       (698D)
2000 IF ffz(z1)=2 THEN GOTO 1800      (1A2B)
2010 IF b$(z1)=CHR$(bkap)THEN GOSU
B 2320:GOTO 1980                      (2726)
2020 IF bkap<>0 THEN GOSUB 2480:GO
TO 1980                               (1938)
2030 IF bkap=0 THEN GOSUB 2080:GOT
O 1980                                (19EE)
2040 IF ti=1 THEN bzeit=bzeit+7       (21DF)
2050 IF x1<458 THEN SOUND 1,200,5,
7,3,1:x1=458:z1=1 ELSE IF x1>586 T
HEN SOUND 1,200,5,7,3,1:x1=586:z1=
5                                     (5B2C)
2060 MOVE x3,y:DRAWR 24,0,0:DRAWR
0,-24:DRAWR-24,0:DRAWR 0,24:FOR pa
u=1 TO 100:NEXT:RETURN                (3D9E)
2070 MOVE x1,y:DRAWR 24,0,1:DRAWR
0,-24:DRAWR-24,0:DRAWR 0,24:FOR pa
u=1 TO 100:NEXT:x3=x1:RETURN          (49E9)
2080 IF yb>17 THEN bbu=1 ELSE bbu=
0                                     (2174)
2090 IF yb<9 THEN bbo=1 ELSE bbo=0
                                      (2081)
2100 IF xb=39 THEN lr=1:bbo=1:bbu=
0 ELSE lr=0                           (313B)
2110 IF xb<27 AND yb<11 THEN bbo=1
                                      (20ED)
2120 IF TEST((xb*16)-16,16*(26-yb)
+8)<>0 THEN bbo=1                     (32D1)
2130 IF TEST((xb*16)-16,16*(26-yb)
-24)<>0 THEN bbu=1                    (330C)
2140 IF bbu=0 AND bb1=0 AND b$(z1)
=CHR$(228)THEN xm=xb:ym=yb+1:rb=ll
:rb2=bb:rb3=bb1:ll=1:bb=1:bs=10:GO
TO 2450'gerade nach unten             (9C9D)
2150 IF bbo=0 AND bb=0 AND b$(z1)=
CHR$(228)THEN xm=xb:ym=yb-1:rb=ll:
rb2=bb:rb3=bb1:ll=1:bb1=1:bs=10:GO
TO 2450'gerade nach oben              (9BC1)
2160 IF lr=0 AND ll=0 AND b$(z1)=C
HR$(224)THEN xm=xb+1:ym=yb:rb=ll:r
b2=bb:rb3=bb1:bb=1:bb1=1:bs=10:GOT
O 2450'gerade nach rechts             (9C6B)
2170 IF bb=0 AND b$(z1)=CHR$(230)T
```

```
HEN xm=xb+1:ym=yb:rb=ll:rb2=bb:rb3
=bb1:ll=0:bb=1:bb1=1:bs=20:GOTO 24
50                                  (88BC)
2180 IF lr=0 AND bb1=0 AND b$(z1)=
CHR$(232)THEN xm=xb+1:ym=yb:rb=ll:
rb2=bb:rb3=bb1:ll=0:bb=1:bb1=1:bs=
20:GOTO 2450                        (91F7)
2190 IF bbo=0 AND ll=0 AND b$(z1)=
CHR$(249)THEN xm=xb:ym=yb-1:rb=ll:
rb2=bb:rb3=bb1:ll=1:bb=0:bb1=1:bs=
20:GOTO 2450                        (919A)
2200 IF bbu=0 AND ll=0 AND b$(z1)=
CHR$(226)THEN xm=xb:ym=yb+1:rb=ll:
rb2=bb:rb3=bb1:ll=1:bb=1:bb1=0:bs=
20:GOTO 2450                        (91C4)
2210 IF b$(z1)=CHR$(143)THEN ss=IN
T(RND*6):z3=z3+1 ELSE z3=0:GOTO 24
80                                  (43B6)
2220 IF ss=0 AND lr=0 AND ll=0 THE
N ym=yb-1:xm=xb:rb=ll:rb2=bb:rb3=b
b1:ll=0:bb=1:bb1=1:bkap=253:PLOT(x
b*16)-8,16*(26-yb)+8,1:bs=30:GOTO
2450                                (B084)
2230 IF ss=1 AND lr=0 AND ll=0 THE
N ym=yb+1:xm=xb:rb=ll:rb2=bb:rb3=b
b1:ll=0:bb=1:bb1=1:bkap=223:PLOT(x
b*16)-8,16*(26-yb)-24,1:bs=30:GOTO
 2450                               (B1B0)
2240 IF ss=2 AND xb<38 AND bbo=0 A
ND ll=0 THEN xm=xb+1:ym=yb:rb=ll:r
b2=bb:rb3=bb1:bb=0:bb1=1:ll=1:bkap
=222:PLOT(xb*16)+8,16*(26-yb)-8,1:
bs=30:GOTO 2450                     (BA24)
2250 IF lr=0 AND bb1=0 THEN xm=xb-
1:ym=yb:rb=ll:rb2=bb:rb3=bb1:bkap=
221:bs=30:GOTO 2300                 (7053)
2260 IF bbu=0 AND bb1=0 THEN xm=xb
-1:ym=yb:rb=ll:rb2=bb:rb3=bb1:bkap
=221:lt=0:GOTO 2300                 (7016)
2270 IF ss=3 AND bbu=0 AND ll=0 TH
EN xm=xb+1:ym=yb:rb=ll:rb2=bb:rb3=
bb1:bb1=0:bb=1:ll=1:bkap=222:PLOT(
xb*16)+8,16*(26-yb)-8,1:bs=30:GOTO
 2450                               (B174)
2280 IF lr=0 AND bb=0 THEN xm=xb-1
:ym=yb:rb=ll:rb2=bb:rb3=bb1:bkap=2
21:GOTO 2300                        (6647)
2290 IF bkap=0 AND z3<11 THEN 2210
 ELSE GOTO 2480                     (22D0)
2300 t4=0:FOR i=((xb-2)*16)TO((xb-
1)*16)-4:t1=TEST(i,16*(26-yb)-8):t
2=TEST(i,16*(26-yb)-2):t3=TEST(i,1
6*(26-yb)-14):IF t1<>0 OR t2<>0 OR
 t3<>0 THEN t4=1                    (B13A)
2310 NEXT:IF t4=1 THEN bkap=0:GOTO
 2480 ELSE PLOT(xb*16)-24,16*(26-y
b)-8,1:rb=ll:rb2=bb:rb3=bb1:ll=0:b
b=1:bb1=1:GOTO 2450                 (8520)
2320 IF ll=0 AND bkap=253 THEN xm=
xb+1:ym=yb+1:kapb=bkap:rb=ll:rb2=b
b:rb3=bb1:GOSUB 2390:bkap=0:bs=50:
GOTO 2450                           (88A7)
2330 IF ll=0 AND bkap=223 THEN xm=
xb+1:ym=yb-1:kapb=bkap:rb=ll:rb2=b
b:rb3=bb1:GOSUB 2400:bkap=0:bs=50:
GOTO 2450                           (88AA)
2340 IF bb=0 AND bkap=222 THEN xm=
xb-1:ym=yb-1:kapb=bkap:rb=ll:rb2=b
b:rb3=bb1:GOSUB 2410:bkap=0:bs=50:
GOTO 2450                           (8899)
2350 IF lt=0 AND bkap=221 THEN xm=
xb+1:ym=yb+1:rb=ll:rb2=bb:rb3=bb1:
ll=1:bb=1:bb1=0:lt=1:kapb=bkap:GOS
UB 2440:bkap=0:bs=50:GOTO 2450      (A9C9)
2360 IF ll=0 AND bkap=221 THEN xm=
xb+2:ym=yb:kapb=bkap:rb=ll:rb2=bb:
rb3=bb1:GOSUB 2420:bkap=0:bs=50:GO
TO 2450                             (8612)
2370 IF bb1=0 AND bkap=222 THEN xm
=xb-1:ym=yb+1:kapb=bkap:rb=ll:rb2=
bb:rb3=bb1:GOSUB 2430:bkap=0:bs=50
:GOTO 2450                          (897E)
2380 GOTO 2480                      (0937)
2390 MOVE((xb+1)*16)-10,16*(26-(yb
+1))-4:DRAWR 4,-4,1:DRAWR-4,-4:DRA
WR 4,4:DRAWR-10,0:RETURN'Pfeil nac
h rechts                            (5BFA)
2400 MOVE((xb+1)*16)-10,16*(26-(yb
-1))-4:DRAWR 4,-4,1:DRAWR-4,-4:DRA
WR 4,4:DRAWR-10,0:RETURN'Pfeil nac
h rechts                            (5B8A)
2410 MOVE((xb-1)*16)-14,16*(26-(yb
-1))-10:DRAWR 4,4,1:DRAWR 4,-4:DRA
WR-4,4:DRAWR 0,-10:RETURN'Pfeil na
ch oben                             (59A7)
2420 MOVE((xb+2)*16)-10,16*(26-yb)
-4:DRAWR 4,-4,1:DRAWR-4,-4:DRAWR 4
,4:DRAWR-10,0:RETURN'Pfeil nach re
chts                                (57CF)
2430 MOVE((xb-1)*16)-6,16*(26-(yb+
1))-8:DRAWR-4,-4,1:DRAWR-4,4:DRAWR
 4,-4:DRAWR 0,10:RETURN'Pfeil nach
 unten                              (5900)
2440 MOVE((xb+1)*16)-4,16*(26-(yb+
1))-8:DRAWR-4,-4,1:DRAWR-4,4:DRAWR
 4,-4:DRAWR 0,10:RETURN'Pfeil nach
 unten                              (59BB)
2450 IF ffz(z1)=1 THEN bs=bs+10     (249E)
2460 PEN ffz(z1):LOCATE xb,yb:PRIN
T b$(z1);:xbk=xb:ybk=yb:bmerk$=b$(
z1):fmerk=ffz(z1):bp=bpo(o1):bpo(o
1)=xb:bpo(o1+1)=yb:bpo(o1+2)=ASC(b
$(z1)):o1=o1+3:xb=xm:yb=ym          (F8D5)
2470 FOR pau=1 TO 200:NEXT:GOSUB 1
410:b$(z1)=a$:ffz(z1)=f:PEN ffz(z1
):LOCATE(x1+22)/16,24:PRINT b$(z1)
```

```
;:bscore=bscore+bs:SOUND 1,120,12,
7:PEN 3:LOCATE 25,2:PRINT USING"##
###";bscore:RETURN                    <AFCE>
2480 FOR pau=1 TO 200:NEXT:GOSUB 1
410:b$(z1)=a$:ffz(z1)=f:PEN ffz(z1
):LOCATE(x1+22)/16,24:PRINT b$(z1)
;:RETURN                              <6D30>
2490 '                                <0787>
2500 '****************************
                                      <2314>
2510 '*    LEITUNG FERTIG          *
                                      <23D3>
2520 '****************************
                                      <233C>
2530 '                                <07D7>
2540 pp=0:c2=o1:c3=bscore:c4=o:c6=
ascore+200:c7=1:c8=12:c9=2:d1=302:
d2=288:d3=260:d4=266:d5=270:d6=302
:d7=242:e1=196:e2=202:e3=3:e4=14:e
5=12:e6=112:e8=13:e9=17:aa=1000:a=
-3:GOTO 2560                          <0297>
2550 pp=0:c2=o:c3=ascore:c4=o1:c6=
bscore+200:c7=3:c8=25:c9=2:d1=302:
d2=288:d3=372:d4=378:d5=382:d6=302
:d7=354:e1=436:e2=442:e3=3:e4=29:e
5=27:e6=574:e8=28:e9=24:aa=1000:a=
-3:GOTO 2590                          <0629>
2560 PEN pp:FOR i=0 TO c2-3 STEP 3
:aa=aa+a:SOUND 1,aa,3,7:LOCATE bpo
(i),bpo(i+1):PRINT CHR$(bpo(i+2));
:FOR pau=1 TO 100:NEXT:NEXT           <7E71>
2570 PEN pp:FOR i=0 TO c4-3 STEP 3
:aa=aa+a:SOUND 1,aa,3,7:LOCATE apo
(i),apo(i+1):PRINT CHR$(apo(i+2));
:FOR pau=1 TO 100:NEXT:NEXT:IF pp=
0 THEN pp=3:a=3:GOTO 2560             <9BD6>
2580 GOTO 2620                        <0930>
2590 PEN pp:FOR i=0 TO c2-3 STEP 3
:aa=aa+a:SOUND 1,aa,3,7:LOCATE apo
(i),apo(i+1):PRINT CHR$(apo(i+2));
:FOR pau=1 TO 100:NEXT:NEXT           <7E46>
2600 PEN pp;FOR i=0 TO c4-3 STEP 3
:aa=aa+a:SOUND 1,aa,3,7:LOCATE bpo
(i),bpo(i+1):PRINT CHR$(bpo(i+2));
:FOR pau=1 TO 100:NEXT:NEXT:IF pp=
0 THEN pp=3:a=3:GOTO 2590             <9BB3>
2610 GOTO 2660                        <09A7>
2620 PEN c7:LOCATE c8,c9:PRINT USI
NG"#####";c6:c=FIX(c6/256):c1=c6-(
c*256):POKE 47220,c1:POKE 47221,c     <6D38>
2630 IF bscore>250 THEN c=FIX(bsco
re/256):c1=bscore-(c*256):POKE 472
22,c1:POKE 47223,c:GOTO 2690          <64E7>
2640 IF bscore<260 THEN POKE 47222
,bscore:POKE 47223,0                  <2FC4>
2650 GOTO 2690                        <09E8>
2660 PEN c7:LOCATE c8,c9:PRINT USI
NG"#####";c6:c=FIX(c6/256):c1=c6-(
c*256):POKE 47222,c1:POKE 47223,c     <6D89>
2670 IF ascore>250 THEN c=FIX(asco
re/256):c1=ascore-(c*256):POKE 472
20,c1:POKE 47221,c:GOTO 2690          <640E>
2680 IF ascore<260 THEN POKE 47220
,ascore:POKE 47221,0                  <2F19>
2690 FOR y=44 TO 6 STEP-2:MOVE 22,
y:DRAW 614,y,0:NEXT                   <2946>
2700 SOUND 1,0,628,15,1,0,6:FOR y=
d1 TO d2 STEP-2:MOVE d3,y:DRAW d4,
y,1:FOR pau=1 TO 20:NEXT:NEXT:TAG:
MOVE d5,d6:PRINT CHR$(220);:MOVE d
7,d6:PRINT CHR$(254);:TAGOFF:FOR p
au=1 TO 2500:NEXT                     <968F>
2710 FOR y=46 TO 0 STEP-2:MOVE e1,
y:DRAW e2,y,1:NEXT:PEN e3:LOCATE e
4,25:PRINT CHR$(220):LOCATE e5,25:
PRINT CHR$(254):FOR pau=1 TO 2500:
NEXT                                  <694E>
2720 FOR y=382 TO 372 STEP-2:PLOT
e6,y,3:FOR pau=1 TO 500:NEXT:NEXT     <35A8>
2730 TAG:MOVE d5,d6:PRINT" ";:MOVE
 d7,d6:PRINT" ";:TAGOFF:FOR y=d2 T
O d1 STEP 2:MOVE d3,y:DRAW d4,y,0:
FOR pau=1 TO 20:NEXT:NEXT:FOR pau=
1 TO 2500:NEXT                        <7DE2>
2740 LOCATE e4,25:PRINT" ";:LOCATE
 e5,25:PRINT" ";:FOR y=0 TO 46 STE
P 2:MOVE e1,y:DRAW e2,y,0:NEXT:FOR
 pau=1 TO 1000:NEXT:PLOT 246,398,1
                                      <67D9>
2750 PEN 1:FOR i=239 TO 247:LOCATE
 e8,23:PRINT CHR$(i);:LOCATE e9,7:
PRINT CHR$(i);:FOR pau=1 TO 150:NE
XT:NEXT                               <4DC5>
2760 PEN 3:LOCATE 10,25:PRINT"NOCH
 EIN SPIEL (J/N)?"                    <271F>
2770 IF INKEY(45)=0 THEN 2790         <1250>
2780 IF INKEY(46)=0 THEN CLS:LOCAT
E 14,12:PRINT"S C H A D E...":CALL
 &BB03:END ELSE GOTO 2770             <3686>
2790 GOSUB 340:FOR y=7 TO 8:LOCATE
 1,y:PRINT SPACE$(15);:LOCATE 26,y
:PRINT SPACE$(15);:NEXT:FOR y=10 T
O 18:LOCATE 1,y:PRINT SPACE$(18);:
LOCATE 23,y:PRINT SPACE$(18);:NEXT
:q=PEEK(47219):q$=STR$(q):PLOT-2,0
,3:TAG:MOVE 360,76:PRINT MID$(q$,2
);                                    <A5D5>
2800 TAGOFF:POKE 47219,q+1:LOCATE
3,19:PRINT SPACE$(16);                <2484>
2810 LOCATE 23,19:PRINT SPACE$(16)
;:LOCATE 25,9:PRINT SPACE$(16);:LO
CATE 1,9:PRINT SPACE$(16);:LOCATE
13,23:PRINT" ":LOCATE 28,23:PRINT"
 ":LOCATE 17,7:PRINT" ":LOCATE 24,
7:PRINT" ":LOCATE 10,25:PRINT SPAC
E$(21);:RANDOMIZE TIME:GOTO 490       <781D>
```

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-

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 KByte 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> Zuruecksetzen","<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,
```

```
@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)
```

```
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)
```

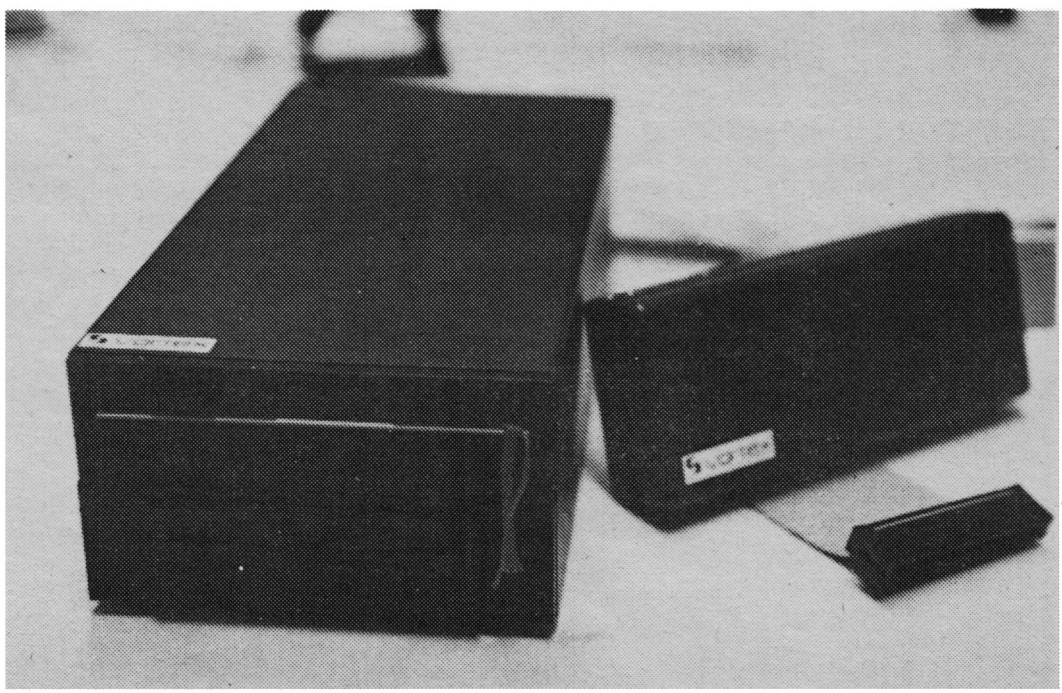
```
100 '*****************************  (2234)
110 '*  ARCHIVATOR-DATALADER   *    (2257)
120 '*   ERZEUGT (ARCH.BIN)    *    (2257)
130 '*          VON            *    (22F5)
140 '*     THOMAS HAUMANN      *    (229E)
150 '*          FUER           *    (2288)
160 '*   SCHNEIDER CPC-WELT     *   (2247)
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)
```

```
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)
```

Vortex-Rom geknackt

Viele Vortex-Anwender interessieren sich auch für das „Innenleben" Ihres Diskettenlaufwerks. Problematisch ist dabei nur, daß Vortex seine ROMs seit einiger Zeit mit trickreichen Verfahren gegen Auslesen schützt. Unser Leser Ludger Papenkort hat eine Methode gefunden, wie man trotzdem an den begehrten Inhalt dieser ROMs kommt. Hier ist sein Bericht über das Leben und Leiden mit der Vortex-Floppy.

Software-Schutz ist ein legitimes Anliegen der Programmersteller, da oft viele arbeitsreiche Monate, wenn nicht sogar, wie bei professionellen Programmen (WordStar, dBase), viele (Mann-)Jahre zwischen der Idee und dem lauffähigen Programm liegen. Private Software-Tüftler begnügen sich häufig mit einem einfachen Listschutz ihres BASIC-Programmes. Andere wiederum verwenden Kopierschutzverfahren verschiedenster Raffinesse.
Wie auch der Software-Schutz geartet ist, es gibt immer ein Mittel (Programm), ihn zu umgehen oder unwirksam zu machen. Den Hackern unter den Lesern sage ich nichts Neues.
Es überraschte mich aber, festzustellen, daß auch Programme, die in ROMs abgelegt sind, geschützt sein können. Ich kam darauf, als ich mit dem Disassembler der Vortex-Floppy F1-X die ROMs meines CPC durchforsten wollte, um zu sehen, was sich da so alles abspielt.
Es ist immer interessant, zu erfahren, wie die Profis programmieren, und mitunter auch sehr lehrreich für eigene Routinen in Maschinensprache.
Da die Erweiterungs-ROMs alle dem gleichen Standard entsprechen müssen, um vom Betriebssystem beim Einschalten des Rechners aktiviert werden zu können, ist das Auffinden bestimmter Routinen recht einfach. Nach den ersten vier Byte eines jeden ROM kommt die Tabelle der RSX-Befehle. Sie besteht aus lauter JP nnnn, wobei nnnn das Sprungziel ist *(Abbildung 1)*.



Kein Geheimnis mehr: Die Vortex-Floppy-Station

Der erste Eintrag in der Sprungtabelle ist kein Jump, sondern die Adresse der Befehlsnamen-Tabelle. Hier finden Sie einen Befehl, zählen seine Position ab und sehen dann in der Sprungtabelle an der gleichen Position die Adresse der gesuchten Routine.
Das BASIC-ROM, das ROM der Schneider Drei-Zoll-Floppy und auch das ROM der Vortex-Speichererweiterung verhielten sich vorschriftsmäßig: Beim Disassemblieren traten keinerlei Probleme auf, sinnvollen Code zu erzeugen. Natürlich mit Ausnahme der Datenbereiche, wie etwa die Tabellen für die Disketten-Formate im Floppy-ROM.
Bei Durchsicht des Vortex-Floppy-ROMs gab's dann aber die große Überraschung: Noch nicht einmal die Sprungtabelle konnte richtig disassembliert werden *(Abbildung 2)*.
Meine erste Vermutung, der Vortex-Disassembler würde die ROM-Nummer erkennen und dann den Code (ebenfalls von der Firma Vortex) verschlüsselt ausgeben, wurde durch Betrachtung mit dem DDTZ entkräftet. Da standen immer noch für eine Sprungtabelle sehr untypische Dinge wie "CALL PE,nnnn", verschiedene "SUB"s und "AND"s oder auch nur ein banales "HALT". Manchmal war sogar das resignierende "???" des DDTZ zu sehen, der mit dem Code auch nichts mehr anzufangen wußte.
Beispiele für einen entschlüsselten und einen nicht-entschlüsselten ROM-Header finden Sie in den *Abbildungen 1 und 2*. Sie werden selbst erkennen, daß sich mit der Originaltabelle nicht viel anfangen läßt.
Darauf setzte bei mir die Denkphase ein und ich schrieb mir die Byte so untereinander, wie sie eigentlich kommen müßten. Parallel dazu die entsprechenden Adressen und die Byte als Binärzahl. In *Abbildung 3* sehen Sie einen Ausschnitt aus der Sprungtabelle, die das System der Vortex-Verschlüsselung zeigt: Zunächst erfordert ein Sprungbefehl als erstes Byte immer ein "C3", was den Op-Code für den Jump darstellt. In der Tabelle fällt auf, daß das Byte, welches eigentlich immer C3 enthalten müßte, zwischen E3, CB und EB wechselt (Marke 1).

Adresse	Inhalt	Bemerkungen
C000	01	ROM-Typ
C001	01	ROM-Markierungs-Nummer
C002	21	ROM-Versions-Nummer
C003	00	ROM-Modifiktions-Ebene
C004	5F E9	Adresse Befehlsnamen-Tabelle
C006	C3 EC C1	Sprungtabellen-Eintrag 0 (Initialisierungs-Routine)
C009	C3 DC C2	Sprungtabellen-Eintrag 1
C00C	C3 A1 B6	usw.
C00F	C3 A4 B6	
usw.		

Abb. 1: Beispiel ROM-Header (Vortex-Floppy entschlüsselt)

Adresse	Inhalt	Bemerkungen
C006	E3	EX (SP),HL
C007	EC C1 C3	CALL PE,C3C1
C00A	DC C2 E3	CALL C,E3C2
C00D	A1	AND C
C00E	D6 E3	SUB E3
usw.		

Abb. 2: ROM-Header (Vortex-Floppy nicht entschlüsselt)

Adresse	Inhalt	1. Byte binär: 7654 3210	2. Byte binär: 7654 3210
C006	E3 EC C1	1110 0011	0000 0110
C009	C3 DC C2	1100 0011	0000 1001
C00C	E3 A1 D6	1110 0011	0000 1100
C00F	E3 A4 D6	1110 0011	0000 1111
C012	CB B2 D6	1100 1011	0001 0010
C015	EB 76 D6	1110 1011	0001 0101
C018	CB 79 D6	1100 1011	0001 1000
C01B	CB 8F D6	1100 1011	0001 1011
usw.			
	Marke 1	Marke 2	
		Marke 3	

Abb. 3: Beginn der Sprungtabelle bei C006 laut Konvention der Firma Amstrad

In der Binärdarstellung ist zu erkennen, daß hierfür nur das dritte beziehungsweise fünfte Bit (von null bis sieben gezählt) verantwortlich sein kann. Noch etwas fällt auf: Wenn das dritte Bit gesetzt ist, also anstatt einer 3 ein B im unteren Nibble des Op-Codes steht, dann ist das vierte Bit in der Adresse auch gesetzt (Marke 2). Das selbe gilt auch für das fünfte Bit des Op-Codes und das zweite Bit der Adresse (Marke 3). Nachdem ich diesen Zusammenhang herausgefunden hatte, baute ich ein Filterprogramm. Es invertierte mir die Bit drei und fünf eines jeden Byte des Codes im ROM entsprechend den Bit vier und zwei des unteren Byte der jeweiligen Adresse. Also:

Adresse unteres Byte, Bit 2 = 1: Bit 5 an der Adresse invertieren;

Adresse unteres Byte, Bit 2 = 0: Bit 5 bleibt wie es ist;

Adresse unteres Byte, Bit 4 = 1: Bit 3 an der Adresse invertieren;

Adresse unteres Byte, Bit 4 = 0: Bit 3 bleibt wie es ist.

Die Sache war also klar. Vortex hat einfach die Adreßleitungen mit den Datenleitungen am ROM entsprechend über Gatter und Inverter verknüpft. Daß das Problem gelöst war, zeigte sich beim DDTZ-Durchlauf des gefilterten Codes: Es kamen funktionierende Sprungtabellen heraus, und auch sonst wurde sinnvoller Code disassembliert.
Wenn Sie jetzt das Filterprogramm suchen sollten, um schnell mal den ROM-Code durchzunudeln, muß ich Sie leider enttäuschen. Bei genauerem Hinsehen stellte sich nämlich die nächste Hürde ein: Die Sprünge der Befehlstabelle gingen zum Teil auf Adressen, die mitten in einem Befehl lagen, zum Beispiel auf das dritte Byte eines Vier-Byte-Befehls. Das ist der Funktion eines Programms nicht gerade zuträglich. Des Rätsels Lösung: Die oben geschilderte Invertierung betrifft nur die Byte, die den Op-Code eines Befehls enthalten.
Alle anderen Byte, die nur Daten wie Sprungadressen oder irgendwelche Konstanten enthalten, machen diese Invertierung nicht mit. Das ist schon an der Tabelle der Befehlsnamen zu sehen, die im Klartext im ROM stehen und für jedermann mittels Hex-Dump zu lesen sind.
Die zusätzlichen Gatter der Firma Vortex befinden sich wahrscheinlich in den kleinen schwarzen Särgen, die man nach guter alter Unsitte der Typaufschrift durch Abschleifen beraubt, und somit in die Anonymität getrieben hat – wenigstens in meinem Floppy-Controller. Aber wie können Sie unterscheiden, ob der Prozessor gerade ein Befehlsbyte oder ein Datenbyte adressiert? Aber genau das meldet der Prozessor seiner Umwelt mittels seiner M1-Steuerleitung.

DER ABLAUF IM PROZESSOR

Diese Steuerleitung gibt Auskunft über den Stand der prozessor-internen Befehlsverarbeitung. Sie hat drei wichtige Phasen: die Holphase (Fetch-Zyklus), die Dekodierphase und die Ausführungsphase.
Die Holphase ist die interessanteste, denn in ihr wird der Op-Code eines Befehls geholt, was mit der M1-Steuerleitung angezeigt wird. Dies ist wichtig für externe Einheiten wie DMA-Bausteine, die einen Interrupt beim Prozessor angefordert haben und ihrerseits einen Befehl auf den Datenbus legen wollen, meistens ein RST (Restart) oder CALL. Der Prozessor meldet den externen Einheiten mit der M1-Steuerleitung und dem IORQ, daß er einen Befehl empfangen will.
In der Dekodierungspha-

Kein Software-Schutz ist gegen Mißbrauch gefeit. Das verständliche Anliegen der Hersteller reizt private Software-Tüftler, den Schutz mit Hilfe eines Programms zu umgehen oder unwirksam zu machen.

se wird der Befehl intern weiterverarbeitet, also dekodiert, und in der Ausführungsphase ausgeführt.
Die Ausführung kann nicht nur darin bestehen, zum Beispiel etwas zu addieren, sondern auch weiter Datenbyte zu holen. Aber für uns ist wichtig: Dieses Holen gehört nicht zur Hol-, sondern zur Ausführungsphase. Mit der M1-Steuerleitung kann also zwischen der Adressierung von Op-Code-Byte und der von Datenbyte unterschieden werden.
Daraus folgt, daß die Vortex-Entwickler nicht nur die Adreßleitungen, sondern auch die M1-Leitung mißbraucht haben, um ihren Code zu schützen. Aber das ist für Sie jetzt kein Problem mehr. Sie brauchen nur noch einen Disassembler zu programmieren, der immer weiß, was für ein Byte er vor sich hat und es entsprechend der oben aufgestellten Regel entschlüsselt oder auch nicht.

Ich begnüge mich mit zwei Listings: einem entschlüsselten für die Op-Codes und einem Original-Hexdump für die Adressen und Daten, die sich nebeneinander auf meinem Schreibtisch meterweise herumräkeln und mich hämisch angrinsen, wenn ich wieder mal vergessen habe, woher der Sprung eigentlich kam, den ich gerade verfolge. *Ludger Papenkort*□

Hinweis der Redaktion: Für Informationen wie in obigem Beitrag übernehmen wir keine Verantwortung. Die Vorschriften des Urheberrechtsgesetzes sind zu beachten.

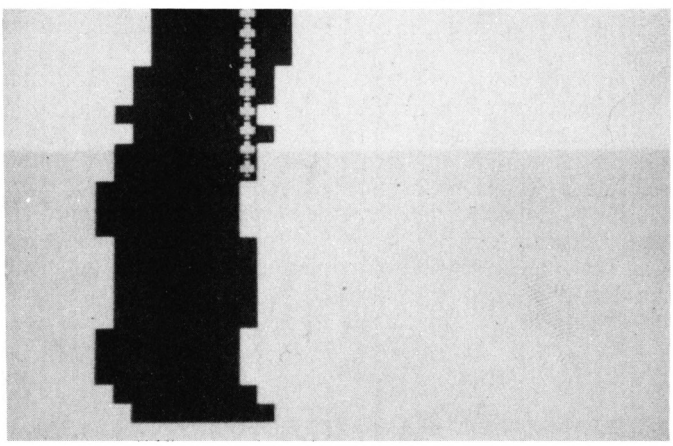
Canyon

Dieses Mini-Programm unseres Autors Martin Sachenbacher besteht aus fünf (!) Programmzeilen. Es hat trotz seiner Kürze alles, was ein gutes Spiel ausmacht: Hohe Geschwindigkeit, einstellbaren Schwierigkeitsgrad und die Vergabe von Punkten.

Gesteuert wird das Programm mit den Cursortasten. Es kommt darauf an, durch eine Schlucht die Spielfigur so zu steuern, daß es zu keiner Kollision mit den Felswänden kommt.

Der Eindruck einer Schlucht wird durch ständiges Scrolling des Bildschirms erreicht. Ein gelungenes Programm, das Sie in drei Minuten abgetippt haben und das dazu noch ausbaufähig ist.

In der Redaktion standen wir vor dem Problem der Honorierung dieses Listings. Da wir normalerweise die Programme nach der Länge des Listings bezahlen, wäre in diesem Fall der Autor schlecht davon gekommen. Spontan wurde beschlossen, daß zusätzlich zum (bescheidenen) Honorar der Autor ein Original-Spiel als „Dankeschön" bekommt. Wir haben das Programm „MGT" ausgewählt und bereits übersandt. Gleichzeitig sollte das Ansporn für die anderen Autoren sein, nicht unbedingt nur auf die Zeilenzahl zu achten, sondern auf Idee und Ausführung. ■ *JE*



```
1 '******************************** <2506>
2 '*            CANYON            * <2595>
3 '*             VON              * <25F6>
4 '*       MARTIN SACHENBACHER     * <2512>
5 '*             FUER             * <2558>
6 '*       SCHNEIDER CPC-WELT      * <2539>
7 '*        CPC 464/664/6128    JE* <255F>
8 '******************************** <2514>
10 DEFINT a-z:PEN 1:INPUT"Breite d
es Canyons ";l:MODE 1:LOCATE 1,25:
PLOT 0,0,1:x=12:s=24:l=l-1:PAPER 2
                                    <5856>
20 x=x+RND*2-1:IF x<2 THEN x=2 ELS
E IF x>39-l THEN x=39-l             <4036>
30 s=s+(INKEY(8)=0)+INKEY(1)+1:WIN
DOW#1,x,x+l,25,25:CLS#1:PRINT       <3F07>
40 IF TEST(s*8+2,216)>0 OR TEST(s*
8+14,216)>0 THEN 50 ELSE TAG:MOVE
s*8,224:PRINT CHR$(226);:TAGOFF:p=
p+1:GOTO 20                         <58DE>
50 PRINT:PAPER 3:PEN 0:PRINT"Punkt
e:"p-13:CALL &BB03:CALL &BB18:RUN <2A0D>
```


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

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

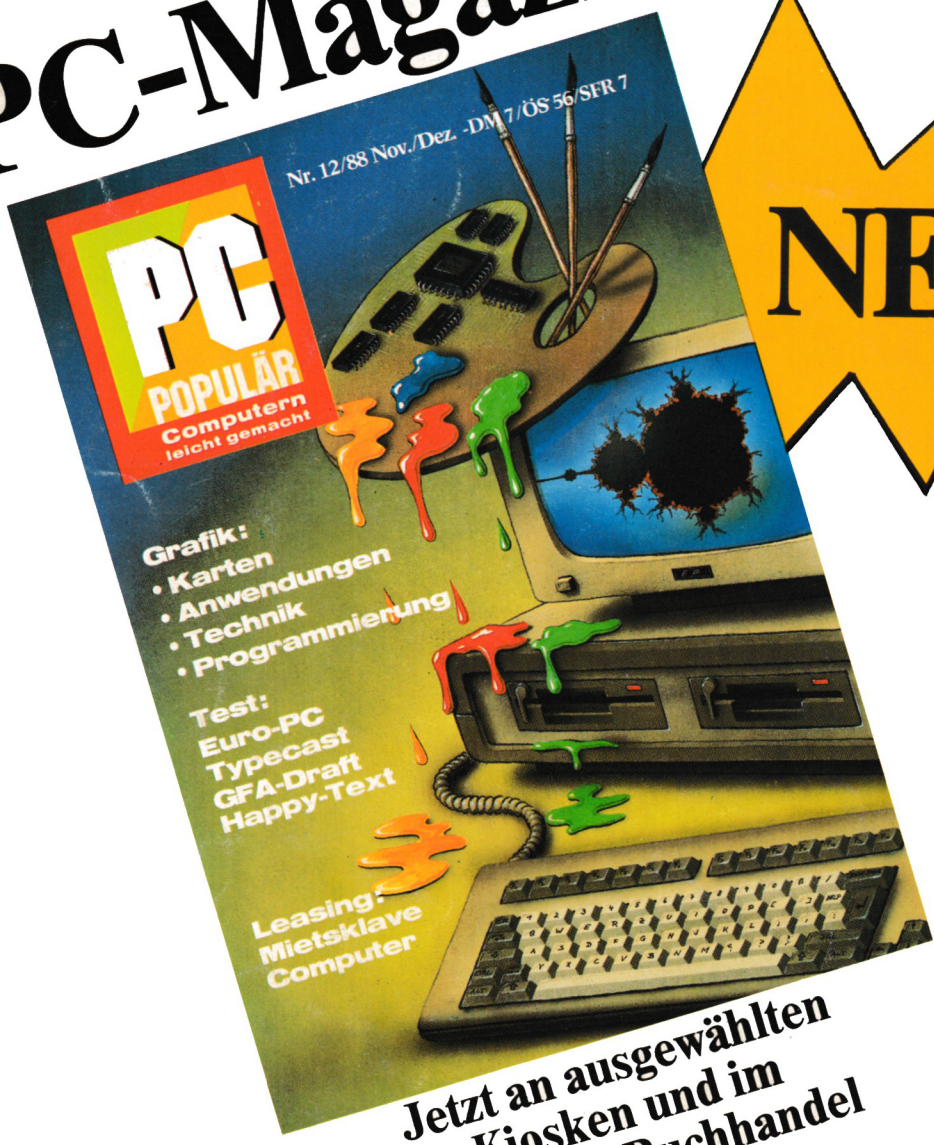
ANSCHRIFT FÜR ALLE VERANTWORTLICHEN: Postfach 1161, 8044 Unterschleißheim, Tel.: 089/1 29 80 11, Telex: 5214428 cav-d

VERTRIEB: Verlagsunion Wiesbaden

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

COMPUTERN LEICHT GEMACHT

Das PC-Magazin



NEU

Jetzt an ausgewählten Kiosken und im Bahnhofs-Buchhandel