DAS ELEKTRONISCHE MAGAZIN 5/86

INPUT 64

Infos · News · Programme · Unterhaltung · Tips

WM '86
Fußball in Mexico
Tabellen aktuell

MultiTape
Kassetten-Zugriff ohne Format-Probleme

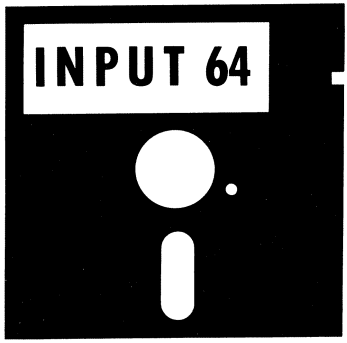
LISP Teil II
Editor, Tracer, Makros und weitere Bausteine

Scrolling
Ein Spiel zum 'durch die Wand gehen'

Spiel
Mathe mit Nico
Tips
ID-Werkstatt

Dokumentation und Bedienungshinweise

INPUT-ein Magazin aus dem Verlag Heinz Heise GmbH, Postfach 61 04 07, 3000 Hannover 61



Ab 4/85 auch auf Diskette-

direkt vom Heise-Verlag, INPUT-Vertrieb,
Postfach 610407, 3000 Hannover 61

für 19.80 DM inkl. Versandkosten + MwSt.-

Nur gegen V-Scheck!

Das Geschenk:

INPUT 64 V . Disk im Sixpack

Die Ausgaben 4/85 bis 9/85 der Disketten-Version bekommen Sie ab sofort zum Paketpreis von 90 DM.

Jetzt bestellen, **24.80 DM** sparen!

Direkt beim Verlag:
(Inclusive Porto und Verpackung) **Nur gegen V-Scheck!**

Verlag Heinz Heise GmbH · Postfach 610407 · 3000 Hannover 61

Kurs komplett

Jetzt als Sampler:
Die Serie BITS & BYTES IM VIDEO-CHIP

Alle Folgen des Kurses auf Kassette und Diskette. Eine grundlegende Einführung in die Programmierung des Video-Chip, mit Exkursen in die Binärarithmetik, Programmiertips und so weiter.

Überarbeitet und um einen Teil zur Multicolor-Grafik erweitert.

Kassette 17.80 DM
(mit SuperTape-Lader und Sicherheitskopie auf der Rückseite)

Diskette 24.80 DM

Direkt beim Versand: (inclusive Porto und Verpackung)

Nur gegen V-Scheck!

Verlag Heinz Heise GmbH · Postfach 610407 · 3000 Hannover 61

INHALTSVERZEICHNIS

INPUT 64

5/86

Leser fragen. . . Seite 3

Das aktuelle Anwenderprogramm:
WM'86 Seite 4

MultiTape Seite 6

Des LISP zweiter Teil:
Macros, Editor, Tracer, Arrays, Sets. . . Seite 7

Im Test:
Star-NLQ-Drucker Seite 18

Scrolling Seite 19

Mathe mit Nico:
Wahrscheinlichkeitsrechnung Seite 21

ID-Werkstatt:
Direct-Sound, Hires-Painter Seite 21

64er-Tips – Die USR-Funktion Seite 22

MPS801-Hardcopy Seite 25

Füll den Kreis Seite 27

Zur c't-Uhr Seite 27

Für Nachzügler und Spätzünder Seite 28

Hinweise zur Bedienung Seite 29

Hinweise für Autoren Seite 29

Vorschau Seite 31

„Liebe 64er-Besitzer(in)"

Das Rennen geht weiter! Hart im Rennen: Programm-Entwickler und -Vertreiber gegen Kopierer-was-das-Zeug-hält.
Wenn man bedenkt, daß die Entwicklung eines mittleren Programms mindestens 100 Stunden in Anspruch nimmt (Fehlschläge nicht eingerechnet), kann man das Interesse an Programm-Schutz wohl nur verstehen. Sicher, auf der anderen Seite stehen die Kosten für Orginal-Software, die so manchen begrenzten Etat überschreiten.

Wie kann man mit der Entwicklung von Software noch Geld verdienen, wenn einem die Programm quasi vom Tisch wegkopiert werden und 'rum' sind, bevor die Produktion und der Vertrieb überhaupt erst in die Wege geleitet wurden? Und von irgendetwas muß ein Software-Experte schließlich auch leben. Die unvermeidliche Folge: Die wirklich guten professionellen Entwickler wenden sich anderen Systemen zu, und gute Software wird immer seltener.

Es sei denn, ein guter Kopierschutz! Doch da kommt sportlicher Ehrgeiz ins Spiel, den Schutz zu 'knacken'. Irgendwann gelingt es irgendwem. Es gibt eben keine 100% Sicherheit (wer das behauptet lügt ganz einfach!). Also ran ans Werk und den nächsten Kopierschutz erfinden und austüfteln.

In diesem Widerspruch befinden wir uns mit jeder INPUT 64 Ausgabe. 170 kByte Software für 14,80 DM bzw. 19,80 DM (Diskette). Auf der einen Seite Software zu einem Preis, der nur durch die hohe Auflage möglich ist, auf der anderen Seite, vorsichtig geschätzt, die vierfache Menge an Raubkopieen unseres Magazins (Wir aus solchen Gründen keine zehn Beihefte für 'gute Freunde' zur Verfügung stellen, wie ein Leser telefonisch anfragte). So müssen wir uns ebenfalls gegen solche Übergriffe schützen, indem wir unseren Kopierschutz weiter verbessern.

Die Leidtragenden sind zuletzt die ehrlichen Anwender, die das vollständige Magazin inklusive Beiheft bevorzugen. Das Durchforsten raubkopierter Software nach Bedienungsmöglichkeiten dauert oft so lange, daß man mit dem Kauf des Orginals im Endeffekt gemessen am Zeitaufwand besser wegkommt. Wer also bisher Programme aus dem Magazin direkt herauskopiert hatte, kann jetzt nicht mehr direkt zugreifen. Natürlich können dann die Programme, die zum Abspeichern vorgesehen sind, auf eigene Datenträger überspielt werden. Von dort können Sie diese dann auf andere kompatible Systeme übertragen.

Viel Spaß mit INPUT 64

Redaktions-Geflüster:

Fragt einer: "Warum werden die Hardcopies auf dem MPS 801 nicht nur 1:1 ausgedruckt?"
Antwort: "Weil die Fernseher unterschiedlich groß sind!"

Auf einen Blick: INPUT 64 - Betriebssystembefehle

Titel abkürzen	CTRL und Q
Hilfsseite aufrufen	CTRL und H
Inhaltsverzeichnis aufrufen	CTRL und I
Farbe für Bildschirm-Hintergrund ändern	CTRL und F
Rahmenfarbe ändern	CTRL und R
Bildschirmausdruck	CTRL und B
Programm sichern	CTRL und S

Laden von Kassette mit LOAD oder SHIFT + RUN/STOP

Laden von Diskette mit LOAD "INPUT*",8,1

Ausführliche Bedienungshinweise finden Sie auf Seite 29

Leser fragen...

INPUT-BASIC' Eprom-Bank

Sie bieten das INPUT-BASIC aus Ausgabe 1/86 auch auf Eprom an. Handelt es sich dabei um ein fertiges Steckmodul, oder muß man da noch löten?
(tel. Anfragen)

Sie kommen dabei um etwas Eigenarbeit nicht herum: Für 49 DM erhalten Sie das komplette INPUT-BASIC auf zwei 2664er-Eproms. Dazu benötigen Sie die in unserem Schwestermagazin c't (Heft 12/85) vorgestellte Eprom-Bank. Die Platine für diese Eprom-Bank ist beim Verlag erhältlich. (18 DM) Zusätzlich benötigen Sie natürlich noch einige Kleinteile (Fassungen für die Eproms, Dill-Schalter etc.). Und, um auf Ihre eingangs gestellte Frage zurückzukommen, etwas Lötzinn und und einen Lötkolben zum Zusammenbau brauchen Sie auch. (d. Red.)

LISP 64's CR

In der Anleitung zu LISP 64 (Ausgabe 4/86) ist häufig von CR die Rede. Was ist damit gemeint?
(tel. Anfrage)

CR steht für "Carriage Return" (wörtlich übersetzt "Wagenrücklauf") und meint die RETURN-Taste. (d. Red.)

Anschriften

Ich habe Ihnen vor drei Monaten eine Diskette geschickt und noch keine Antwort erhalten?
(tel. Anfragen)

Wer uns Datenträger zusendet, sollte unbedingt daran denken, sowohl auf dem Anschreiben, als auch auf dem Datenträger Namen und Anschrift zu vermerken. Es kommt immer wieder vor, daß der Datenträger vom Brief oder vom Umschlag getrennt wird, und dann geht die Sucherei los, wenn der Name fehlt. Noch ein besonderer Wunsch der Redaktion: wenn möglich, schicken Sie Ihre Programm-Einsendungen bitte auf Diskette! Sie kriegen Ihre Datenträger nach der Bearbeitung auf jeden Fall zurück. Uns ersparen Sie langwierige Ladezeiten und das oft unumgängliche Nachjustieren der Recorder. (d. Red.)

Das aktuelle Anwenderprogramm:

WM'86

WM'86 ist ein Programm zur Verwaltung der Daten der Fußball-Weltmeisterschaft 1986 in Mexiko. Es ist möglich, sämtliche Ergebnisse der einzelnen Spiele einzugeben und abzuspeichern. Dies umfaßt auch Spielort, Datum und Uhrzeit. Man kann sich alle Ergebnisse einer Mannschaft zeigen lassen, ebenso hat man die Möglichkeit, bis zu 150 Torschützen zu speichern. Sämtliche Daten der teilnehmenden Mannschaften und der Spiele der Vorrunde 'kennt' das Programm natürlich schon.

Kompliziertes Qualifikations-Prinzip . . .

Alle Mannschaften, die sich für die WM qualifiziert haben, sind in sechs Gruppen zu je vier Ländern aufgeteilt. In diesen Gruppen werden jeweils die beiden Besten (die in jedem Fall weiterkommen) ermittelt, indem jeder gegen jeden spielt. Vier Gruppen-Dritte haben auch noch die Möglichkeit, in das Achtel-Finale zu kommen. Der genaue Mechanismus ist einigermaßen kompliziert:

Die beiden Letzten der Tabelle der Dritten sind automatisch ausgeschieden, die anderen vier Mannschaften werden nach einem komplizierten System auf die ersten vier Spiele des Achtel-Finales verteilt:

Spiel Nr.1: Dritter der Gruppe A/C/D
Spiel Nr.2: Dritter der Gruppe A/B/F
Spiel Nr.3: Dritter der Gruppe C/D/E
Spiel Nr.4: Dritter der Gruppe B/E/F

Nehmen wir einmal an, die vier Besten der Tabelle der Dritten sind die Dritten aus Gruppe C,F,D und E. Somit sind die Dritten aus Gruppe A und B ausgeschieden. Diese beiden Mannschaften kann man nun schon aus der obigen Aufzählung streichen:

Spiel 1: C/D
Spiel 2: F
Spiel 3: C/D/F
Spiel 4: E/F

Bei Spiel 2 steht jetzt nur noch F, somit ist für dieses Spiel automatisch der Dritte aus Gruppe F qualifiziert. Nun kann man wieder Gruppe F streichen:

Spiel 1 C/D
Spiel 2 => 3. aus Gruppe F in Spiel 2
Spiel 3 C/D/E
Spiel 4 E

Nun steht in Spiel 4 wieder nur ein Buchstabe, dadurch ist in Spiel 4 der Dritte aus Gruppe E eine Runde weiter:

Spiel 1 C/D
Spiel 2 => F
Spiel 3 C/D
Spiel 4 => E

Jetzt wird von oben links beginnend der Dritte der ersten Gruppe in das Spiel genommen:

Spiel 1 Dritter der Gruppe C
Spiel 2 Dritter der Gruppe F
Spiel 3 Dritter der Gruppe D
Spiel 4 Dritter der Gruppe E

Mit diesem Verfahren werden alle verschiedenen Möglichkeiten abgedeckt. Durch dieses System ist es ebenfalls gewährleistet, daß die Mannschaften einer Gruppe nicht gleich wieder im Achtel-Finale aufeinandertreffen.

Ab dem Achtel-Finale wird im K.O.-System gespielt, das heißt, nur der Sieger erreicht die nächste Runde. Nach dem Achtel-Finale mit 16 Mannschaften treffen im Viertel-Finale nur noch acht Mannschaften aufeinander. Die Sieger aus den beiden Halbfinalbegegnungen treffen im Finale aufeinander, die Verlierer spielen um den 3.Platz.

. . . erledigt der Rechner

Das Programm ist Menue-gesteuert, es teilt sich in ein Haupt-Menue und vier Unter-Menues auf. In der ersten Zeile steht immer der aktuelle Menue-Punkt, in der letzten die jeweiligen Auswahlmöglichkeiten. Mit 'f1' gelangt man bei jeder Tastaturabfrage in das nächsthöhere Menue. Die Menue-Punkte:

1.) SPIELE

Nach der Auswahl dieses Menue-Punktes kann man sich für die Vor- oder für die Endrunde entscheiden:

1. Vorrunde

Es erscheint ein Hinweis, daß man sich in der Vorrunde befindet, daneben die aktuelle Gruppe und darunter die sechs Spiele. Es wird das Datum, die Uhrzeit (MEZ) und der Spielort gezeigt, dann die beiden Mannschaften und das Ergebnis. Sollte das Spiel noch nicht eingegeben sein, erscheinen zwei Striche. Unter den Spielen steht die Kommandozeile. Mit 'f5' und 'f7' kann zwischen den Gruppen hin- und hergeblättert wer-

den, durch 'f1' gelangt man zurück ins Spiele-Menue, mit 'D' werden die Ergebnisse der jeweiligen Gruppe auf dem Drucker ausgegeben. Mit 'T' wird die aktuelle Tabelle angezeigt. Durch 'K' wird der Cursor auf die erste Paarung gesetzt. Dort kann man die Tore der ersten Mannschaft eingeben und mit einem Return abschließen, danach wird mit der zweiten Mannschaft ebenso verfahren. Es sind maximal zweistellige Eingaben möglich. Beim Korrigieren eines Ergebnisses müssen beide (!) Torergebnisse nochmals eingegeben werden. Durch Space und Return wird das Ergebnis gelöscht. Drückt man nur Return wird der Cursor auf die nächste Paarung gesetzt. Hat man etwas an den Ergebnissen verändert, wird die neue Tabelle automatisch berechnet und angezeigt.

2. Endrunde
Der Bildschirmaufbau ist ähnlich wie bei der Vorrunde, nur wird statt der Gruppe die aktuelle Runde ausgegeben. Sind die teilnehmenden Mannschaften des Achtel-Finales noch nicht berechnet worden (siehe Tabellen), erscheint kein Mannschaftsname. Vor der Datumszeile steht die Spielnummer, die für die Berechnung der einzelnen Runden wichtig ist.
Blättern, Drucken und Korrigieren ist analog der Vorrunde. (Unentschieden kann nicht eingegeben werden, da bei Gleichstand eines Spiels dieses verlängert wird oder die Entscheidung im Elfmeterschiessen fällt.)

Durch Druck auf 'R' wird die nächste Runde berechnet, das heißt, alle Sieger werden den Spielen der nächsten Runde zugeordnet.

2.) ERGEBNISSE

Es werden alle Mannschaften aufgelistet, danach können Sie den Kennbuchstaben der Mannschaft eingeben. Nun werden alle Spiele der Mannschaft gezeigt. Ist die Mannschaft ausgeschieden, wird dies durch ein '. . . ist ausgeschieden' angezeigt.

3.) TABELLEN

1.–6. Gruppe A-F
Diese sechs Punkte haben dieselbe Funktion wie die Tabellenausgabe in der Vorrunde.

7. Achtel-Finale
Die ersten beiden Mannschaften der sechs Vorrunden-Gruppen werden den acht Spielen des Achtel-Finales zugeordnet. Alle Dritten der Vorrunden-Gruppen kommen in eine eigene Gruppe, die sortiert wird. Bevor dies geschieht, werden Sie gefragt, ob sie wählen möchten, welche Mannschaft welchen Platz in der Gruppe der Dritten bekommt, wenn zwei Mannschaften in der Vorrunde völlig gleich gut waren. Nach der Berechnung erscheint die Tabelle der Dritten.

4.TORSCHÜTZEN

In diesem Menue-Punkt ist es möglich, die Tore von 150 Spielern zu verwalten.

1. Neue Torschützen eingeben
Zuerst werden die Mannschaften aufgelistet, dann werden Sie nach einem vierzehnstelligen Namen und dem Kennbuchstaben der Mannschaft gefragt. Nun können Sie entweder noch einen Torschützen eingeben oder den Menue-Punkt verlassen. Im letzteren Fall kommen Sie direkt in die Torschützenliste.

2. Torschützenliste
Es werden die Namen der ersten 15 Torschützen aufgelistet, dahinter die Mannschaft und die geschossenen Tore. Mit 'CRSR-up' und 'CRSR-down' kann das 'größer als'-Zeichen am linken Bildschirmrand bewegt werden. Damit kann ein bestimmter Torschütze angewählt werden. Folgende Tasten besitzen eine Funktion:

'f5' und 'f7' – Blättern
'+' und '-' – Tore erhöhen (max.99) bzw. erniedrigen
'f1' – Zurück zum Menue
'D' – Torschützenliste ausdrucken
'N' – Der Name des Torschützen kann geändert werden. Return bestätigt den alten Namen.
'M' – Alle Mannschaften werden gezeigt, dann kann eine neue Mannschaft ausgewählt werden.
'L' – Nach einer Sicherheitsabfrage wird der Torschütze gelöscht.
'S' – Die Torschützenliste wird nach der Anzahl der Tore sortiert.

5.) DISK/KASSETTE

Im oberen Teil des Bildschirms steht eine Zeile mit dem aktuellen Speichergerät, wobei 'D' für Diskette, 'K' für Kassette steht. Drückt man 'K' erscheint darunter eine Zeile mit dem Kassettenformat. Mit 'S' und 'N' kann zwischen dem SuperTape DII-Format und dem normalen Commodore-Format umgeschaltet werden. SuperTape muß natürlich geladen und initialisiert sein. (SuperTape ist das derzeit schnellste Lade- und Speicherverfahren für Kassette. Die Version

für den C64 wurde in INPUT64 Ausgabe 4/85 veröffentlicht)

1. Daten speichern
Wird die Floppy zum Speichern benutzt, wird man nach einem zehnstelligen Dateinamen gefragt. Drücken Sie nur Return, wird als Name 'Daten' genommen. Vor den Namen wird dann 'WM'86-' gehängt und die Datei abgespeichert.

2. Daten laden
Für das Laden gilt dasselbe wie für das Speichern der Daten

3. Directory
Haben Sie die Floppy als Speichergerät gewählt, wird hier das Directory angezeigt.

4. Befehl senden
Es ist möglich,jeden Diskettenbefehl zu senden.

Achtung! Innerhalb von INPUT64 sind sämtliche Peripherie-Operationen gesperrt!

(Rainer Busch)

MultiTape

Mit diesem Werkzeug haben wir insbesondere der dubiosen Archive gedacht, die sich im Laufe eines Micro-Computer-Schicksals ansammeln. Da mag es Bandkassetten geben ohne Aufkleber mit rätselhaften Inhalten. Dateien und Programme in verschiedenen Aufzeichungsformaten wurden in "historischer" Reihenfolge abgespeichert. Oder jene schlichten Kassetten ohne Label oder Aufkleber, keiner weiß nichts genaues nicht.

Unter anderem aus Mitgefühl für solche Situationen wurde MultiTape entwickelt. Wenn Sie MultiTape in den Rechner geladen und mit RUN gestartet haben, genügt ein einfaches LOAD-Kommando. MultiTape erkennt selbstständig, in welchem Aufzeichnung-Format Programme auf der einliegenden Kassette vorhanden sind. Es folgen Meldung der Art:

FOUND "ERODEMMOC" COMMODORE
oder
FOUND "EPATREPUS" SUPERTAPE
oder
FOUND "TAMROFSTAF" FASTFORMAT

Wurde das gewünschte Programm entdeckt, wird es automatisch im entsprechenden Format geladen. Sie brauchen sich also nicht mehr um Initialisierung von Supertape oder Fastformat oder ähnlichem zu kümmern. Fastformat entspricht im Aufzeichungsformat dem FastTape aus dem "Cassetten Buch zu C64 und VC20" von Data Becker 1984 (ISBN 3-89011-030-4). Ein unserer Meinung nach informatives und sachkundiges Buch für Benutzer der Datasette.

Um den Ordnungsbedürfnissen Chaos-geschädigter Softwaresammler entgegenzukommen, können eingelesene Programme anschließend wahlweise im Commodore- oder SuperTape-Aufzeichunungsformat wieder abgespeichert werden.

SAVE "DASSPIEL" oder
SAVE "DASSPIEL,S" speichert in SuperTape
SAVE "SPIELDERSPIELE,C" im Commodore-Format

Zudem steht der Befehl VERIFY nach wie vor zur Verfügung. Wir empfehlen ein Archiv aufzubauen, wobei sich SuperTape aufgrund seiner großen Aufzeichungsgeschwindigkeit und des damit verbundenen geringen Bandverbrauchs besonders gut eignet. Experten werden bemerken, daß mit MultiTape Supertape-LOAD, -SAVE und -VERIFY zur Verfügung gestellt wird. Daniel Daboul, der Autor von MultiTape, arbeitet inzwischen an einer Erweiterung, mit der auch sequentielle Dateien erfaßt werden können.

Eine Warnung zum Abschluß: Autostart-Programme und Programme, die den gleichen Speicherbereich wie Multitape (ab $C800 = 51200) benutzen, vertragen sich nicht mit MultiTape.

Des LISP's zweiter Teil

Tracer, Editor, Mengen-Verarbeitung, Array-Handling

Mit diesen Erweiterungen wird der LISP-Interpreter aus der letzten Ausgabe zum (fast) kompletten LISP-Paket. Alle Files sind LISP-Programme (natürlich nur zusammen mit den LISP-Interpreter aus INPUT 64 4/86 lauffähig), die Sie aus dem Magazin heraus als sequentielle Dateien auf Kassette oder Diskette abspeichern können. Geladen – und das heißt unter LISP64 auch: in den Interpreter eingebunden – werden die einzelnen Erweiterungen mit

(LOAD ga"fn")

ga ist die Geräte-Adresse (1,7 oder 8), fn der Programmname (zum Beispiel MACROS.LSP)

MACROS.LSP

Durch diese Funktionen wird die weitgehend dem Standart 1.5 entsprechende Fassung des LISP 64 verschiedenen 'moderneren' Versionen angenähert. Vor allem die Einführung weiterer Kontrollstrukturen (FOR, IF, REPEAT, SELECTQ, WHILE) erleichtert die Programmierung. Das 'klassische' LISP kennt nämlich nur COND und die Rekursion, und das ist, gelinde gesagt, ziemlich ungewohnt.

MACROS sind Funktionen, die zunächst einen evaluierbaren S-Expr erzeugen, dessen Ergebnis dann den Funktionswert darstellt. Zum Definieren einer Funktion vom MACRO-Typ dient der Befehl DM (Define Macro).

Beim Aufruf eines Macros bildet der GESAMTE Macro-Aufruf einschliesslich des Macro-Namens das Argument.
Beispiel:

```
(DM XCONS L
(LIST 'CONS (CAR (CDDR L))
(CADR L)))
```

Nach dem Aufruf (XCONS 'A 'B) wird der gesamte Ausdruck an die Variable L gebunden, das heißt L = (XCONS 'A 'B). Danach ergibt sich als Zwischenergebnis der Ausdruck (CONS 'B 'A), dessen Evaluierung schliesslich das Ergebnis (B.A).

Unter der EXPANSION eines Macros versteht man dessen ERSETZEN durch den erzeugten S-Expr, zum Beispiel mit den Funktionen RPLACA, RPLCD und so weiter.

Die Macros des Files MACROS.LSP

MACRO-EXPANSION
globale Variable: wenn = T dann Macros expandieren

(EXPAND)
setzt MACRO-EXPANSION = T

(NO-EXPAND)
setzt MACRO-EXPANSION = NIL, das heißt Macros werden nicht expandiert

Macro ,Erläuterung, Beispiel, Expansion

Ähnlich wie bei den Interpreter-Befehlen, gilt die Reihenfolge: Befehl, Erläuterung, Beispiel und Expansion.

DECR
Dekrementieren einer Variablen
(DECR X)
= (SETQ X (SUB1 X))

INCR
Inkrementieren einer Variablen
(INCR X)
= (SETQ X (ADD1 X))

PUSH
Wert auf einer Liste ablegen
(PUSH X 'Y)
= (SETQ X (CONS 'Y X))

POP
Wert von einer Liste holen, Liste um dieses Element verkürzen
(POP X)
= (PROG1 (CAR X)
(SETQ X (CDR X)))

NEQ
Vergleich auf non-eq
(NEQ X Y)
= (NOT(EQ X Y))

MCONS
'multiple CONS'
(MCONS 'A 'B 'C)
= (CONS 'A (CONS 'B 'C))
= (A B.C)

NCONS
CONS Wert mit NIL
(NCONS 'A)
= (CONS 'A NIL)
= (A)

XCONS
vor dem CONS Werte vertauschen
(XCONS 'A 'B)
= (CONS 'B 'A)
= (B . A)

FUNCTION
= QUOTE, aber lesbarer, zum Beispiel bei (APPLY (FUNCTION CONS) '(A B))
(FUNCTION CAR)
=(QUOTE CAR)

F:L
verkürzende Schreibweise für (FUNCTION (LAMBDA. . .))
(F:L(X) (LIST X X))
= (FUNCTION (LAMBDA (X)
(LIST X X)))

Q:L
verkürzende Schreibweise für (QUOTE (LAMBDA. . .))
(Q:L(Y) (PRINT Y))
= '(LAMBDA (Y) (PRINT Y))

LOCAL
Einführung lokaler Variablen, die mit NIL vorbesetzt werden
(LOCAL(A B) (PRINT(LIST A B)))
= ((LAMBDA(A B)
(PRINT (LIST A B)))
NIL NIL)

LET
Einführung lokaler Variablen, die mit den zugeordneten Werten besetzt werden
(LET((A 3) (B 4)) (PLUS A B))
= ((LAMBDA (A B)
(PLUS A B))
3 4)
= 7

IF
IF-THEN-ELSE: erster S-Expr = Bedingung, zweiter S-Expr = THEN-Teil, restliche S-Expr = ELSE-Teil
(IF (LESSP X 4) (PRINT 'VIER)
= (COND ((LESSP X 4)
(PRINT 'VIER))
(T (PRINT '‹4)))

WHILE
WHILE-Schleife: solange die Bedingung (= erster S-Expr) ≠ NIL ist, restliche S-Expr evaluieren
(WHILE (LESSP X 20)
(PRINT X)
(INCR X))
= (PROG ()
LOOP
(COND ((LESSP X 20)
(PRINT X)
(INCR X))
(T (RETURN NIL)))
(GO LOOP))

REPEAT
Die S-Expr wiederholt so oft evaluieren, wie der Wert der ersten S-Expr (der nur einmal evaluiert wird) angibt
(REPEAT 4 (PRINT X)
(INCR X))
= (PROG (N)
(SETQ N 4)
LOOP
(COND ((ZEROP N)
(RETURN NIL))
(T (PRINT X)
(INCR X)))
(SETQ N (SUB1 N))
(GO LOOP))

FOR
FOR-NEXT-Schleife: Laufvariable (lokal), Startwert, Endwert, Schrittweite = 1 (oder -1 wenn Startwert ›Endwert) beliebig viele S-Expr als Anweisungen
(FOR I 10 20 (TAB I)
(PRINT I))
= (PROG (I)
(SETQ I 10)
LOOP
(COND ((GREATERP I 20)
(RETURN NIL))
(T (TAB I)
(PRINT I)))
(SETQ I (ADD1 I))
(GO LOOP))

SELECTQ
Fallunterscheidung ('CASE')
(SELECTQ X
(4 (PRINT 'VIER))
(5 (PRINT 'FUENF))
((6 7 8) (PRINT '›5&‹9))
(9 (PRINT 'NEUN)
(PRINT '›9))
= (COND ((EQ X 4) (PRINT 'VIER))
((EQ X 5) (PRINT 'FUENF))

```
((MEMBER X '(6 7 8))
(PRINT '›5&‹9))
((EQ X 9) (PRINT 'NEUN))
(T (PRINT'›9)))
```

TRACER.LSP

Wie dem Name schon andeutet, dient TRACER.LSP zum Testen von LISP-Programmen. Mit vier Befehlen werden verschiedene Trace-Modi ein- beziehungsweise ausgeschaltet:

(TRACE Function)/
(TRACE Function1 Function 2 ...)
setzt den Tracer auf eine oder mehrere selbstdefinierte Funktionen an, zum Beispiel (TRACE PERM1 PERM2). Bei jedem Aufruf einer "getracten" Funktion erscheint die Meldung 'ENTERING Function' sowie in Klammern die der Funktion übergebenen Argumente. Beim Verlassen der Funktion gibt der Tracer die Meldung 'EXITING Funktion =' sowie das Funktions-Ergebnis aus.

Angewandt auf die Funktion TOH.LSP (Towers of Hanoi, in der letzten Ausgabe vorgestellt), stellt sich der Ablauf des Tracing folgendermaßen dar:

1.) TRACER.LSP laden
2.) TOH.LSP laden
3.) Den Tracer auf beide Funktionen von TOH.LSP ansetzen:
    (TRACE TOH BEWEGE)
4.) Funktion aufrufen, die Eingabe von
    (TOH 3) liefert das folgende Trace-Protokoll, das die rekursiven Aufrufe von BEWEGE mit den jeweiligen Argumenten deutlich macht.

(SINGLE-STEP)?
schaltet den Einzelschritt-Modus ein, das heißt, nach jedem 'ENTERING . . .' wartet der Tracer auf das Drücken einer (beliebigen) Taste.

(NO-SINGLE-STEP)
macht (SINGLE-STEP) wieder rückgängig.

(UNTRACE Function)/
(UNTRACE Function1 Function2 ...)
hebt den TRACE-Befehl für die jeweilige Funktion auf.

Der TRACE-Befehl verändert die angegebene Funktion. Deshalb muß natürlich vor dem eventuellen Abspeichern einer Funktion der TRACE-Modus durch (UNTRACE . . .) aufgehoben werden.

ARRAYS.LSP

Bisweilen kann es auch in LISP nützlich sein, mit größeren Datenfeldern (Arrays) zu arbeiten – BASIC oder PASCAL ähnlich.

Die Darstellung von eindimensionalen Feldern ist sehr einfach: es sind Listen, deren geordnete Elemente die Feldkomponenten bilden. Der Zugriff auf das n-te Element geschieht mit
(CAR (NTH L N))
das Schreiben mit
(RPLACA (NTH L N) E)
Mehrdimensionale Felder kann man in dieser Weise darstellen, indem die Listenelemente wiederum Listen sind und so fort. Dabei wird jedoch der Zugriff auf ein einzelnes Feldelement sehr umständlich und unübersichtlich. Die folgenden Funktionen nehmen dem Anwender diesen Teil der Programmier-Arbeit ab.

(ARRAY N I Dim1 Dim2 ... Dimn)
ARRAY definiert ein n-dimensionales Feld mit dem Bezeichner N und initialisiert alle Feldelemente mit dem Wert I. Das Feld wird in der Eigenschaftsliste des Bezeichners unter der Eigenschaft ARRAY abgelegt.
(ARRAY A 0 3 2 2)
definiert das Array mit dem Bezeichner A als dreidimensional (1. Dimension) mit einer Tiefe von jeweils zwei in der zweiten und dritten Dimension. (Statt A könnte der Bezeichner natürlich auch ein sinnvoller Name sein, wie RAEUME oder dergleichen . . .) Der Befehl
(GETPROP 'A 'ARRAY)
macht dies sichtbar, er gibt zurück:
(((0 0)(0 0)) ((0 0)(0 0)) ((0 0)(0 0)))

(LOD N Index1 Index2 ... Indexn)
LOD ermittelt das durch die Indizes bestimmte Feldelement des Feldes N. Werden mehr Indizes angegeben als vorhanden, liefert LOD NIL. Werden weniger Indizes angegeben als die Dimensionierung erfordert, wird auf eine größere Feldstruktur ("Feldvektor") zugegriffen:
(LOD A 1 1 1) = 0
aber
(LOD A 1 1) = (0 0)
und
(LOD A 1) = ((0 0)(0 0))
und folgerichtig
(LOD A) = gesamtes Feld A

(STO N V Index1 Index2 ... Index3)
STO ("store") schreibt den Wert V an die durch die Indizes bestimmte Stelle des Feldes N; der alte Wert des Feldelementes geht verloren. Auch

hier gilt: werden weniger Indizes angegeben als die Dimensionierung erfordert, so wird an eine größere Feldstruktur zugewiesen. Der Wert von STO ist V. Zum Beispiel:

(STO A 55 1 1 1) = 55

ändert das Feld A folgendermaßen

(((55 0)(0 0)) ((0 0)(0 0))((0 0)(0 0)))

auch feststellbar durch

(LOD A 1 1) = (55 0)

aber:

(STO A 55 1 1)

ergibt

((55 (0 0)) ((0 0)(0 0))((0 0)(0 0)))

für Feld A.

SETS.LSP

Mengen sind in LISP Listen, in denen jedes Element höchstens einmal vorkommt:

(A B C D) ist eine Menge mit vier Elementen,

(A B A D) ist keine Menge (siehe aber MAKESET).

Mengenprädikate

(MEM1 M1 M2)

MEM1 prüft, ob mindestens ein Element der Menge M1 in der Menge M2 enthalten ist.Zum Beispiel:

(MEM1 '(A B C) '(X B Y)) = T
(MEM1 '(A B) '(C D E)) = NIL

(MEM1 '(A B C) '(X B Y)) = (B Y)
(MEM1 '(A B) '(C D E)) = NIL

(SETEQ M1 M2)

SETEQ prüft auf Mengengleichheit von M1 und M2.

(SETEQ '(A B C) '(B A C)) = T
(SETEQ '(A B C) '(B C D)) = NIL

(SUBSETP M1 M2)

SUBSETP prüft, ob M1 eine Untermenge von M2 ist. (Das heißt, ob alle Elemente von M1 auch in M2 enthalten sind).

(SUBSET '(A B C) '(A B C D)) = T
(SUBSET '(A B) '(D B C A X)) = T
(SUBSET '(A B C) '(A B D E)) = NIL

Mengenfunktionen

(ATTACH X L)

ATTACH plaziert X destruktiv (unter Verwendung von RPLACA/RPLACD!) an den Anfang von L.

(ATTACH 'D '(A B C)) = (D A B C))
(ATTACH 'A '()) = (A)

(DREMOVE X L)

DREMOVE entfernt alle Vorkommen von X in der Liste L destruktiv.

(DREMOVE 'A '(A B A C D)) = (B C D)

((ENTER X M)

Wenn X noch kein Element der Menge M ist, wird es hinzugefügt, ansonsten bleibt M unverändert.

(ENTER 'C '(A B D E) = (C A B D E)
(ENTER 'C '(A B C D) = (A B C D)

(INSERT X N L)

INSERT fügt das Element X vor dem n-ten Element der Liste (Menge) L ein.

(INSERT 'C 3 '(A B D E)) = (A B C D E)
(INSERT 'A 2 '(A B C)) = (A A B C)

(INTERSEKTION M1 M2)

Bildung des Mengendurchschnitts.

(INTERSECTION '(A B C D) '(C D E F)) = (C D)

(INTERSECTION '(A X D C) '(C A Y D)) = (A D C)

(MAKESET L)

MAKESET macht aus der Liste L eine Menge, indem mehrfach vorkommende Elemente nur einmal aufgenommen werden.

(MAKESET '(A B B A C D)) = (A B C D)
(MAKESET '(A B A A) = (A B)

(REMOVE X L)

REMOVE entfernt alle Vorkommen von X in der Liste L, indem eine neue Liste ohne das Element X aufgebaut wird (siehe auch DREMOVE).

(REMOVE 'A '(A B A C D A)) = (B C D)

(REMOVE '(A B) '(A (A B) C (A B) D)) = (A C D)

(SUBSET FN M)

SUBSET bildet eine Untermenge aus den Elementen von M, die der Prädikatfunktion FN genügen.

(SUBSET 'NUMBERP '(A 1 B 2 44 C)) = (1 2 44)

(SUBSET 'ATOM '(A (A B) C (D E) D)) = (A C D)

(SYMM-DIFF M1 M2)

Bildung der Mengendifferenz zwischen M1 und M2.

(SYMM-DIFF '(A B C D E) '(B D)) = (A C E)

(SYMM-DIFF '(A B C) '(A)) = (B C)

(UNION M1 M2)

Bildung der Vereinigungsmenge von M1 und M2.

(UNION '(A B C D) '(A D E B C)) = (A B C D)

(UNION '(A B C) '(C D)) = (A B C D)

EDITOR.LSP

Dieser in LISP geschriebene Editor ist ein sogenannter Listen-Editor, das heißt, er dient dem Editieren und Modifizieren beliebiger S-Expr (Listen). Solche Listen sind dabei auch Funktions-Definitionen (die entweder nicht 'laufen' oder geändert werden sollen), Atom-Eigenschaften oder Variablen-Werte. Mit dem Aufruf des Editors wird ein besonderer EDIT-Modus betreten. Der Editor meldet sich mit dem Prompt *ED*, es gelten folgende Regeln:

- es können mehrere Editier-Kommandos in einer Zeile eingegeben werden
- Editier-Kommandos ohne Argumente brauchen nicht geklammert zu werden, es genügt die Angabe des Kommando-Namens.
- nach Abarbeitung einer Zeile mit Editier-Befehlen erfolgt nur dann eine Ausgabe, wenn Ausgabebefehle (P und PP) darunter sind.
- Zahlen haben im EDIT-Modus eine besondere Funktion, da sie zur Anwendung bestimmter Editier-Kommandos führen (siehe unten)

Der EDIT-Modus wird durch Eingabe von OK verlassen.

(EDIT X)
Allgemeine Editier-Funktion für beliebigen S-Expr. Zum Beispiel:

(EDIT '(A B (C D) E))

(EDITF FN)
Editieren der Funktion FN. EDITF sucht in der Eigenschaftsliste von FN nach der Eigenschaft EXPR bzw. FEXPR und editiert sie. Im Ergebnis wirkt EDITF wie DE bzw. DF.

(EDITF FAK)
(EDITF EXPT)

(EDITP X Y)
EDITP editiert die Eigenschaft (property) Y des Bezeichners X und wirkt insgesamt wie ein PUTPROP.

(EDITP PUNKT XYZ-KOORD) = (3 4 1)

(EDITV X)
EDITV editiert den Wert der Variablen X und entspricht somit (EDITP X VALUE).

(EDFNS)
EDFNS ist eine Variable, die alle Editier-Funktionen sowie deren Hilfsfunktionen enthält.

Im folgenden ist unter der Bezeichnung "aktuelle Liste" bzw. "aktueller S-Expr" der gerade bearbeitete S-Expr zu verstehen.

Es gibt drei Klassen von Editier-Kommandos::!l
- Ausgabe-Kommandos: Kommandos, die den aktuellen S-Expr oder anderes ausgeben. Diese Kommandos sind ohne Klammern einzugeben; Ausnahme: (E X)
- "Zeiger"-Kommandos: Kommandos, die nicht den aktuellen S-Expr verändern, sondern im S-Expr ab- bzw. aufsteigen und so die "Aufmerksamkeit" des Editors auf bestimmte Teile des S-Expr richten. Ebenfalls ohne Klammern einzugeben; Ausnahme: (F X)
- Modifizierende Kommandos: Kommandos, die den aktuellen S-Expr modifizieren (unter Verwendung der LISP-Funktionen RPLACA, RPLACD, NCONC). Diese Kommandos werden **mit** Klammern eingegeben; Ausnahme: UNDO

Alle Änderungen während des Editierens lassen sich mit UNDO rückgängig machen.

Falls der Prompt des Editors nicht mehr erscheint, ist er wahrscheinlich aufgrund einer Fehlbedienung ausgestiegen! In den folgenden Beispielen wird als aktuelle Liste angenommen:

(A B (C D E) F G H)

Ausgabekommandos

P
P druckt den aktuellen S-Expr, wobei alle nichtatomaren Elemente durch das Zeichen & ersetzt werden

P
(A B & F G H)

PP
PP erzeugt ein "pretty print" des aktuellen S-Expr.

PP
(A B
(C D E) F G H)

P:
wie PP

(E X)
E evaluiert den S-Expr X und druckt das Ergebnis.

(E (PLUS 1 2 3 4)) = 10

"Zeigerkommandos"

n

n steht für einen numerischen Wert, der bewirkt:
bei n = 0: Rücksprung zum Listenanfang, das heißt: die Liste, in der der gegenwärtige aktuelle S-Expr enthalten ist, wird zur neuen Liste.
bei n größer 0: das n-te Element der aktuellen Liste wird zum neuen aktuellen S-Expr.
bei n kleiner 0: das n-te Element der aktuellen Liste, vom Listen**ende** aus **rüchwärts** gezählt, wird zum neuen aktuellen S-Expr. Bezogen auf die oben angegebene Beispiel-Liste, kann dies folgendermaßen aussehen:

Erst die ganze Liste betrachten
```
P
(A B & F G H)
```
Auf den dritten S-Expr (= das dritte Listenelement) zeigen lassen und diesen ausgeben
```
3 P
(C D E)
```
Zurück zum Anfang
```
0 P
(A B & F G H)
```
Vorletzten S-Expr betrachten (wäre nicht zuvor das 0-Kommando gegeben worden, wäre der aktuelle S-Expr weiterhin (C D E)!)
```
-2 P
G
```
Erneut zurück zum Anfang, letztes Element auswählen und anzeigen
```
0 -1 P
H
```

NX

meint NEXT, das heißt, wenn ein n-Kommando gegeben wurde, wird nun das n+1-te Element zum aktuellen S-Expr.
```
P
(A B & F G H)
4 P
F
NX P
G
```

← L

←L führt so oft das 0-Kommando aus, bis der Anfang der ursprünglich zu editierenden Liste erreicht ist.
```
P
(A B & F G H)
3 2 P
D
← L P
(A B & F G H)
```

(F X)

F (Find) sucht den S-Expr X im aktuellen S-Expr. Wird er gefunden, so wird die Liste, in der X enthalten ist, zur neuen aktuellen Liste. Wird X nicht gefunden, bleibt das F-Kommando ohne Wirkung.
```
(F D)
(C D E)
0 P
(A B & F G H)
(F Z)
(A B & F G H)
```

Modifikations-Kommandos

(Die Beispiele sind meist mit UNDO abgeschlossen, um beim Ausprobieren nicht jedesmal die Test-Liste neu eingeben zu müssen. Normalerweise ist der beendende UNDO-Befehl natürlich unsinnig!)

(n)

n ist, wie bei den Zeigerkommandos, eine ganze Zahl. Es gilt:
bei n = 0: keine Wirkung
bei n größer 0: das n-te Element der aktuellen Liste wird gelöscht (aus der Liste entfernt)
bei n kleiner 0: das n-te Element der aktuellen Liste, vom Listenende aus rückwärts gezählt, wird gelöscht.
```
P
(A B & F G H)
(2) P
(A & F G H)
(-2) p
(A & F H)
(3) (3) P
(A &)
UNDO
```

(n E1 E2 ... Ei)

Lösch- und Insert-Kommando, es gilt:
bei n = 0: die S-Expr E1 bis Ei werden am Anfang der aktuellen Liste eingefügt.
bei n größer 0: das n-te Element der aktuellen Liste wird gelöscht und dafür die S-Expr E1 bis Ei eingesetzt.
bei n kleiner 0: das n-te Element der aktuellen Liste, vom Listenende aus rückwärts gezählt, wird gelöscht und durch die E1 bis Ei ersetzt.
```
P
(A B & F G H)
(2 X) (-2 Y) P
(A X & F Y H)
(1 (CAR X) Z (CDR X)) P
((CHAR X) Z & X & F Y H)
```

(0 A B) P
(A B & Z & X & F Y H)
UNDO

(A E1 E2 ... Ei)
A (after) fügt die S-Expr E1 bis Ei hinter das aktuelle Listenelement ein.

P
(A B & F G H)
4 (A X Y Z) P
(A B & F X Y Z G H)
2 (A (CAR X)) P
(A B & & F X Y Z G H)
UNDO

(B E1 E2 ... Ei)
B (begin) fügt die S-Expr E1 bis Ei vor dem ersten Element der aktuellen Liste ein.

P
(A B & F G H)
(B X Y) P
(X Y A B & F G H)
UNDO

(N E1 E2 ... Ei)
N (NCONC) fügt die S-Expr E1 bis Ei am Ende der aktuellen Liste ein (mit der Funktion NCONC).

P
(A B & F G H)
(N X Y)
(A B & F G H X Y)
UNDO

(: E1 (E2 ... Ei))
: ersetzt den aktuellen S-Expr durch die Liste der S-Expr E1 bis Ei.

3 (: X Y) PP
(A B (X Y) F G H)
UNDO

(R E1 E2)
R (replace) ersetzt **alle** Vorkommen von E1 im aktuellen S-Expr durch E2

P
(A B & F G H)
(R G X) P
(A B & F X H)
(3 A X A A Y) P
(A B A X A A Y F X H)
(R A Z)
(Z B Z X Z ZY F X H)
UNDO

(BI n1 n2)
BI (both in) setzt das n1-te bis n2-te Element der aktuellen Liste in Klammern.

P
(A B & F G H)
(BI 1 3) P
((A B &) F G H)
(BI 2 3) PP
((A B &) (F G) H)
UNDO

(BO n)
BO (both out) entfernt die Klammern um das n-te Element der aktuellen Liste, wenn es kein Atom ist.

P
(A B & F G H)
(BO 3)
(A B C D E F G H)
UNDO

(LI n)
LI (left in) setzt vor das n-te Element und nach dem letzten Element der aktuellen Liste eine Klammer.

P
(A B & F G H)
(LI 2) PP
(A (B (C D E) F G H))
UNDO

(LO n)
LO (left out) entfernt die Klammer vor dem n-ten Element der aktuellen Liste, wenn es nicht-atomar ist.

PP
(A B (C D E) F G H)
(LO 3) PP
(A B C D E)
Achtung! die Elemente n+1, n+2 usw. gehen verloren!

(RI n1 n2)
RI (right in) verschiebt, wenn das n1-te Element der aktuellen Liste nicht-atomar ist, die schließende Klammer des n1-ten Elements hinter dessen n2-tes Element. Die Beispiele mögen dies näher verdeutlichen

PP
(A B (C D E) F G H)
(RI 3 2) PP
(A B (C D) E F G H)
(RI 3 1) PP
(A B (C) D E F G H)
UNDO

(RO n1 n2)
RO (right out) verschiebt, wenn das n1-te Element der aktuellen Liste nicht-atomar ist, dessen schließende Klammer hinter das n2-te Element der aktuellen Liste.

PP

```
  (A B (C D E) F G H)
  (RO 3 5) PP
  (A B (C D E F G) H)
  (RO 3 4) PP
  (A B (C D E F G H))
UNDO
```

UNDO
UNDO macht alle vorgenommenen Listen-Modifizierungen rückgängig und springt an den Anfang der ursprünglich zu editierenden Liste (genauer: einer Kopie der Liste) zurück).

Beispielsitzung mit dem Editor

Die bekannte Fakultäts-Funktion sei wie folgt fehlerhaft definiert

```
  (DE FAK (X Y Z)
  (COND (ZEROP X 0)
(T (TIMES (X) (FAK (ADD1 Z X))))))
```

Richtig muß es heißen

```
  (DE FAK (X)
  (COND ((ZEROP X) 1)
(T (TIMES X (FAK (SUB1 X))))))
```

Im folgenden wird gezeigt, wie dieses Ziel durch schrittweises Anwenden der Editier-Funktionen erreicht werden kann. Die Ausgaben des Editors sind dabei eingerückt.

```
Betreten des EDIT-Modus:
  (EDITF FAK)
  ((X Y Z) &)
  Variablen-Liste korrigieren:
  (1 (X)) P
  ((X) &)
  2 P
  (COND & &)
  2 P
  (ZEROP X 0)
  Erstes und zweites Element klammern:
  (BI 1 2) P
  ((ZEROP X) 0)
  Löschen der 0 und ersetzen durch 1:
  (-1 1) P
  ((ZEROP X) 1)
  Eine Stufe zurück:
  0 P
  (COND & &)
  -1 P
  (T &)
  -1 P
  (TIMES & &)
  2 P
  (X)
  Löschen und Ersetzen durch X:
  (: X) P
  X
  0 P
  (TIMES X &)
  3 2 P
  (ADD1 Z X)
  Löschen des Z:
  (2) P
  (ADD1 X)
  Ersetzen des ADD1 durch SUB1:
  (R ADD1 SUB1) P
  (SUB1 X)
  0 P
  (FAK &)
  Zurück zum Anfang:
  ←L P
  ((X) &)
  Fertig:
  OK
FAK
```

Das Protokoll könnte dann folgendermaßen weitergehen:

```
  (PDEF FAK)
  (DE FAK (X)
  (COND ((ZEROP X) 1)
  (T (TIMES X (FAK (SUB1 X))))))

  (FAK 5)
120
```

(Natürlich würde jeder vernünftige Mensch ein derart kurzes Programm einfach neu eingeben. Aber es geht ja um die Handhabung des Editors.)

PRINTFILE.LSP

PRINTFILE.LSP ist ein Utility, das die Handhabung von Ein-/Ausgabe-Operationen erleichtert. Außerdem ist es sehr instruktiv für die Benutzung der I/O-Befehle (OPEN, INPUT, OUTPUT, LINE usw.) von LISP64.

(PRINT-FILE gn "name")
Das File "name" wird vom Gerät gn (1 = Kassette, 7 = SuperTape, 8 = Diskette) auf den Drucker gelesen.

(PRINT-ON-SCREEN gn "name")
Wie PRINT-FILE, aber die Ausgabe erfolgt auf den Bildschirm.

Listing MACROS.LSP

```
T

(DE EXPAND NIL
  (SETQ MACRO-EXPANSION T))
NIL

(DE NO-EXPAND NIL
  (SETQ MACRO-EXPANSION NIL))
NIL

NIL
NIL

(DM FOR L
  (REPLACE L
    (PROG(VAR VON NACH COUNT-FN
        TEST-FN)
      (SETQ VAR
        (CADR L))
      (SETQ VON
        (EVAL(CAR(CDDR L))))
      (SETQ NACH
        (EVAL(CADR(CDDR L))))
      (COND((GREATERP VON NACH)
          (SETQ TEST-FN
            'LESSP)
          (SETQ COUNT-FN
            'SUB1))
        (T(SETQ TEST-FN
            'GREATERP)
          (SETQ COUNT-FN
            'ADD1)))
      (RETURN(LIST 'PROG
          (LIST VAR)
          (LIST 'SETQ VAR VON)
           'LOOP
          (LIST 'COND
            (LIST(LIST TEST-FN VAR
                NACH)
              '(RETURN NIL))
            (CONS 'T
              (CDDR(CDDR L))))
          (LIST 'SETQ VAR
            (LIST COUNT-FN VAR))
           '(GO LOOP))))))
NIL

(DM SELECTQ L
  (REPLACE L
    (CONS 'COND
      ((LABEL SELECTQ1
        (LAMBDA(X L)
          (COND((ATOM(CDR L))
              (LIST(LIST 'T
                  (CAR L))))
           (T(CONS(CONS(LIST(COND((
                     ATOM(CAAR L))
                       'EQ)
                     (T 'MEMBER)) X
                       (LIST 'QUOTE
                         (CAAR L)))
                     (CDAR L))
                   (SELECTQ1 X
                     (CDR L)))))))
        (CADR L)
        (CDDR L)))))
NIL

NIL
NIL

(DM IF L
  (REPLACE L
    (LIST 'COND
      (LIST(CADR L)
        (CAR(CDDR L)))
      (CONS 'T
        (CDR(CDDR L))))))
NIL

(DM WHILE L
  (REPLACE L
    (LIST 'PROG NIL
       'LOOP
      (LIST 'COND
        (LIST(LIST 'NOT
            (CADR L))
          (LIST 'RETURN NIL))
        (CONS 'T
          (CDDR L)))
       '(GO LOOP))))
NIL

(DM REPEAT L
  (REPLACE L
    (LIST 'PROG
       '(N)
      (LIST 'SETQ
         'N
        (EVAL(CADR L)))
       'LOOP
      (LIST 'COND
        (LIST '(ZEROP N)
           '(RETURN NIL))
        (CONS 'T
          (CDDR L)))
       '(SETQ N
        (SUB1 N))
       '(GO LOOP))))
NIL

(DM LET L
  (REPLACE L
    (CONS(CONS 'LAMBDA
        (CONS(MAPCAR 'CAR
            (CADR L))
          (CDDR L)))
      (MAPCAR 'CADR
        (CADR L)))))
NIL
```

```
(DM LOCAL L
  (REPLACE L
    (CONS(CONS 'LAMBDA
        (CDR L)) NIL)))
NIL

(DM INCR L
  (REPLACE L
    (LIST 'SETQ
      (CADR L)
      (LIST 'ADD1
        (CADR L)))))
NIL

(DM DECR L
  (REPLACE L
    (LIST 'SETQ
      (CADR L)
      (LIST 'SUB1
        (CADR L)))))
NIL

(DM PUSH L
  (REPLACE L
    (LIST 'SETQ
      (CADR L)
      (LIST 'CONS
        (CAR(CDDR L))
        (CADR L)))))
NIL

(DM POP L
  (REPLACE L
    (LIST 'PROG1
      (LIST 'CAR
        (CADR L))
      (LIST 'SETQ
        (CADR L)
        (LIST 'CDR
          (CADR L))))))
NIL

(DM MCONS L
  (REPLACE L
    (COND((ATOM(CDDR L))
        (CADR L))
      (T(LIST 'CONS
          (CADR L)
          (CONS 'MCONS
            (CDDR L)))))))
NIL

(DM NCONS L
  (REPLACE L
    (LIST 'CONS
      (CADR L) NIL)))
NIL

(DM XCONS L
  (REPLACE L
    (LIST 'CONS
      (CAR(CDDR L))
      (CADR L))))
NIL

(DM FUNCTION L
  (REPLACE L
    (LIST 'QUOTE
      (CADR L))))
NIL

(DM F:L L
  (REPLACE L
    (LIST 'QUOTE
      (CONS 'LAMBDA
        (CDR L)))))
NIL

(DM Q:L L
  (REPLACE L
    (LIST 'QUOTE
      (CONS 'LAMBDA
        (CDR L)))))
NIL

(DM NEQ L
  (REPLACE L
    (LIST 'NOT
      (LIST 'EQ
        (CADR L)
        (CAR(CDDR L))))))
NIL

NIL
NIL
```

Listing TRACER.LSP

```
T
(DF TRACE L
  (SETQ TRACE-SPACES 0)
  (NO-SINGLE-STEP)
  (MAPC '(LAMBDA(FUNC)
      (PROG(X)
        (SETQ X
          (OR(GETPROP FUNC
              'EXPR)
            (GETPROP FUNC
              'FEXPR)
            (GETPROP FUNC
              'MACRO) NIL))
        (COND((NULL X)
            (RETURN NIL)))
        (RPLACD(CDR X)
          (LIST(LIST 'EVTRACE FUNC
              (CADR X)
              (CDDR X)))))) L) L)
NIL

(DF UNTRACE L
```

```
  (SETQ TRACE-SPACES 0)
  (MAPC '(LAMBDA(FUNC)
      (PROG(X)
        (SETQ X
          (OR(GETPROP FUNC
                'EXPR)
             (GETPROP FUNC
                'FEXPR)
             (GETPROP FUNC
                'MACRO) NIL))
        (COND((NULL X)
            (RETURN NIL)))
        (RPLACD(CDR X)
          (LAST(LAST X))))) L) L)
NIL

(DF EVTRACE
  (TRFUN TRVARS TRBODY)
  (PROG(TRRESULT)
    (PRINTENTRY TRFUN TRVARS)
    (SETQ TRRESULT
      (APPLY 'PROGN TRBODY))
    (PRINTEXIT TRFUN TRRESULT)
    (RETURN TRRESULT)))
NIL

(DE PRINTENTRY
  (TRFUN TRVARS)
  (SPACES(SETQ TRACE-SPACES
      (ADD1 TRACE-SPACES)))
  (MSG "ENTERING " TRFUN " [")
  (PRINTENTRY1 TRVARS)
  (MSG "]" T)
  (COND(SINGLE-STEP-V(WAITCHAR))))
NIL

(DE PRINTENTRY1
  (TRVARS)
  (COND((NULL TRVARS)
       NIL)
    ((ATOM TRVARS)
      (PRIN1(EVAL TRVARS)))
    ((ATOM(CDR TRVARS))
      (PRIN1(EVAL(CAR TRVARS))))
    (T(PRIN1(EVAL(CAR TRVARS)))
      (MSG ",")
      (PRINTENTRY1(CDR TRVARS)))))
NIL

(DE PRINTEXIT
  (TRFUN TRRESULT)
  (SPACES(SETQ TRACE-SPACES
      (SUB1 TRACE-SPACES)))
  (MSG " EXITING  " TRFUN " = ")
  (PRINT TRRESULT)
  (COND(SINGLE-STEP-V(WAITCHAR))))
NIL

(DE SINGLE-STEP NIL
  (SETQ SINGLE-STEP-V T))
NIL

(DE NO-SINGLE-STEP NIL
  (SETQ SINGLE-STEP-V NIL))
NIL

NIL
NIL

NIL
NIL
```

TRACE-PROTOKOLL TOH/BEWEGE

```
T

 ENTERING TOH [3]
  ENTERING BEWEGE [LINKS,RECHTS,MITTE,3]
   ENTERING BEWEGE [LINKS,MITTE,RECHTS,2]
    ENTERING BEWEGE [LINKS,RECHTS,MITTE,1]
     ENTERING BEWEGE [LINKS,MITTE,RECHTS,0]
     EXITING  BEWEGE = T
(LEGE SCHEIBE VON LINKS NACH RECHTS)
     ENTERING BEWEGE [MITTE,RECHTS,LINKS,0]
     EXITING  BEWEGE = T
    EXITING  BEWEGE = T
(LEGE SCHEIBE VON LINKS NACH MITTE)
    ENTERING BEWEGE [RECHTS,MITTE,LINKS,1]
     ENTERING BEWEGE [RECHTS,LINKS,MITTE,0]
     EXITING  BEWEGE = T
(LEGE SCHEIBE VON RECHTS NACH MITTE)
     ENTERING BEWEGE [LINKS,MITTE,RECHTS,0]
     EXITING  BEWEGE = T
    EXITING  BEWEGE = T
   EXITING  BEWEGE = T
(LEGE SCHEIBE VON LINKS NACH RECHTS)
   ENTERING BEWEGE [MITTE,RECHTS,LINKS,2]
    ENTERING BEWEGE [MITTE,LINKS,RECHTS,1]
     ENTERING BEWEGE [MITTE,RECHTS,LINKS,0]
     EXITING  BEWEGE = T
(LEGE SCHEIBE VON MITTE NACH LINKS)
     ENTERING BEWEGE [RECHTS,LINKS,MITTE,0]
     EXITING  BEWEGE = T
    EXITING  BEWEGE = T
(LEGE SCHEIBE VON MITTE NACH RECHTS)
    ENTERING BEWEGE [LINKS,RECHTS,MITTE,1]
     ENTERING BEWEGE [LINKS,MITTE,RECHTS,0]
     EXITING  BEWEGE = T
(LEGE SCHEIBE VON LINKS NACH RECHTS)
     ENTERING BEWEGE [MITTE,RECHTS,LINKS,0]
     EXITING  BEWEGE = T
    EXITING  BEWEGE = T
   EXITING  BEWEGE = T
  EXITING  BEWEGE = T
 EXITING  TOH = T
T
```

Listing ARRAYS.LSP

```
T

(DF ARRAY L
  (PUTPROP(CAR L)
      'ARRAY
    (DIM(MAPCAR 'EVAL
        (CDDR L))
      (EVAL(CADR L))))
  (CAR L))
```

```
(DF LOD L
  (LOD1(GETPROP(CAR L)
       'ARRAY)
    (MAPCAR 'EVAL
      (CDR L))))
(DF STO L
  (STO1(GETPROP(CAR L)
       'ARRAY)
    (EVAL(CADR L))
    (MAPCAR 'EVAL
      (CDDR L))))
(DE DIM
  (NLIS E)
  (COND((ATOM NLIS)
      (COPY E))
    (T(BUILD(CAR NLIS)
        (DIM(CDR NLIS) E)))))
(DE BUILD
  (N E)
  (COND((ZEROP N)
       NIL)
    (T(CONS(COPY E)
        (BUILD(SUB1 N) E)))))
(DE STO1
  (L E DIMS)
  (COND((ATOM DIMS)
       NIL)
    ((ATOM(CDR DIMS))
      (RPLACA(NTH L
           (CAR DIMS)) E))
    (T(STO1(CAR(NTH L
             (CAR DIMS))) E
        (CDR DIMS)))))
(DE LOD1
  (L DIMS)
  (COND((ATOM DIMS)
       L)
    (T(LOD1(CAR(NTH L
             (CAR DIMS)))
        (CDR DIMS)))))

(DF FOR L
(PROG(VAR FST LST EXPRS TEST-FN
      COUNT-FN)
    (SETQ VAR
      (CAR L))
    (SETQ FST
      (EVAL(CADR L)))
    (SETQ LST
      (EVAL(CAR(CDDR L))))
    (COND((LESSP LST FST)
        (SETQ TEST-FN
          'LESSP)
        (SETQ COUNT-FN
          'SUB1))
      (T(SETQ TEST-FN
          'GREATERP)
        (SETQ COUNT-FN
          'ADD1)))
    (SETQ EXPRS
      (CDR(CDDR L))) LOOP
    (COND((TEST-FN FST LST)
        (RETURN NIL)))
    (SET VAR FST)
    (MAPC 'EVAL EXPRS)
    (SETQ FST
      (COUNT-FN FST))
    (GO LOOP)))
(DF WHILE L
  (PROG(CON EXPRS)
    (SETQ CON
      (CAR L))
    (SETQ EXPRS
      (CDR L)) LOOP
    (COND((EVAL CON)
        (MAPC 'EVAL EXPRS)
        (GO LOOP)))))
(DF IF L
  (COND((EVAL(CAR L))
      (EVAL(CADR L)))
    (T(LAST(MAPCAR 'EVAL
         (CDDR L))))))
NIL
NIL
```

Hardwaretest

NLQ-Drucker

Drei verschiedene Drucker der STAR Micronics stellten sich uns zum Test:

SG10-C NL10 SR15

Diese Drucker haben Gemeinsamkeiten, aber auch beachtliche Unterschiede, die sich nicht zuletzt im Preis niederschlagen.

Alle Geräte verarbeiten Einzelblatt oder Endlospapier. Darüber hinaus bieten sie, neben einer guten Nadel-Drucker Qualität, zusätzlich Schönschrift-Qualität NLQ (Near Letter Quality). Die Drucker werden mit einem übersichtlichen Handbuch ausgeliefert, indem auch den Bedürfnissen von BASIC-Programmierern Rechnung getragen wird.

Der NL-10 ist das jüngste Kind der STAR-Drucker Familie. Er gilt als Nachfolgemodell des SG10.

Der SG10C ist von Haus aus für den C64 ausgerüstet. Er kann direkt an den seriellen Bus der Anlage angeschlossen werden. C64-Benutzer wissen das zu schätzen, auch wenn man damit einen Drucker gekauft hat, der sich bei einem System-Wechsel nicht mehr verwenden läßt.

Beim Auspacken fallen einem eine Reihe von Einzelteilen in die Hände, die zur mechanischen Ausrüstung gehören: Zwei Trenngitter und das Stellrad für den Papiertransport, das Buskabel für den seriellen Anschluß und eine Ersatzsicherung. Besonders erfreulich ist das 'normale' Schreibmaschinenfarbband. Leider ist die Transportmechanik der Spulen auf 4-Loch Anordnung in Spulenkörpern festgelegt, sodaß nicht jedes x-beliebige Farbband paßt.

Gut gelöst wurde die Anordnung der DIP-Schalter an der Außenseite des Druckers, sodaß das umständliche Aufschrauben des Gehäuses fortfällt; Über diese Schalter können neben verschiedenen internationalen Zeichensätzen, Seitenlänge (11 oder 12 Zoll), die Geräteadresse (4 oder 5), der Papiersensor (ein/aus) und die Schriftarten:

Normal/Kursiv, Pica/Engdruck, Fettdruck/NLQ
Da der Befehlsvorrat des SG-10C in etwa dem des MPS-802 entspricht, sind die Sonderfunktionen, wie Schriftart zum Beispiel, nur über Schalter einzustellen. Die 'C'-Version des SG-10 kennt keine 'ESC'-Sequenzen. Mit der direkten Anpassung an den C64 muß man auf Druckerpuffer und Hex-Dump verzichten.

Der NLQ-Zeichensatz steht bei diesem Modell nur in einer Schriftart (Pica normal) zur Verfügung. Das Schriftbild ist im Allgemeinen scharf. Die NLQ-Qualität kann durchaus mit einem üblichen Schreibmaschinen-Schriftbild mithalten.

Zur Einzelblatt-Verarbeitung wird der Traktoraufsatz vorher entfernt, was mit einem Handgriff erledigt ist. Die Papierführung beim Endlostransport liegt oberhalb der Abreißkante und neigt zum Wiedereinziehen des beschriebenen Blattes, was irgendwann vom Drucker mit schändlichen Geräuschen und Stillstand quittiert wird. Unverständlich bleibt, warum die Hersteller die Anschlußkabel unbedingt an der Rückseite anbringen mußten, sodaß der Stecker den Papiereinzug behindert, zumal der Traktor schon bei leichter Behinderung die Randlochung ausreißt und seiner Aufgabe nichtmehr gerecht werden kann. Wer einen reibungslosen Betrieb genießen will, muß den Drucker mit Hilfe eines entsprechenden Regals oder anderer Hilfsmittel so aufstellen, daß das Papier ohne Behinderung durch den Drucker geführt wird.

Der Nachfolger NL-10 ist als konsequente Weiterentwicklung von eben diesen Mängeln befreit worden. Durch bessere Gestaltung der Papierführung, des Traktors, der Abreißkante und einer Abdeckplatte gab es im Test keinerlei Probleme. Der NL-10 versieht auch ohne ein wachsames Auge seinen Dienst, erfreulicherweise auch mit weniger Geräuschentwicklung. Der Einzelblatteinzug ist halbautomatisch. Zwar muß erst der Traktor verstellt werden und die Papieranlage aufgerichtet werden, aber dann lassen sich Einzelblätter ohne Probleme einziehen und gut positionieren. Nur bei linksbündigem Anlegen im Einzugschacht ist die eine Traktorwalze im Weg.

Besonders beeindruckend ist das mitgelieferte Modul. Ein kompakter, solide verarbeiteter Block, der in den Modulschacht des Druckers an der Rückseite leicht einzuschieben ist. Es können wahlweise Parallel- , IBM-PC- und C64/128-Module bestellt werden. Ein klarer Vorteil gegenüber dem Vorgänger. Bei einem Systemwechsel wird nur ein neues Modul benötigt. Außerdem ist der NL-10 als vorläufig Einziger der STAR-Crew sowohl mit dem C64 als auch mit dem C128 verträglich! Der serielle Bus ist beim Commodore-Modul durchgeschleift, das heißt, das Modul hat zwei Anschlußbuchsen.

Das NLQ-Schriftbild (18*23) ist besser ausdifferenziert als beim SG10 (17*11) und läßt sich beim NL10 auch für Kursiv-Schrift einstellen. Außerdem bietet das Gerät die Möglichkeit zur doppelten und vierfachen Vergrößerung. Das Gerät kann auch in der Commodore-Anpassung über ESC-Sequenzen angesprochen werden, ist damit also in allen Funktionen über Software erreichbar. Trotzdem wurden die Schriftarten Pica (80 Zeichen/Zeile), Elite (96 Zch/Zl) und Engdruck (136), NLQ (80), Kursiv/NLQ (80) und Fettdruck auf die Schalter-Leiste an der Vorderseite gelegt. Auch der HEX-Dump-Modus läßt sich von hieraus direkt einschalten. Leichte Entäuschung verursacht die Suche nach den DIP-Schaltern. Sie wurden auf der

Rückseite unterhalb der Papierführung angebracht. Schade!

Wer Größeres vorhat, kann auf das Flaggschiff der STAR-Flottille umsteigen, den SR-15. Ohne Interface ist jedoch beim C64 nichts zu machen, da der SR-15 einen Centronics-Eingang hat. Wir erprobten ein Wiesemann-Interface mit guten Ergebnissen. Der SR-15 ist mit Breitpapier und Hochgeschwindigkeit (200 Zeichen pro Sekunde) und 16 kByte Puffer besonders für umfangreiche Listings interessant. Bei meterlangen Assembler-Sources erspart einem der SR-15 übermäßig lange Wartezeiten. Der Drucker gehört in die Klasse der professionellen Geräte und stellt, neben einem 'STAR'-Zeichensatz, zwei IBM-kompatible Zeichensätze bereit. Die mechanische Ausrüstung ist ausgereift. Einzig die Phon-Zahl könnte durch Verringern überflüssiger Resonanzräume noch ein wenig gesenkt werden. Dies Gerät kommt am ehesten für Leute in Frage, die viel 'Druck' auszuhalten haben, und bleibt auch bei Anschaffung einer anderen Anlage einsatzbereit.

Die Preise

SG10C	incl. C64-Interface	995,- DM
NL10	incl. 1 Interface-Modul	1145,- DM
	Modul einzeln	150,- DM
SR15	Centronics	2150,- DM
	Interface für C64	250,- DM

Unser Eindruck:

Druckertyp	SG10C	NL10	SR15
Schriftbild	+	++	+
Geschwindigkeit	+	+	++
Bedienung	–	++	+
mech. Aufbau	+	++	+
Preis	+	++	
Systemabhängigkeit	–	+	+

Spiel

Scrolling

Dieses Irrgarten-Spiel verlangt von Ihnen eine Menge an Konzentration und Durchhaltevermögen. Ziel des Spieles ist es, 26 Zielsteine zu bekommen. Doch bevor Sie diese abschießen können, haben Sie noch zwei weitere Bedingungen zu erfüllen. Sie müssen 99 Augen gefunden haben und nebenbei noch 500 000 Punkte sammeln. Doch leichter gesagt als getan: eine Menge von Hindernisse wollen überwunden werden.

Spielbeginn

Innerhalb des Magazines startet das Spiel automatisch. Laden Sie es von Ihrem eigenen Datenträger, starten Sie es einfach mit RUN. Die Anfangslaufschrift erinnert Sie noch einmal an Ihre Aufgabe. Mit einem beliebigen Tastendruck kommen Sie weiter. Nehmen Sie nicht die Leertaste, denn sonst könnte es passieren, daß Sie die Legende der Spielsteine und Punkte überspringen. Zu den 16 aufgeführten Symbolen noch einige Erklärungen:

Symbol 1 Scrollfelder **Abschuß = 15 Punkte**

An die Stelle der abgeschossenen Scroll-Felder werden im Laufe des Spieles Begrenzungssteine (Symbol 5) gesetzt.

Symbol 2 Diamanten **Abschuß = 25 Punkte**

Das im Kontrollfeld angezeigte Zeitlimit wird wieder auf 9999 gesetzt, eine eventuell eingeschaltete Weitschußmöglichkeit wird ausgeschaltet.

Symbol 3 Wand **Abschuß = 30 Punkte**

Die Wand kann logischerweise nicht überall abgeschossen werden, da hilft im Bedarfsfall nur aussprobieren. Fehlende Wandsteine werden im Laufe der Zeit wieder ersetzt. Vorsicht! Lassen Sie sich nicht einschließen.

Symbol 4 Gelbe Steine **Abschuß = 33 Punkte**

Diese Steine werden auch wieder ersetzt, sie setzen die Zeit auf 9999 und schalten den Weitschuß aus.

Symbol 5 Begrenzungssteine
Abschuß = 45 Punkte

Diese Steine sind entweder schon vorhanden, oder werden an Stelle der abgeschossenen Scrollfelder gesetzt. Sie können nicht ohne weiteres abgeschossen werden. Ein Treffer wandelt sie in Blaue Steine (Symbol 6) um. Um die Umwandlung durchführen zu können, muß allerdings vorher eine Weiße Figur (Symbol 11) abgeschossen werden. Eine getroffene Weiße Figur ermöglicht die Umwandlung von 10 Steinen, mehrere Weiße erhöhen diese Zahl nicht!

Symbol 6 Blaue Steine Abschuß = 60 Punkte

Diese Steine kennen Sie ja schon. Blaue Steine können direkt abgeschossen werden, allerdings nur, wenn Sie vorher ein Weißes Dreieck (Symbol 10) getroffen haben. Nach 7 Blauen müssen Sie wieder nach einem Weißen Dreieck Ausschau halten.

Symbol 7 Blaue Dreiecke Abschuß = 80 Punkte

Hier nun das Symbol, das Ihnen die Weitschüsse garantiert, die Sie unbedingt brauchen, um das Spiel zu meistern. Das Zeitlimit wird zurückgesetzt.

Symbol 8 Augen Abschuß = 00 Punkte

Abgeschossene Augen bringen zwar keine Punkte, doch brauchen Sie alle 99 im Spiel verteilten Augen, um die Zielsteine zu erreichen. Abgeschossene Augen werden in Weiße Dreiecke (Symbol 10) umgewandelt. Jedes abgeschossene Auge verändert die Geschwindigkeit der Fahrstühle.

Symbol 9 Tore / Begrenzer Abschuß = 90 Punkte

Die Sperre (Symbol 14) muß blinken, damit diese Steine abgeschossen werden können. Auch diese Steine verschwinden nicht dauerhaft, sondern werden wieder ersetzt.

Symbol 10 Weiße Dreiecke Abschuß = 111 Punkte

Kennen Sie ja auch schon. Ein abgeschossenes Weißes Dreieck gibt Ihnen die Möglichkeit, 7 Blaue Steine (Symbol 6) abzuschießen.

Symbol 11 Weiße Figur Abschuß = 130 Punkte

Ebenfalls bekannt. Ein Abschuß garantiert Ihnen eine Umwandlung von 10 Begrenzungssteinen (Symbol 5) in Blaue Steine (Symbol 6). Der Weitschuß wird ausgeschaltet, das Zeitlimit zurückgesetzt.

Symbol 12 Senkrechte Strahlenbarriere

Dieses Symbol kann nicht abgeschossen werden. Es versperrt zeitweise den Weg – und unter Umständen wird die Spielfigur erfaßt und quer durch's Spiel befördert.

Symbol 13 Waagerechte Strahlenbarriere

Mit diesen Sperren verhält es sich genauso wie mit dem Symbol 12.

Symbol 14 Sperre

Diese Sperre wird beim Abschuß von Kleeblättern (Symbol 16) abwechselnd ein- und ausgeschaltet. Blinkt die Sperre, wird beim Abschuß eines Kleeblattes der gerade bestehende Zustand eingefroren. Das nächste getroffene Kleeblatt läßt die Sperre wieder blinken.

Symbol 15 Sperre

Durch diese Sperre kann horizontal hindurchgeschossen werden, um dahinter befindliche Steine oder Symbole zu erreichen. Ein eingeschalteter Weitschuß ist hierbei oft von Nutzen. In Abhängigkeit von der Anzahl der abgeschossenen Blauen Dreiecke werden Teile dieser Sperre gelöscht.

Symbol 16 Kleeblätter Abschuß = 1000 Punkte

Es ist zwar ein weiß-rotes Kleeblatt, doch es hat es in sich. 1000 Punkte erhöhen das Erfolgserlebnis und bringen einem dem Ziel bedeutend näher. Die Sperre des Symbols 14 wird ein- und ausgeschaltet. Und ein Abschuß setzt den Energiebalken im Kontrollfeld wieder auf den Maximalwert.

Ihr Punktestand beträgt anfangs (wie sollte es auch anders sein) Null Punkte. Der Energiebalken wird erst beim Aufbau des Spielfeldes eingeschaltet. Die Augenzahl steht auf 99. Sie zeigt Ihnen, wieviele Augen noch bis zum Ziel fehlen.

Jetzt geht's los . . .

Während Sie noch die Legende der 16 Symbole betrachten, wird höchstwahrscheinlich die Zeit bis auf Null abgelaufen sein. Nach dem Start mit der Leertaste wird die Legende ausgeblendet und es erscheint die Anzeige 'Nur noch 8 Leben'. Betrug denken Sie nun sicherlich, schon eins von Ihren neun Leben verloren. Aber bei den nächsten Spielen werden Sie sicherlich nicht mehr soviel Zeit für das Lesen der Legende benötigen. Und rechtzeitiges Drücken der Leertaste sorgt dafür, das Sie Ihre 9 Leben bekommen und die Zeit wird auf 9999 gesetzt – die Welt ist wieder in Ordnung!

In der Mitte sehen Sie Ihre Spielfigur, der Pfeil gibt die jeweilige Bewegungs- und Schußrichtung an. Nur wenn Sie sich nach unter bewegen, 'lächelt' die Figur Sie an. Nach Unten kann nicht (!) geschossen werden. Steuern können Sie die Figur mit einem Joystick in Port 2. Oder Sie steuern mit den Tasten 'A', 'Z' für die Richtungen oben/unten, '.', '/' für links/rechts und die Leertaste für Feuer. Nun können Sie sich auf Augen- und Punktejagd begeben.

Ihr Aktionsradius umfasst 18 Bildschirme, von Ihrer Ausgangsposition aus zwei nach links und 16 nach rechts. Die begehrten Zielsteine befinden sich am ganz äußersten rechten Spielfeldrand. Sollten Sie es nicht gleich bei den ersten Spielen schaffen, bis dorthin vorzudringen, können Sie diese auch schon einmal anschauen, indem Sie ganz nach links gehen. Beim Einsammeln der Zielsteine wollen noch ein paar Überraschungen überwunden werden. Doch haben Sie es bis dorthin geschafft, sollte das kein ernstzunehmendes Problem mehr darstellen.

Eigenleben

Manchmal werden Sie das Gefühl haben, daß sich das Spielfeld von alleine bewegt. Und richtig, das Spiel führt ein Eigenleben, das Ihnen oft schwer zu schaffen machen wird. Das Spielfeld bewegt sich immer ein Stück in die eine und dann wieder in die andere Richtung. Und wird Ihre Spielfigur in so einem Moment von einer Wand oder von Spielsteinen erfasst, werden Sie ein Stück durch die Gegend transportiert. Der Transport dauert solange, bis die Spielfigur keine Kollision mehr erzeugt. Oftmals ist dieser Transport die einzige Möglichkeit weiterzukommen, doch Vorsicht ist angebracht. Jeder Transport verbraucht Energie, und ist der Energiebalken abgelaufen, verlieren Sie ein Leben. Je mehr Augen Sie schon haben, desto mehr Energie verbraucht ein Transport. Gegen Ende des Spieles kann es passieren, daß ein Transport von 10 cm Sie ein ganzes Leben kostet. Dieser Auto-Transport passiert auch, wenn Sie sich in die senkrechten oder waagerechten Strahlenbarrieren begeben. Sie werden immer in die entgegengesetzte Richtung transportiert.

Nicht verzweifeln, wenn nicht alles auf Anhieb klappen sollte. Sie wissen doch: Übung macht den Meister, auch beim Spielen.

• • •

Mathe mit Nico

Wahrscheinlich liegt die Wahrscheinlichkeit, daß Niko sich diesmal mit der Wahrscheinlichkeits-Rechnung beschäftigt, bei mindestens 150 Prozent. Die Wahrscheinlichkeit, daß er beim Würfeln bereits beim ersten Mal zwei Sechsen hinlegt, ist mit Sicherheit wesentlich geringer.

Ihnen steht bekanntlich eine Rechenseite zur Verfügung, auf der Sie alle Rechenoperationen ausführen können, die Ihnen vom Direkt-Modus her bekannt sind. Diese Rechenseite können Sie mit der Taste R und RETURN aufrufen, wenn Sie bei der Aufgabenstellung zu einer Eingabe aufgefordert werden.

ID - Werkstatt

Wir stellen Ihnen diesmal zwei Programme aus dem Bereich der Unterhaltung vor. Bevor wir aber zu diesen Programmen einige Anmerkungen machen, wollen wir noch einmal auf zwei wichtige Sachverhalte hinweisen. Erstens: Die Programme der ID-Werkstatt sind nicht innerhalb von INPUT 64 lauffähig. Das Auswahlmenü ermöglicht Ihnen 'nur' ein gezieltes Abspeichern auf Ihren eigenen Datenträger. Zweitens: Wenn Sie das Programm von Ihrem Datenträger (selbstverständlich außerhalb von INPUT 64) wieder in den C 64 laden, müssen Sie das Programm als normales BASIC-Programm (also mit ,1 oder mit ,8 und nicht mit ,1,1 oder ,8,1) laden. Beachten Sie bitte, daß zum Beispiel speed-dos die Programm normalerweise mit 8,1 läd.

Da wir uns bei dieser Rubrik in erster Linie als Mittler verstehen, wollen wir die Autoren nun selbst zu Wort kommen lassen.

Die Autorin von Hires Painter beschreibt Ihr Programm wie folgt: "... Mit Hires Painter kann direkt auf dem Grafikschirm gezeichnet werden. Als 'Stift' dient ein Sprite (Dreieck), welches über die CRSR-Tasten gesteuert wird. ..." Das Programm bietet viele Optionen, die im Eingangs-Menü aufgeführt sind. Die Struktur des Programms erlaubt es, ohne größeren Aufwand eigene Ergänzungen einzubinden. Sie müssen aber beachten, daß Sie vor dem Laden von Hires Painter unbedingt INPUT-BASIC aus der Ausgabe 1/86 laden müssen.

Der Autor des Programms Direct Soundcontrol schreibt unter anderem: "... Nachdem ich meine Kenntnisse in BASIC stark verbessert hatte, begann ich mit der Programmierung. Immer wieder verbesserte ich mein Programm, und nannte es schließlich Direct Soundcontrol. Direct Soundcontrol ist ein direkt bespielbarer

3-Stimmiger-Synthesizer ... Alle Register zum Einstellen der Hüllkurve und so weiter sind wie bei einem Equalizer gut zu übersehen. Trotz des BASICs ist die Tastenabfrage schnell genug. . ." Uns bleibt an dieser Stelle nur noch, wieder darauf hinzuweisen, daß Werkstatt-Produkte von der Redaktion nicht betreut werden. Wir können daher auch keine Fragen zur Bedienung, zu Programmfehlern oder Ähnlichem beantworten.

64er Tips

von Frank Börncke

Diesmal: Die USR(X)-Funktion

Thema dieses Monats ist die USR(X)-Funktion des BASIC-Interpreters. Auch denjenigen, die BASIC sonst perfekt beherrschen, ist diese Funktion häufig ein Rätsel. Es soll im Folgenden erklärt werden, wozu sie gut ist, wie sie funktioniert und wie man sie nutzbringend für eigene Zwecke einsetzen kann.

Allgemeines:

SYS und USR – Sinn und Zweck

Mit der USR(X)-Funktion können Sie von BASIC aus einen Wert an ein Maschinen-Programm übergeben. Dies kann aus Ihrer eigenen 'Programm-Küche' stammen oder auch direkt aus dem Betriebssystem. Der für X eingesetzte Wert wird dann im Maschinen-Programm verarbeitet und das Ergebnis wieder an das BASIC-Programm übergeben.

Sie können natürlich mit Recht sagen, daß der SYS-Befehl dazu auch in der Lage ist. Jedoch muß die Übergabe von Werten an ein Maschinen-Programm beim SYS-Befehl innerhalb des Maschinen-Programmes selbst geregelt werden, was durch Aufruf entsprechender System-Routinen (CHRGET, CHKKOM, GETBYT, etc. (4)) geschieht. Das kostet Zeit, Speicherplatz und den Fleiß des Programmierers. Aber warum so schwierig, wenn es auch einfacher geht?

Bei der USR(X)-Funktion erledigt das Betriebssystem die Wertübergabe. Doch das ist noch nicht alles: Mit USR(X) ist es zusätzlich noch möglich, eigene mathematische Funktionen zu entwerfen, die sich dann ohne Probleme in umfangreichen Formeln einsetzen lassen, wie das hier gezeigt ist:

A = 3*SIN(1+USR(X/10))+PI

Die Syntax von USR(X) entspricht also der Anwendungsweise der üblichen BASIC-Funktionen (siehe Handbuch (2),(3),(4)). Solch eine Anwendung ist mit dem SYS-Befehl nicht möglich. Wie Sie sehen, scheint es lohnenswert zu sein, sich mit dieser Funktion einmal genauer zu beschäftigen.

Arbeitsweise der USR(X)-Funktion

Ein Ausflug in die Fliesskomma-Arithmetik

Es gibt im Computer zwei Fließkomma-Akkumulatoren (kurz FAC und ARG genannt), über die alle Berechnungen im Betriebssystem ablaufen. Zum besseren Verständnis könnte man FAC und ARG auch als Variablen auf Maschinensprache-Ebene bezeichnen. Beide Akkumulatoren belegen Bereiche in der Zero-Page ('Null-Seite' bei den Adressen von $0000 bis $00FF), FAC von 97-102 ($0061-$0066), ARG 105-110 ($0069-$006E).

Das Fließkomma-Format ist eine dem C64 eigene Methode, rationale Zahlen (Zahlen mit Nachkomma-Stellen) und Berechnungsergebnisse im Speicher abzulegen. Wer mehr wissen möchte, sei auf den Literatur-Hinweis am Ende des Artikels verwiesen. ((3),(5))

Enthält der FAC einen Wert, kann man z.B. durch Aufruf der SINUS-Funktion (SYS 57963 in BASIC / JSR $E26B in ASSEMBLER) den Sinus vom FAC berechnen. Das Ergebnis wird dann in den FAC zurückgeschrieben, wo es zur weiteren Verwendung zur Verfügung steht. Auf diese Weise ist es also möglich, durch gezielten Aufruf der Betriebssystem-Routinen eigene mathematische Funktionen zu entwickeln. Wie oben bereits geschildert, erfolgt die Übergabe der Werte dabei automatisch mit dem Aufruf der Funktion. Das Argument von USR(X), in diesem Fall also der X-Wert, wird dann in den FAC übertragen. Vor dem Aufruf muß nur noch die Startadresse des Maschinen-Programms im LO/HI-Byte Format in die Adressen 785/786 ($0311/$0312) geschrieben werden.

Viele Begriffe:

LO- und HI-Bytes, sowie ein geheimnisvoller Vektor

Es ist nur halb so schlimm, wie es sich anhört. Aber nun der Reihe nach. Die LO/HI-Byte-Schreibweise ermöglicht die Darstellung eines 16-Bit-Wertes (0-65535) durch die Zerlegung der Zahl in zwei 8-Bit-Werte (0-255). Dabei erhält man ein niederwertiges und ein höherwertiges Byte (High- und Lowbyte), die man wie folgt berechnet:

HIGHBYTE = INT(ADRESSE/256)
LOWBYTE = ADRESSE-HIGHBYTE*256

Unter Vektor versteht man eine Zwei-Byte-Adresse. Dies kann zum Beispiel die Start-Adresse eines Maschinen-Programms sein. Es gibt nun im Speicher eine ganze Reihe von sogenannten Vektoren. Diese bestehen einfach aus zwei aufeinanderfolgenden Speicherzellen, die als Werte eine in LO/HI-Byte zerlegte Adresse enthalten, auf die das Betriebssytem bei Bedarf zurückgreift. Auch für die USR(X)-Funktion gibt es einen solchen Vektor, den Sie durch Verändern seines Inhaltes auf eine gewünschte Routine 'verbiegen' können. Im Normalfall (nach dem Einschalten des Computers) zeigt dieser Vektor auf die Routine, die die Fehlermeldung '?illegal quantity error' ausgibt.
Probieren Sie mal A=USR(0)!

Anwendung:

Die USR(X)-Funktion in BASIC und ASSEMBLER

Die Anwendung von USR(X) bleibt aber nicht nur den ASSEMBLER-Programmierern vorbehalten. Auch als eingefleischter BASIC-Programmierer können Sie diese Funktion benutzen. Wenn Sie anstatt selbsgeschriebener Funktionen einfach die Adressen von geeigneten Systemroutinen angeben, bekommen Sie auf diese Weise neue Funktionen, ohne in ASSEMBLER programmieren zu müssen. Doch sei gesagt, daß sich dieser Befehl erst mit den Möglichkeiten der Maschinensprache richtig ausnutzen läßt.

Hier ein Beispiel:
Ab Adresse 47842 ($BAE2) befindet sich eine Routine, die den Inhalt vom FAC mit der Zahl 10 multipliziert. Das nachfolgende Programm verbiegt den USR-Vektor auf diese Adresse und führt dann für eingegebene Werte die Berechnungen aus. Wie eine Messung ergibt, ist diese Methode erheblich schneller, als wenn man die Werte direkt mit 10 multipliziert. So ein Geschwindigkeitsvorteil macht sich besonders bei der Programmierung von Schleifen angenehm bemerkbar.

```
10 POKE785,226:POKE786,186:REM STARTADRESSE LO/HI
20 INPUT X                 :REM WERT EINGEBEN
30 PRINT USR(X)            :REM X*10 AUSGEBEN
40 GOTO 20                 :REM ENDLOSSCHLEIFE
```

Hier noch ein paar Adressen als Anregungen für ihre eigenen Experimente:

```
HEX    DEZ    LO/HI    FUNKTION
-------------------------------
$BAE2 47842 226/186 FAC = FAC * 10
$BAFE 47870 254/186 FAC = FAC / 10
$B849 47177 073/184 FAC = FAC + 0.5
```

Nützliche Anwendungen für diese Routinen mag jeder selber finden (Siehe Literatur-Hinweise).

Natürlich ist es auch möglich, die vorhandenen BASIC-Funktionen über USR(X) aufzurufen, um damit z. B. SQR(X) zu ersetzen. Wenn Sie den USR-Vektor mehrmals im Programm verändern, können Sie damit ihre Programme für Unbefugte undurchschaubarer machen, weil ein Uneingeweihter nicht direkt erkennen kann, welche Funktion gerade mit USR(X) gemeint ist. Die Adressen dieser BASIC-Funktionen können Sie in jedem gut dokumentierten ROM-Listing finden ((1),(3),(4)).

Es ist auch denkbar, den USR-Vektor auf eine Adresse, z.B. 49152 ($C000) zu setzen, um mit PRINT USR(X) ein dort liegendes Maschinenprogramm (zum Beispiel einen Monitor) zu starten. Dabei ist es vollkommen egal, welchen Wert Sie für X wählen. Dieser Wert wird zwar in den FAC übertragen, hat aber auf die Funktion des Monitors keinen Einfluß.

Der Inhalt des USR-Vektors bleibt auch nach einem RUN STOP/RESTORE erhalten. Eine einmal geänderte Startadresse brauchen Sie sich also nicht mehr zu merken oder in einer Varibalen abzulegen, wie das bei dem SYS-Befehl der Fall wäre.

Als weiteres Bespiel und als Anregung finden Sie auf dem Datenträger ein kleines ASSEMBLER-Programm als BASIC-Lader, das es Ihnen ermöglicht, Gradmaß in Bogenmaß umzurechnen. Die Handhabung der Winkelfunktionen fällt damit etwas leichter. Das Programm ist so gestaltet, daß Sie es sich nach eigenen Wünschen frei im Speicher verschieben können. Zudem ist es mit einer Länge von nur 20 Bytes

angenehm kurz gehalten. Auf diese Weise dürfte sich die Routine an so ziemlich alle Bedürfnisse anpassen lassen. Bei der Programmierung wurde die folgende Formel umgesetzt:

$$\text{Bogenmaß} = \frac{\text{Gradmaß} * PI}{180} = \text{Gradmaß} * PI * 0.555555$$

Als Maschinen-Programm sieht das dann so aus:

```
C000 LDA #$A8   ;Adresse von der Zahl PI im
C002 LDY #$AE   ;Betriebssystem(Lo/Hi-Byte Format)

C004 JSR $BA28 ;FAC = FAC * PI

C007 LDA #$0F   ;Adresse des Faktors 1/180
C009 LDY #$C0   ;im Speicher (Lo/Hi-Byte Format)

C00B JSR $BA28 ;FAC = FAC * 1/180

C00E RTS        ;Zurück nach BASIC - FAC wird
                ;an Variable übergeben

C00F $79,$36,$0B,$60,$B6 ;1/180 = 0.55555555 im
                         ;Fliesskommaformat
```

Die Anwendungsweise dieser Routine ist denkbar einfach. Will man z. B. den COSINUS von 67 Grad berechnen, sieht das so aus:
PRINT = COS(USR(67))
Eine Umrechnung der Winkelmaße mit einem gesonderten Unterprogramm ist dann nicht mehr notwendig. Außerdem macht sich auch hier der bereits erwähnte Geschwindigkeitsvorteil positiv bemerkbar.

Eine Sache bleibt noch zu erwähnen:
Da das Programm die mathematischen Routinen des BASIC-Interpreters benutzt, ist es den üblichen Rechenungenauigkeiten und Rundungsfehlern des Computers ausgesetzt. Wenn Sie also eine 1 erwarten und 0.99999999 erhalten, ist das keine Schlamperei im Programm, sondern ein Fehler im Betriebssystem ihres Commodore 64.
(Man möge es ihm verzeihen.)

Fazit:

Nichts gegen SYS, aber gebt USR eine Chance!

Wie Sie sehen, kann man mit dieser unscheinbaren, eigentlich zu Unrecht vernachlässigten Funktion eine ganze Menge anstellen. Es zeichnen sich hier aber auch sehr deutlich die Grenzen der Programmiersprache BASIC ab. In Verbindung mit ASSEMBLER bietet die USR(X)-Funktion sicher noch viele ungeahnte Möglichkeiten, die nur darauf warten, entdeckt zu werden. Vielleicht ein Anlaß, sich doch einmal näher mit Maschinensprache zu beschäftigen? Besonders gelungene Funktionen können Sie gerne an die Redaktion einsenden. Wir können Sie dann in der Rubrik "ID-Werkstatt" der Öffentlichkeit vorstellen.

Literatur:

(1) 64 Intern
Angerhausen/Brückmann/Englisch/Gerits
DATA BECKER Düsseldorf 1983
(2) 64 Tips & Tricks
Angerhausen/Englisch/Gerits
DATA BECKER Düsseldorf 1983
(3) C-64 Computer Handbuch
Raeto West (Engl. Autor)
te-wi Verlag München 1985
ISBN 3-921803-24-1
(4) Alles über den C64
Commodore Frankfurt 1984
ISBN 3-89133000-6
(5) Das Maschinensprache Buch
für Fortgeschrittene
Englisch
DATA BECKER Düsseldorf 1984
ISBN 3-89011-022-3

Hilfsprogramm

Hires-Hardcopy für MPS 801

Was lange währt, wird endlich gut: ein Grafik-Hardcopy-Programm für den Commodore-Drucker MPS 801 ist das von unseren Lesern mit Abstand meistgewünschte Hilfsprogramm. Es hat zwar etwas gedauert, bis wir diesen Wunsch erfüllen konnten, dafür ist es aber auch besonders schnell und komfortabel. Der Autor, Andreas Gauger, hat sich nämlich einige tempositeigernde Überlegungen gemacht.

Das Prinzip

Zum Ausdruck hochauflösender Graphik muß der Drucker auf Einzelnadel-Steuerung (Bit-Image-Mode) umgeschaltet werden. Der Trick besteht darin, daß nur im Bit-Image-Mode gedruckt wird, wenn es unbedingt sein muß. Das heißt, das Programm prüft, ob überhaupt in der zu sendenden 7*6-Matrix ein Punkt gesetzt ist. Wenn nicht (ist dieses Feld also leer), wird nur ein Leerzeichen im Text-Modus gedruckt. Und besteht die ganze Zeile nur aus Leerzeichen, wird schlicht ein Carriage Return (CHR$(13)) an den Drucker geschickt. Dadurch variiert die Zeit, die für den Ausdruck einer kompletten Hires-Seite benötigt wird. Beim Testen in der Redaktion maßen wir Werte zwischen 12 Sekunden und (maximal) 3 Minuten (bei Normalgröße).

Die Bedienung

Das Programm kann wie gewohnt mit CTRL & s auf den eigenen Datenträger überspielt werden. Wenn Sie es von dort wieder geladen haben, starten Sie es mit RUN. Es geschieht dadurch augenscheinlich gar nichts. (Außer eventuell der Meldung "SYNTAX ERROR", diese Fehlermeldungen können Sie aber ignorieren.) Es wird nämlich zunächst nur der BASIC-Anfang hochgesetzt. Geben Sie nach dem ersten Laden und Starten NEW ein. Danach kann ein beliebiges BASIC-Programm nachgeladen oder neu eingegeben werden. Sie können die MPS 801-Hardcopy wahlweise zusammen mit diesem BASIC-Programm abspeichern. Setzen Sie dazu vor dem SAVE-Befehl den BASIC-Anfang durch POKE 44,8 wieder auf Normal-Werte.

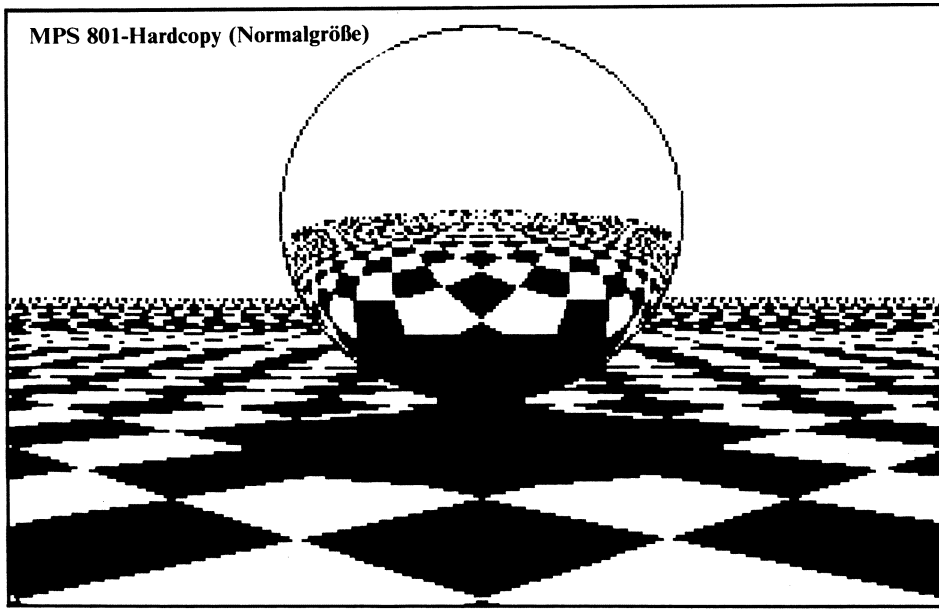
Aufgerufen wird das Hardcopy-Programm durch:

SYS2400,be,gr

be ist eine Zahl zwischen 2 und 14, die in 4-KB-Schritten die Anfangsadresse des Graphik-Bildschirms festlegt. Eine 2 entspricht $2000 (dezimal 8192, ein sehr häufig benutzter Wert); ist be = 14, wird die auszudruckende Graphik bei $E000 (dezimal 57344) erwartet, dort legt zum Beispiel INPUT-BASIC (aus Ausgabe 1/86) das hochauflösende Bild ab. Wenn Sie sich nicht ganz sicher sind, versuchen Sie es mit de = 20, dann ermittelt das Programm die Anfangsadresse des eingeschalteten (!) Graphik-Bildschirms.

gr bestimmt die Größe des Ausdrucks. gr = 0 steht für Normalgröße (13.5 * 8 cm); ist gr = 1, wird's viermal so groß (26 * 17 cm)

MPS 801-Hardcopy (Normalgröße)



Spiel

Füll den Kreis

Eigentlich ist zu diesem Spiel nicht viel zu sagen, den den meisten wird es bekannt sein. Sicher haben Sie es schon einmal unter der Schulbank oder am Biertisch gespielt, mit Streichhölzern, Knöpfen oder ähnlichem.

Sie können zu zweit spielen oder, falls Ihnen ein Spielpartner fehlt, auch alleine gegen den Computer. Abhängig von der gewählten Spielstärke werden 15, 26 oder 36 Kreise vorgegeben. Abwechselnd werden jetzt die Kreise ausgefüllt. Mindestens ein Kreis muß ausgefüllt werden, und abhängig von der Spielstärke können es maximal drei, vier oder fünf sein. Ziel des Spiels ist es, die Kreise so geschickt zu füllen, daß der Gegenspieler gezwungen ist, den letzten Kreis zu nehmen.

c't-Uhr

Wir standen vor einer schweren Entscheidung! Wie so oft bei technisch ausgeklügelten Entwicklungsarbeiten, tauchten weitere kleine Probleme beim Betrieb der Echtzeit-Uhr aus c't 4/85 im C64 auf. Wie in der letzten Ausgabe bereits erwähnt, stellen der spezifische Zugriff des 6502 und Störimpulse auf den Bus-Leitungen im Rechner das eigentliche Problem dar. Mit anderen Worten: Wird die Echtzeit-Uhr über ein ROM gesetzt (zum Beispiel BASIC-ROM), fühlt sie sich durch die Störimpulse immer wieder berechtigt, ihre Informationen ins ROM einzublenden. Greift der Interpreter gerade auf den kritischen Bereich zu, sind die Auswirkungen nicht mehr vorhersehbar. Sie reichen vom schlichten Absturz bis hin zu absurden Fehlermeldungen und Rechenergebnissen.

Wir gehen davon aus, daß der C64 durch die Uhr in seiner ursprünglichen Funktionsweise nicht verändert werden darf. Soweit sind die Entwicklungsarbeiten jedoch noch nicht gediehen. Für Maschinen-Programme reinsten Wassers läßt sich die Uhr bereits verwenden. Zugriffe ins BASIC-ROM sind jedoch kritisch. Der versuchsweise Umzug zum Character-ROM brachte keine sinnvollen Ergebnisse; der VIC hat hier halt absolute Vorfahrt.

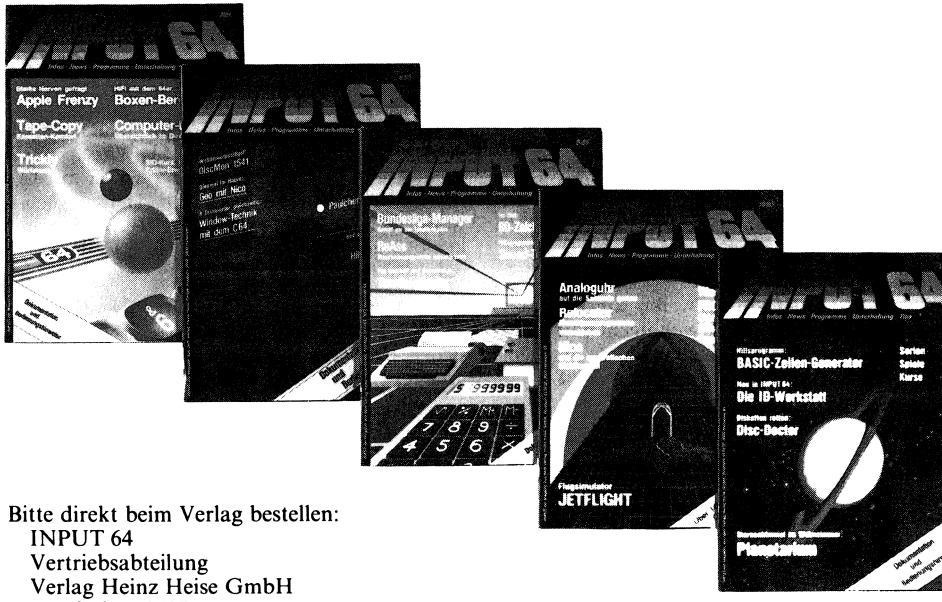
Sie dürfen also gespannt sein, wie der Wettstreit mit den Marotten des C64 ausgeht. Nur eins ist sicher: Die Echtzeit-Uhr muß, wenn sie im C64 arbeitet, störungsfrei laufen!

Im Six-Pack und Solo

Wegen der großen Nachfrage haben wir bereits vergriffene Ausgaben von INPUT 64 nachproduziert, so daß alle bisher erschienenen Ausgaben wieder lieferbar sind. Ab Ausgabe 4/85 ist INPUT 64 auch auf Diskette erhältlich. Preis: Kassettenversion 12,80 DM / Diskettenversion 19,80 DM. (jeweils incl. Porto und Verpackung)

Kassettenversion ab Ausgabe 1/86: 14.80 DM

Außerdem können Sie die Diskettenversion der Ausgaben 4/85 bis 9/85 im Sechser-Pack beziehen. Komplettpreis: 90 DM. Sie sparen: 24,80 DM!



Bitte direkt beim Verlag bestellen:
INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 610407
3 Hannover 1
(Lieferung nur gegen Vorkasse, bitte Verrechnungsscheck beilegen)

Aus dem Inhalt

Ausgabe 1/85 – Dateiverwaltung, drei (!) Spiele * Ausgabe 2/85 – Textprogramm, Zeichensatzeditor * Ausgabe 3/85 – Spriteeditor, Maschinensprache-Monitor * Ausgabe 4/85 – SuperTape D II, Grafikerweiterung, Urlaubskalender * Ausgabe 5/85 – Mathe mit Nico (Teil 1), Talk to me (Dialogsimulation), Hintergrundmonitor * Ausgabe 6/85 – Textadventure "Fuchsjagd", SID-Kurs (Teil 1), Recorder-Justage, BASIC-Compactor * Ausgabe 7/85 – HiFi-Boxen-Berechnung, Tape-Copy, Scroll Editor * Ausgabe 8/85 – Discmonitor, Reisekostenberechnung, Musik-Harware * Ausgabe 9/85 – Reassembler, Bundesliga-Simulation * Ausgabe 10/85 – Flugsimulator, Maschinensprache-Relocator * Ausgabe 11/85 – Planetarium, Datei-Kopierprogramm, DiscDoctor * Ausgabe 12/85 – Funky Drummer, Kosten-Nutzen-Analyse, Hardcopy-Routine für (fast) alle Drucker * Ausgabe 1/86 – Über 40 neue Befehle: INPUT-BASIC, Maskengenerator "TextMagic", Lohnsteuer-Berechnung *

Hinweise zur Bedienung

Bitte entfernen Sie eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren Rechner einmal kurz aus. Geben Sie nun zum Laden der Kassette *LOAD* und *RETURN* oder *SHIFT* und gleichzeitig *RUN/STOP* bzw. der Diskette *LOAD"INPUT*",8,1* und *RETURN* ein. Alles weitere geschieht von selbst.

Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins. Dieses können Sie nun mit der *SPACE* (Leertaste) durchblättern. Mit *RETURN* wird das angezeigte Programm ausgewählt. Im Fenster unten rechts erhalten Kassettenbesitzer weitere Hinweise ("Bitte Band zurückspulen" und so weiter . . .). Haben Sie bei der Auswahl eines Programms eventuell nicht weit genug zurückgespult, und es wurde nicht gefunden, spulen Sie bis zum Bandanfang zurück. Diskettenbesitzer stellen bitte sicher, daß noch die INPUT 64-Diskette eingelegt ist. Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie. Sollten Sie eventuell mit einem der Programme Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

Neben der Programmauswahl mit *SPACE* und dem Ladebefehl mit *RETURN* (im Inhaltsverzeichnis) werden die übrigen 'System-Befehle' mit der Kombination aus *CTRL*-Taste und einem *Buchstaben* eingegeben. Sie brauchen sich eigentlich nur *CTRL* und *H* zu merken (Aufruf der Hilfsseite), denn dort erscheinen die jeweils möglichen weiteren 'System-Befehle'. Nicht immer sind alle Optionen möglich (eventuell werden Sie zu Beginn des Programms auf Einschränkungen hingewiesen). Hier nun alle INPUT 64-Systembefehle:

CTRL und *Q* (ab Ausgabe 3/85)

Sie kürzen die Titelgrafik ab; INPUT 64 geht dann sofort ins Inhaltsverzeichnis.

CTRL und *H* (ab Ausgabe 1/85)

Es wird ein Hilfsfenster angezeigt, auf dem alle verfügbaren Befehle aufgeführt sind.

CTRL und *I* (ab Ausgabe 1/85)

Sie verlassen das Programm und kehren in das Inhaltsverzeichnis zurück.

CTRL und *F* (ab Ausgabe 1/86)

Ändert die Farbe des Bildschirm-Hintergrundes (auch im Inhaltsverzeichnis erreichbar).

CTRL und *R* (ab Ausgabe 1/86)

Ändert die Rahmenfarbe (auch im Inhaltsverzeichnis erreichbar).

CTRL und *B* (ab Ausgabe 4/85)

Sie erhalten einen Bildschirmausdruck – natürlich nicht von Grafikseiten oder Sprites! Angapaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Das Programm wählt automatisch die richtige Geräteadresse (4, 5 oder 6) aus.

Fortsetzung Seite 30

Hinweise für Autoren

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise: Selbstverständlich können Sie uns Ihr Programm nur anbieten, wenn Sie es selbst erstellt haben und das Programm noch nicht veröffentlicht wurde. Ihr Programm sollte in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können Sie die bisher in INPUT 64 erschienenen Tools (PRINT AT, INKEY, Hiresspeed und die Sprite-Befehle) benutzen, wobei Ihr Programm aber insgesamt nicht länger als 100 Blöcke (25 KByte) sein sollte. Das Programm muß auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht. Es gibt außerdem einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse: 1. Belegen Sie nur den Bereich des normalen BASIC-RAM ($0801-$9FFF) und unter dem BASIC-ROM ($A000-$BFFF). 2. Das Programm muß als BASIC-File zu laden und mit RUN zu starten sein. 3. Die CTRL-Taste darf nicht benutzt werden.

Aber auch wenn Ihr Programm zur Zeit diese Anforderungen nicht erfüllt, sprechen Sie uns ruhig an. Bei ausgefallenen Programmentwicklungen sind wir gerne bereit, bei der Anpassung behilflich zu sein. Senden Sie uns Ihr Programm auf Kassette oder Diskette mit einer Programmbeschreibung und notieren bitte auf allen Einzelteilen Ihren Namen und Ihre Anschrift. Sowohl Auto-Start als auch List-Schutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlich erschwert wurde, zukünftig ungeprüft zurücksenden.

CTRL und *S* (ab Ausgabe 1/85)

Wenn das Programm zum Sichern vorgesehen ist, erscheinen weitere Hilfsfenster. Sie haben die Wahl, ob Sie:

im Normalverfahren auf Cassette	*C*
im SuperTape-Format	*S*
auf Diskette	*D*

sichern wollen.(Die SuperTape-Option ist ab Ausgabe 1/86 realisiert.) Beachten Sie bitte, daß Sie die Programme von Ihrem Datenträger immer als normale BASIC-Programme mit *LOAD "Name ",1* bzw. *LOAD "Name ",8* laden müssen.

Bei Ladeproblemen:

Schimpfen Sie nicht auf uns, die Bänder sind normgerecht nach dem neuesten technischen Stand aufgezeichnet und sorgfältig geprüft. Sondern: Reinigen Sie zunächst Tonköpfe und Bandführung Ihres Kassettenrecorders. Die genaue Vorgehensweise ist im Handbuch der Datasette beschrieben. Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

Wir haben deshalb ein Programm entwickelt, mit dessen Hilfe Sie den Aufnahme-/Wiedergabekopf justieren können. Tippen Sie das Programm JUSTAGE ein, und speichern Sie es ab. Dieses Programm wertet ein etwa 30 Sekunden langes Synchronisationssignal aus, das sich am Ende jeder Kassettenseite befindet. Starten Sie das JUSTAGE-Programm mit RUN, jetzt sollte die Meldung PRESS PLAY ON TAPE kommen, drücken Sie also die PLAY-Taste. Nach dem Drücken der Taste geht der Bildschirm zunächst wie immer aus. Wird das Synchro-Signal erreicht, wechselt die Bildschirmfarbe; und zwar – bei nicht total verstellter Spurlage – völlig regelmäßig etwa dreimal pro Sekunde. Liegt die Spur des Tonkopfes grob außerhalb der zulässigen Toleranzgrenzen, geschieht entweder nichts oder die Farben wechseln unregelmäßig. Nehmen Sie jetzt einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Datasette. Über der REWIND-Taste befindet sich ein kleines Loch. Wenn Sie bei gedrückter PLAY-Taste durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage. Drehen Sie diese Einstellschraube. Aber Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Drehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Zur Feineinstellung lassen Sie das Synchro-Signal noch einmal von Anfang an laufen. Die Schraube jetzt nach links drehen, bis der Farbwechsel unregelmäßig wird. Diese Stellung genau merken, und die Schraube jetzt langsam wieder nach rechts drehen: Der Farbwechsel wird zunächst gleichmäßig, bei weiterem Drehen wieder unregelmäßig. Merken Sie sich auch diese Stellung, und drehen Sie die Schraube nun in Mittelstellung, das heißt zwischen die beiden Randstellungen. Denken Sie daran, daß während der Einstellung kein Druck auf den Schraubenkopf ausgeübt werden darf! Der Tonkopf Ihres Recorders ist jetzt justiert.

Sollte sich auch nach dieser Einstellung INPUT 64 nicht laden lassen, erhalten Sie von uns eine Ersatzkassette. Schicken Sie bitte die defekte Kassette mit einem entsprechenden Vermerk an den Verlag ein (Adresse siehe Impressum).

PS! In der Ausgabe 6/85 haben wir das Programm RECORDER-JUSTAGE veröffentlicht, das die Einstellung des Daten-Recorders zum Kinderspiel macht.

Listing Justage

```
800 fori=49199to49410:read d:ps=ps+d:poke i,d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys49338
1000 rem von 49199 bis 49410
1010 data173, 13,220,169,217,174,  4,220,172,  5,220,141, 14,220, 48, 44, 56
1020 data102, 88, 36, 89, 48, 12,144, 10,165, 88,133, 90,169,128,133, 88,133
1030 data 91,192,121,144,  4,224,115,176,  7,169,  0,133, 92, 56,176, 11,165
1040 data 92, 73,128,133, 92, 36, 92, 16, 19, 24,102, 88, 36, 89, 48, 12,144
1050 data 10,165, 88,133, 90,169,128,133, 88,133, 91,104,168,104,170,104, 64
1060 data 96, 36, 91, 16,252,132, 91,165, 90, 96,160,128,132, 89,165, 88,201
1070 data 22,208,250,132, 88,160, 10,132, 89,132, 91, 36, 91, 16,252,132, 91
1080 data165, 90,201, 22,208,226,136,208,241, 32,133,192,201, 22,240,249, 96
1090 data 32,147,252,120, 32, 23,248,165,  1, 41, 31,133,  1,133,192,169, 47
1100 data141, 20,  3,169,192,141, 21,  3,169,127,141, 13,220,169,144,141, 13
1110 data220,173, 17,208, 41,239,141, 17,208,169, 70,141,  4,220,169,129,141
1120 data  5,220, 88, 32,142,192,201, 42,208,249,173, 32,208, 41, 15,168,200
1130 data140, 32,208, 76,237,192,208, 76

ready.
```

Am 9. Juni '86 an Ihrem Kiosk: INPUT 64 Ausgabe 6/86

Wir bringen unter anderem:

ASMED 64

Ein Assembler mit allen Funktionen, die man bei anderen Assemblern schon immer vermißt hat: eigener Full-Screen-Editor (fast eine kleine Textverarbeitung ...), blockweises Laden und Speichern von Source-Text, Include-Funktion, Macro-Verarbeitung, bedingte Assemblierung und und und ... Vor allem aber: Tempo! 120 Blöcke Quell-Code werden in weniger als fünf Sekunden assembliert!

Des LISP's dritter Teil

Zwei klassische Anwenderprogramme aus dem Bereich der Künstlichen Intelligenz: das berühmte Dialog-Programm 'Eliza' und ein Experten-System.

Die versunkene Stadt

Keine hektische Joy-Stick-Artistik, sondern Pixel-genaue Führung ist gefordert, wenn Sie sich auf die Suche nach der versunkenen Stadt begeben.

ZS-Hardcopy

Da haben Sie sich mit viel Mühe einen eigenen Zeichensatz gebastelt, aber wie jetzt die chinesischen Schriftzeichen auf den Drucker bekommen? ZS-Hardcopy bringt (sofern Ihr Drucker graphikfähig ist) jeden Bildschirm zu Papier.

und außerdem:

Mathe mit Nico, 64er-Tips zur leidigen 'Garbage Collection', Rätsel-Auflösung ...

---

c't Magazin für Computertechnik

Ausgabe 6/86 – am 15. 5. 86 am Kiosk

* Die RISC-Maschinen kommen – ausführlicher Report über die neue Prozessor-Architektur * RTOS/PEARL jetzt auch für den Atari ST * EPAC09 – ein Einplatinenrechner mit den leistungsfähigsten Achtbitter * Software-Knowhow: Aus UCSD-Pascal mach Turbo * u.v.a.m

elrad – Magazin für Elektronik

Ausgabe 6/86 – ab 26. 5. am Kiosk

* Bauanleitung: Programmierbarer Signalform-Generator * elrad-Test: Low-Cost-Oszilloskop D1010 von Siemens * Bauanleitung: Mini-Maxi-Tester – kleines Universalgenie für das Elektroniklabor * Grundlagen: Die elrad-Laborblätter – diesmal mit Leistungs-MOSFETS und typischen Schaltungen * Marktreport: Transformatoren * Schaltungstechnik: Delta-Modulation für Digital-Audio * u.v.a.m.

IMPRESSUM

INPUT 64
Das elektronische Magazin

Verlag Heinz Heise GmbH
Bissendorfer Str. 8
3000 Hannover 61
Postanschrift:
Postfach 610407
3000 Hannover 61
Tel.: (05 11) 53 52-0

Technische Anfragen
nur dienstags von 9-16.30 Uhr

Postgiroamt Hannover, Konto-Nr. 93 05-308
(BLZ 250 100 30)
Kreissparkasse Hannover, Konto-Nr. 000-01 99 68
(BLZ 250 502 99)

Herausgeber: Christian Heise

Redaktion:
Christian Persson (Chefredakteur)
Ralph Hülsenbusch
Wolfgang Möhle
Karl-Friedrich Probst
Jürgen Seeger

Ständige Mitarbeiter:
Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Eckart Steffens

Vertrieb: Anita Kreutzer-Tjaden

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 610407
3000 Hannover 61
Tel.: (05 11) 53 52-0

Grafische Gestaltung:
Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:
Reprotechnik Hannover

Druck:
Leunisman GmbH, Hannover
CW Niemeyer Hameln

Konfektionierung:
Lettershop Brendler, Hannover

Kassettenherstellung:
SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.
Einzelpreis DM 14,80
Jahresabonnement Inland Kassette DM 140.-
Diskette DM 198.-
Diskettenversion im Direktbezug: DM 16,80
+ DM 3,- Porto und Verpackung

Vertrieb (auch für Österreich, Niederlande, Luxemburg und Schweiz):
Verlagsunion Zeitschriften-Vertrieb
Postfach 5707
D-6200 Wiesbaden
Ruf (0 61 21) 2 66-0

Verantwortlich:
Christian Persson
Bissendorfer Str. 8
3000 Hannover 61

ISSN 0177-3771

Titelidee **INPUT 64**
Titelfoto: Bavaria

Titel – Grafik und – Musik:
Tim Pritlove
Fabian Rosenschein

Betriebssystem:
Hajo Schulz

hier abtrennen

INPUT 64-Abonnement

Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT64-Ausgaben ab Monat

_____

(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementsgebühren werden sofort anteilig erstattet.)

Das Jahresabonnement kostet: ○ auf Kassette DM 140,— inkl. Versandkosten und MwSt.

○ auf Diskette DM 198,— inkl. Versandkosten und MwSt.

(Bitte ankreuzen/Nichtzutreffendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

PLZ Wohnort

_____

Datum/Unterschrift

Von meinem Recht zum schriftlichen Widerruf dieser Order innerhalb einer Woche habe ich Kenntnis genommen. Zur Wahrung der Frist genügt die rechtzeitige Absendung.

Unterschrift _____

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

INPUT 64-Abonnement Abruf-Coupon

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleizahl Konto-Nr.

Geldinstitut Ort des Geldinstituts

Bankeinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

HEISE

Bitte im (Fenster-)Briefumschlag einsenden.
Nicht als Postkarte verwenden!

INPUT64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 61 04 07

3000 Hannover 61