

INPUT 64

Infos · News · Programme · Unterhaltung · Tips

Bundesliga-Manager

Kampf um die Tabellenspitze

ReAss

Maschinenprogramme entschlüsseln

Für Geschäft und Geburtstage:

Terminkalender

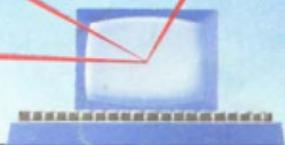
Spiele, Serien, Kurse

Im Test:

80-Zeichen-Karten

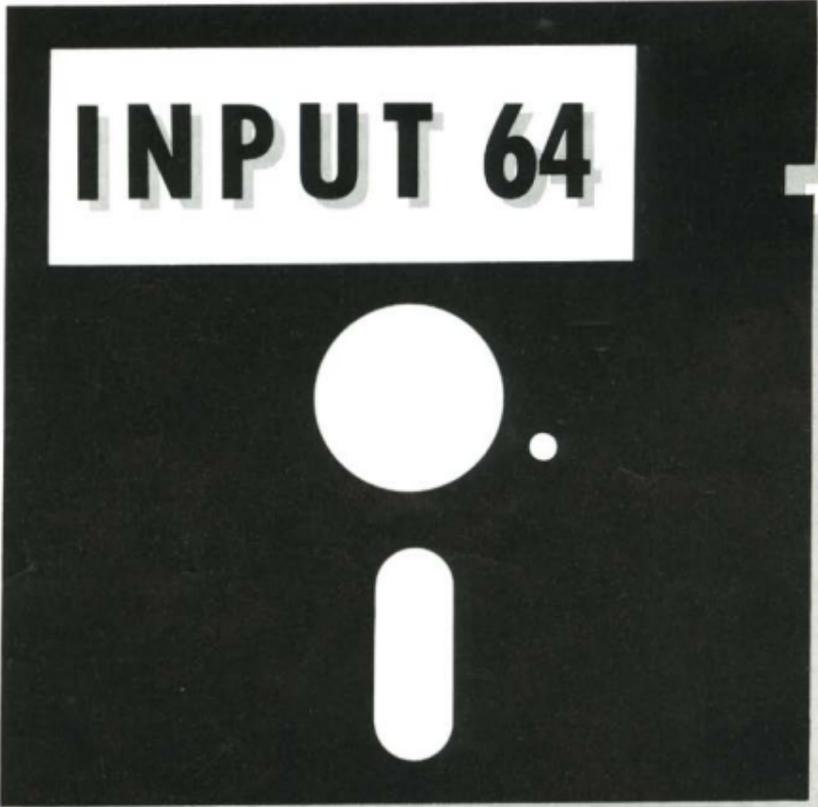
Hilfsprogramme:

Sound-Tools



Dokumentation
und
Bedienungshinweise

INPUT 64



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!

Bedienungsanleitung	Seite 2
Leser fragen. . .	Seite 4
Terminkalender	Seite 6
Superkäsi	Seite 11
Mathe mit Nico	Seite 12
Wettbewerbsgewinner: ReAss	Seite 12
News: Was bringen 80-Zeichen-Karten?	Seite 22
3000.-DM-Wettbewerb	Seite 24
64er-Tips	Seite 28
Cave Climber	Seite 29
Manager	Seite 30
SID-Kurs	Seite 33
Hinweise für Autoren	Seite 34
Hilfsprogramme: Sound-Erweiterung	Seite 35
Für Nachzügler und "Spätzünder"	Seite 40
Rätselecke	Seite 41
Bei Ladeproblemen	Seite 45
Vorschau	Seite 47

Liebe 64er-Besitzer(in),

wenn ein Teil unserer Leser uns schreibt, er interessiere sich vor allem für die Kurse in INPUT 64, andere meinen, sie kauften das Magazin nur wegen der Hilfsprogramme, und wieder andere die Aussicht auf neue Spiele jeden Monat zum Kiosk treibt - dann schliessen wir erst einmal daraus, daß wir mit unserem Konzept im Prinzip richtig liegen.

Andererseits bleiben natürlich viele Fragen offen. Vor allem: Gibt es vielleicht einen Bereich, der Sie interessiert, aber innerhalb des Magazins gar nicht vorkommt? Mit anderen Worten: was fehlt Ihnen in INPUT 64?

Schreiben Sie uns doch mal!

Ihre INPUT 64-Redaktion

Hinweise zur Bedienung

- 1.) Entfernen Sie - bei ausgeschaltetem Rechner - eventuell vorhandene Steckmodule. Schalten Sie vor dem Laden von INPUT 64 ihren 64er einmal kurz aus und dann wieder ein.
- 2.) Geben Sie zum Laden der Kassette (LOAD) und (RETURN) ein. Diskettenbesitzer: (LOAD"INPUT*",8,1) und (RETURN). Alles weitere geschieht von selbst. Nach der Titelgrafik springt das Programm ins Inhaltsverzeichnis des Magazins.
- 3.) Das können Sie nun mit der Leertaste durchblättern und mit (RETURN) das angezeigte Programm auswählen. Im Fenster unten rechts erhalten Sie dann weitere Hinweise ("Bitte Band zurückspulen" und so weiter ...).
- 4.) Merken Sie sich fünf Kommandos:
 - (CTRL) und gleichzeitig (h) ruft aus dem laufenden Programm eine Hilfsseite des INPUT 64-Betriebssystems auf. Sie finden darauf weitere Hinweise. Drücken Sie erneut (CTRL) und (h), verschwindet das eingeblendete Fenster, und es geht weiter im Programm.
 - (CTRL) und (i) beendet das aktuell laufende Programm und führt zurück ins Inhaltsverzeichnis. Sie können dann ein anderes Modul anwählen oder das bereits geladene Programm neu starten. (Letzteres

funktioniert nicht immer, einige Programme lassen dies nicht zu. Sie werden dann zum "Band zurückspulen" aufgefordert. Von Diskette wird automatisch neu geladen.)

- (CTRL) und (h) bzw. (CTRL) und (i) können Sie im Prinzip immer anwählen.

- (CTRL) und (s) steht für SAVen und funktioniert bei allen Programmen, die auch außerhalb von INPUT 64 anwendbar sind: Spiele, Tools, Anwenderprogramme etc. Legen Sie Ihre eigene Kassette oder Diskette ein, drücken Sie (CTRL) und (s) und befolgen Sie die weiteren Anweisungen - Sie haben eine Kopie des Programms. Diese Option ist bei vielen Programmen nur am Programmumfang möglich - entnehmen Sie dies jeweils den Hinweisen zu den einzelnen Modulen in diesem Heft.

- (CTRL) und (q) ist für die ganz Eiligen: Mit diesen beiden Tasten können Sie das Titelbild abkürzen.

- (CTRL) und (b) ermöglicht einen Bildschirmausdruck - natürlich nicht von Grafikseiten oder Sprites! Angepaßt ist diese Hardcopy für Commodore-Drucker und kompatible Geräte. Denken Sie bitte daran, immer zuerst den Drucker und dann den Rechner einzuschalten. Voraussetzung ist Geräteadresse 4 oder 6.

6.) 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 Rückseite der Verpackung finden Sie das Inhaltsverzeichnis der Kassette/Diskette.

Auf einen Blick: INPUT 64-Betriebssystembefehle

Inhaltsverzeichnis aufrufen	CTRL und i
Hilfsseite aufrufen	CTRL und h
Programm abspeichern	CTRL und s
Bildschirmausdruck	CTRL und b
Titelbild abkürzen	CTRL und q

Auf der 2. Kassettenseite befindet sich eine Sicherheitskopie von Seite 1. Sollten Sie eventuell mit einem der Programme auf der 1. Seite Ladeschwierigkeiten haben, versuchen Sie es auf Seite 2. Führt auch dies nicht zum Erfolg, lesen Sie bitte die entsprechenden Hinweise im Kapitel "Bei Ladeproblemen"!

PS: Drücken Sie nicht RUN/STOP und RESTORE. Dadurch kann der Rechner "abstürzen". Gelangen Sie doch versehentlich in den Direktmodus, so befördert Sie ein SYS50307 zurück in das INPUT 64-Betriebssystem.

Leser fragen...

Ohne Frage

Bei Ihrem Supertape DII (Ausgabe 4/85) stört mich die Abfrage "Maschinenprogramm sichern", die nach jedem Start des Programms erscheint. Kann man das einfach ändern?

(Hartlage, Langenhagen)

Man kann. Wenn Sie nach dem Laden des Programms eingeben

POKE 2231,208 : POKE 2232,99

und das Programm dann wieder abspeichern, wird es sich nach dem Start künftig sofort mit "SUPERTAPE AKTIVIERT" melden. Diese Veränderung kann übrigens nur an der in Ausgabe 4,85 veröffentlichten Version durchgeführt werden. Durch diese Änderung wird vor der Abfrage ein Sprungbefehl (branch by not equal) eingefügt, dessen Bedingung an dieser Stelle des Programms immer zutrifft, und es geht direkt zur Initialisierung.

(d.Red.)

Spät entdeckt

Der Zeichensatzgenerator (Ausgabe 2/85) spielte bislang ganz gut. Als ich allerdings die reversen Sonderzeichen ändern wollte, waren die neuen Zeichen nach dem Abspeichern und Neuladen des Zeichensatzes verschwunden!

(kein Brief, sondern Anmerkung eines Autors)

Bestimmte Bugs werden entscheidend erst Monate später entdeckt. Folgender POKE behebt das Problem:

POKE 4583,26 (Dadurch wird auch der letzte Block des erzeugten BASIC-Files mit abgespeichert und nicht, wie bislang, 32 Zeichen zu wenig.)

Also: Zeichengenerator laden (nicht starten!), POKE durchführen, korrigierte Version abspeichern.

(d. Red.)

Anpassung von MOTORBIKE an Autos

" ... in INPUT 5/85 stellen Sie das Programm "MOTORBIKE" vor. Da dieses Programm für mich sehr interessant ist ... wäre ich Ihnen sehr dankbar, wenn Sie mit mitteilen würden, wie dieses Programm für PKW umgeschrieben werden kann ... "

(S. Biniasz, Bremen)

Die Anwendung der in Input 5/85 versprochenen Tips zur Anpassung von Motorbike an Autos setzt voraus, daß Sie über eine umfangreiche technische Dokumentation, zum Beispiel "Querschnitt durch die Automobiltechnik", zu Ihrem Wagen verfügen.

Anpassung des Motors: Motorbike geht bei der Dimensionierung der Ansaugwege und Vergaser von je einem Vergaser pro Zylinder aus und optimiert auf guten Durchzug. Da bei Automotoren oftmals nur ein Einzelvergaser für 4 Zylinder vorhanden ist, muß dieser natürlich im Durchlaß größer sein. Dazu wird der Wert $K3=.675$ in Zeile 6100 auf $K3=.82$ geändert.

Anpassung der Uebersetzung: Hier liegen die Dinge leider nicht so einfach wie beim Motor. Ersetzen Sie bitte in Zeile 10520 die Werte für "gf" und "gm" durch ihre eigenen Werte (siehe auch das Begleitheft 5/85, Seite 26). Analog zu Zeile 10520 ändern Sie Zeile 10200 (Reifen). Falls Ihr Auto ein 5-Gang-Getriebe hat, können Sie die Einteilung von Zeile 10210 übernehmen. Bei einem 4-Gang-Getriebe fällt "g5" weg. Passen Sie die Zeilen 10260,10740 und 10750 entsprechend an, und achten Sie darauf, daß kein "Division by zero error" auftritt.

Die Werte für die Gangstufen sind die Werte der Primaerübersetzung, also des Getriebes. Die Werte für Ritzel und Kettenblatt entsprechen den Werten für die Zahnräder möglicher Hinterachsen. Eine Routine zur Berechnung der Sekundaerübersetzung, also dem Produkt von Ritzel und Kettenblatt, ist in Motorbike nicht vorgesehen, Sie müßten das Programm entsprechend umschreiben.

Anpassung der Fahrleistungen: Falls Sie das Getriebe erfolgreich angepaßt haben, sollte Motorbike auch die Beschleunigung 0-100 km/h korrekt berechnen. Leider kann die Höchstgeschwindigkeit nicht wie bei Motorrädern mit einer Faustformel hinreichend genau ermittelt werden. Sie müssen hier von der folgenden Formel ausgehen:

Höchstgeschw.= Faktor * dritte Wurzel aus Nennleistung

Der Faktor besteht aus dem cw-Wert * der Stirnfläche des Autos. Falls Sie Stirnfläche oder cw-Wert nicht kennen, muß der Faktor über einen "Trick" ermittelt werden.

Die Formel wird nach dem Faktor (F) umgestellt:

$$F = \frac{\text{Höchstgeschwindigkeit.}}{3. \text{ Wurzel aus der Nennl.}}$$

Die Höchstgeschwindigkeit kann man Testberichten oder dem Kfz-Schein entnehmen. Die Nennleistung entspricht dann genau der Serie. Ersetzen Sie die "45" durch den so ermittelten Faktor in Zeile 10170, um die Höchstgeschwindigkeit des getunten Fahrzeugs zu ermitteln. Den Wert für den liegenden Fahrer (vl) bzw. die Verkleidung (vf) müssen Sie streichen. (Division by zero error vermeiden und die Bildschirmausgabe überarbeiten!).

H.-J. Kriehoff

Terminkalender — ein PASCAL-Programm —

Ihre Termine werden vom C 64 verwaltet. Sie können bis zu 250 Termine im Speicher anlegen und anschließend auf Datenträger abspeichern. Da Sie verschiedene Termin-Dateien anlegen können, ist die Kapazität nahezu unbegrenzt.

Ihre Termine können frühestens 1985 beginnen und auch über mehrer Jahre verteilt werden. Solche Dauertermine/Jahr bleiben im Speicher, bis Sie von Ihnen gelöscht werden.

Das Programm wurde in PASCAL geschrieben. Auf den folgenden Seiten sehen Sie das Listing.

Die Bedienungs-Anleitung finden Sie auf Seite 26.

Listing Termin-Kalender

```
program terminkalender;
;
(* variablen + konstanten *)
;
const max=200;
var gefunden:packed array[0..207] of boolean;

; floppy,0,5:file;
; cassin,1,0:file;
; cassout,1,1:file;
; turbain,7,0:file;
; turboout,7,1:file;
; drucker,4,7:file;
; z1,z0,device,null,menuewahl,druckstop:char;
; datenname:array[1..14] of char;
; string.bottomstring:array[1..40] of char;
; floppystring:array[1..21] of char;
; tagdatum,suchname,datum:array[1..10] of char;
; tag,monat,jahr,code,t,e,j,c,st,sm,ej,ec,lauf,il,i2:integer;
; eingabeart,tagnummer:integer;

; tag:array[1..12] of integer;
; tagname:array[1..7,1..10] of char;
; termin:array[1..max] of record
;   a:integer;
;   b:integer;
;   c:array[1..37] of char;
;   end; (* of record *)
;
(* funktionen + prozeduren *)
;
function eof (dummy:integer):boolean;
begin
  eof := (peek(14) and 64)=64;
end; (* of function *)
;
procedure formatausgabe;
begin
  string := '';
  string[0..0] := termin.c[lauf,
1..length(termin.c[lauf])];
```



```

1 function schluess (art:integer):char;
2 begin
3   if art=1 then begin
4     pose z1y,z3, writes;
5     write ('# schluess y (j/m) #');
6     until (schluess='j') or (schluess='p');
7     writeln ('#');
8     end; (* of if *)
9   else schluess := 'j';
10  end; (* of function *)
11
12 procedure eingabe;
13 begin
14   write ('#');
15   write ('# Eingabe des... termin ');
16   write ('#');
17   write ('#');
18   writeln ('#');
19   termin:=clauf; (* string(1..37) );
20 end; (* of procedure *)
21
22 procedure weiter;
23 begin
24   schaltjahr (sj);
25   if schaltjahr then st:=st+1;
26   else begin
27     if sj=st then begin
28       st:=st+1;
29     end; (* of if *)
30   else begin
31     sj:=sj+1; st:=1;
32     end; (* of else *)
33   end; (* of procedure *)
34 end; (* of procedure *)
35
36 procedure wochentag (ta,no,ja:integer; var index:integer);
37 var k,l,o,z1,z2,z3,z4,z:integer;
38 p:real;
39 integer;
40 i:=trunc((61+no)/7);
41 o:=no-12*k;
42 p:=1/180;
43 z1:=p div 4;
44 z2:=trunc(p);
45 z3:=trunc((3+z1+5)/5);
46 index:=z1+z2-z3+ta-1;
47 index:=index mod 7+1;
48 end; (* of procedure *)
49
50 procedure datumsingabe (var z1,z2,z3:integer);
51 begin integer;
52 string(1..10);
53 write ('#');
54 write ('# Datum eingeben ');
55 write ('#');
56 write ('#');
57 write ('#');
58 end;
59
60 function schluess (art:integer):char;
61 begin
62   if art=1 then begin
63     pose z1y,z3, writes;
64     write ('# schluess y (j/m) #');
65     until (schluess='j') or (schluess='p');
66     writeln ('#');
67     end; (* of if *)
68   else schluess := 'j';
69   end; (* of function *)
70
71 procedure eingabe;
72 begin
73   write ('#');
74   write ('# Eingabe des... termin ');
75   write ('#');
76   write ('#');
77   writeln ('#');
78   termin:=clauf; (* string(1..37) );
79 end; (* of procedure *)
80
81 procedure weiter;
82 begin
83   schaltjahr (sj);
84   if schaltjahr then st:=st+1;
85   else begin
86     if sj=st then begin
87       st:=st+1;
88     end; (* of if *)
89   else begin
90     sj:=sj+1; st:=1;
91     end; (* of else *)
92   end; (* of procedure *)
93 end; (* of procedure *)
94
95 procedure wochentag (ta,no,ja:integer; var index:integer);
96 var k,l,o,z1,z2,z3,z4,z:integer;
97 p:real;
98 integer;
99 i:=trunc((61+no)/7);
100 o:=no-12*k;
101 p:=1/180;
102 z1:=p div 4;
103 z2:=trunc(p);
104 z3:=trunc((3+z1+5)/5);
105 index:=z1+z2-z3+ta-1;
106 index:=index mod 7+1;
107 end; (* of procedure *)
108
109 procedure datumsingabe (var z1,z2,z3:integer);
110 begin integer;
111 string(1..10);
112 write ('#');
113 write ('# Datum eingeben ');
114 write ('#');
115 write ('#');
116 write ('#');
117 end;
118
119 function schluess (art:integer):char;
120 begin
121   if art=1 then begin
122     pose z1y,z3, writes;
123     write ('# schluess y (j/m) #');
124     until (schluess='j') or (schluess='p');
125     writeln ('#');
126     end; (* of if *)
127   else schluess := 'j';
128   end; (* of function *)
129
130 procedure eingabe;
131 begin
132   write ('#');
133   write ('# Eingabe des... termin ');
134   write ('#');
135   write ('#');
136   writeln ('#');
137   termin:=clauf; (* string(1..37) );
138 end; (* of procedure *)
139
140 procedure weiter;
141 begin
142   schaltjahr (sj);
143   if schaltjahr then st:=st+1;
144   else begin
145     if sj=st then begin
146       st:=st+1;
147     end; (* of if *)
148   else begin
149     sj:=sj+1; st:=1;
150     end; (* of else *)
151   end; (* of procedure *)
152 end; (* of procedure *)
153
154 procedure wochentag (ta,no,ja:integer; var index:integer);
155 var k,l,o,z1,z2,z3,z4,z:integer;
156 p:real;
157 integer;
158 i:=trunc((61+no)/7);
159 o:=no-12*k;
160 p:=1/180;
161 z1:=p div 4;
162 z2:=trunc(p);
163 z3:=trunc((3+z1+5)/5);
164 index:=z1+z2-z3+ta-1;
165 index:=index mod 7+1;
166 end; (* of procedure *)
167
168 procedure datumsingabe (var z1,z2,z3:integer);
169 begin integer;
170 string(1..10);
171 write ('#');
172 write ('# Datum eingeben ');
173 write ('#');
174 write ('#');
175 write ('#');
176 end;
177
178 function schluess (art:integer):char;
179 begin
180   if art=1 then begin
181     pose z1y,z3, writes;
182     write ('# schluess y (j/m) #');
183     until (schluess='j') or (schluess='p');
184     writeln ('#');
185     end; (* of if *)
186   else schluess := 'j';
187   end; (* of function *)
188
189 procedure eingabe;
190 begin
191   write ('#');
192   write ('# Eingabe des... termin ');
193   write ('#');
194   write ('#');
195   writeln ('#');
196   termin:=clauf; (* string(1..37) );
197 end; (* of procedure *)
198
199 procedure weiter;
200 begin
201   schaltjahr (sj);
202   if schaltjahr then st:=st+1;
203   else begin
204     if sj=st then begin
205       st:=st+1;
206     end; (* of if *)
207   else begin
208     sj:=sj+1; st:=1;
209     end; (* of else *)
210   end; (* of procedure *)
211 end; (* of procedure *)
212
213 procedure wochentag (ta,no,ja:integer; var index:integer);
214 var k,l,o,z1,z2,z3,z4,z:integer;
215 p:real;
216 integer;
217 i:=trunc((61+no)/7);
218 o:=no-12*k;
219 p:=1/180;
220 z1:=p div 4;
221 z2:=trunc(p);
222 z3:=trunc((3+z1+5)/5);
223 index:=z1+z2-z3+ta-1;
224 index:=index mod 7+1;
225 end; (* of procedure *)
226
227 procedure datumsingabe (var z1,z2,z3:integer);
228 begin integer;
229 string(1..10);
230 write ('#');
231 write ('# Datum eingeben ');
232 write ('#');
233 write ('#');
234 write ('#');
235 end;
236
237 function schluess (art:integer):char;
238 begin
239   if art=1 then begin
240     pose z1y,z3, writes;
241     write ('# schluess y (j/m) #');
242     until (schluess='j') or (schluess='p');
243     writeln ('#');
244     end; (* of if *)
245   else schluess := 'j';
246   end; (* of function *)
247
248 procedure eingabe;
249 begin
250   write ('#');
251   write ('# Eingabe des... termin ');
252   write ('#');
253   write ('#');
254   writeln ('#');
255   termin:=clauf; (* string(1..37) );
256 end; (* of procedure *)
257
258 procedure weiter;
259 begin
260   schaltjahr (sj);
261   if schaltjahr then st:=st+1;
262   else begin
263     if sj=st then begin
264       st:=st+1;
265     end; (* of if *)
266   else begin
267     sj:=sj+1; st:=1;
268     end; (* of else *)
269   end; (* of procedure *)
270 end; (* of procedure *)
271
272 procedure wochentag (ta,no,ja:integer; var index:integer);
273 var k,l,o,z1,z2,z3,z4,z:integer;
274 p:real;
275 integer;
276 i:=trunc((61+no)/7);
277 o:=no-12*k;
278 p:=1/180;
279 z1:=p div 4;
280 z2:=trunc(p);
281 z3:=trunc((3+z1+5)/5);
282 index:=z1+z2-z3+ta-1;
283 index:=index mod 7+1;
284 end; (* of procedure *)
285
286 procedure datumsingabe (var z1,z2,z3:integer);
287 begin integer;
288 string(1..10);
289 write ('#');
290 write ('# Datum eingeben ');
291 write ('#');
292 write ('#');
293 write ('#');
294 end;
295
296 function schluess (art:integer):char;
297 begin
298   if art=1 then begin
299     pose z1y,z3, writes;
300     write ('# schluess y (j/m) #');
301     until (schluess='j') or (schluess='p');
302     writeln ('#');
303     end; (* of if *)
304   else schluess := 'j';
305   end; (* of function *)
306
307 procedure eingabe;
308 begin
309   write ('#');
310   write ('# Eingabe des... termin ');
311   write ('#');
312   write ('#');
313   writeln ('#');
314   termin:=clauf; (* string(1..37) );
315 end; (* of procedure *)
316
317 procedure weiter;
318 begin
319   schaltjahr (sj);
320   if schaltjahr then st:=st+1;
321   else begin
322     if sj=st then begin
323       st:=st+1;
324     end; (* of if *)
325   else begin
326     sj:=sj+1; st:=1;
327     end; (* of else *)
328   end; (* of procedure *)
329 end; (* of procedure *)
330
331 procedure wochentag (ta,no,ja:integer; var index:integer);
332 var k,l,o,z1,z2,z3,z4,z:integer;
333 p:real;
334 integer;
335 i:=trunc((61+no)/7);
336 o:=no-12*k;
337 p:=1/180;
338 z1:=p div 4;
339 z2:=trunc(p);
340 z3:=trunc((3+z1+5)/5);
341 index:=z1+z2-z3+ta-1;
342 index:=index mod 7+1;
343 end; (* of procedure *)
344
345 procedure datumsingabe (var z1,z2,z3:integer);
346 begin integer;
347 string(1..10);
348 write ('#');
349 write ('# Datum eingeben ');
350 write ('#');
351 write ('#');
352 write ('#');
353 end;
354
355 function schluess (art:integer):char;
356 begin
357   if art=1 then begin
358     pose z1y,z3, writes;
359     write ('# schluess y (j/m) #');
360     until (schluess='j') or (schluess='p');
361     writeln ('#');
362     end; (* of if *)
363   else schluess := 'j';
364   end; (* of function *)
365
366 procedure eingabe;
367 begin
368   write ('#');
369   write ('# Eingabe des... termin ');
370   write ('#');
371   write ('#');
372   writeln ('#');
373   termin:=clauf; (* string(1..37) );
374 end; (* of procedure *)
375
376 procedure weiter;
377 begin
378   schaltjahr (sj);
379   if schaltjahr then st:=st+1;
380   else begin
381     if sj=st then begin
382       st:=st+1;
383     end; (* of if *)
384   else begin
385     sj:=sj+1; st:=1;
386     end; (* of else *)
387   end; (* of procedure *)
388 end; (* of procedure *)
389
390 procedure wochentag (ta,no,ja:integer; var index:integer);
391 var k,l,o,z1,z2,z3,z4,z:integer;
392 p:real;
393 integer;
394 i:=trunc((61+no)/7);
395 o:=no-12*k;
396 p:=1/180;
397 z1:=p div 4;
398 z2:=trunc(p);
399 z3:=trunc((3+z1+5)/5);
400 index:=z1+z2-z3+ta-1;
401 index:=index mod 7+1;
402 end; (* of procedure *)
403
404 procedure datumsingabe (var z1,z2,z3:integer);
405 begin integer;
406 string(1..10);
407 write ('#');
408 write ('# Datum eingeben ');
409 write ('#');
410 write ('#');
411 write ('#');
412 end;
413
414 function schluess (art:integer):char;
415 begin
416   if art=1 then begin
417     pose z1y,z3, writes;
418     write ('# schluess y (j/m) #');
419     until (schluess='j') or (schluess='p');
420     writeln ('#');
421     end; (* of if *)
422   else schluess := 'j';
423   end; (* of function *)
424
425 procedure eingabe;
426 begin
427   write ('#');
428   write ('# Eingabe des... termin ');
429   write ('#');
430   write ('#');
431   writeln ('#');
432   termin:=clauf; (* string(1..37) );
433 end; (* of procedure *)
434
435 procedure weiter;
436 begin
437   schaltjahr (sj);
438   if schaltjahr then st:=st+1;
439   else begin
440     if sj=st then begin
441       st:=st+1;
442     end; (* of if *)
443   else begin
444     sj:=sj+1; st:=1;
445     end; (* of else *)
446   end; (* of procedure *)
447 end; (* of procedure *)
448
449 procedure wochentag (ta,no,ja:integer; var index:integer);
450 var k,l,o,z1,z2,z3,z4,z:integer;
451 p:real;
452 integer;
453 i:=trunc((61+no)/7);
454 o:=no-12*k;
455 p:=1/180;
456 z1:=p div 4;
457 z2:=trunc(p);
458 z3:=trunc((3+z1+5)/5);
459 index:=z1+z2-z3+ta-1;
460 index:=index mod 7+1;
461 end; (* of procedure *)
462
463 procedure datumsingabe (var z1,z2,z3:integer);
464 begin integer;
465 string(1..10);
466 write ('#');
467 write ('# Datum eingeben ');
468 write ('#');
469 write ('#');
470 write ('#');
471 end;
472
473 function schluess (art:integer):char;
474 begin
475   if art=1 then begin
476     pose z1y,z3, writes;
477     write ('# schluess y (j/m) #');
478     until (schluess='j') or (schluess='p');
479     writeln ('#');
480     end; (* of if *)
481   else schluess := 'j';
482   end; (* of function *)
483
484 procedure eingabe;
485 begin
486   write ('#');
487   write ('# Eingabe des... termin ');
488   write ('#');
489   write ('#');
490   writeln ('#');
491   termin:=clauf; (* string(1..37) );
492 end; (* of procedure *)
493
494 procedure weiter;
495 begin
496   schaltjahr (sj);
497   if schaltjahr then st:=st+1;
498   else begin
499     if sj=st then begin
500       st:=st+1;
501     end; (* of if *)
502   else begin
503     sj:=sj+1; st:=1;
504     end; (* of else *)
505   end; (* of procedure *)
506 end; (* of procedure *)
507
508 procedure wochentag (ta,no,ja:integer; var index:integer);
509 var k,l,o,z1,z2,z3,z4,z:integer;
510 p:real;
511 integer;
512 i:=trunc((61+no)/7);
513 o:=no-12*k;
514 p:=1/180;
515 z1:=p div 4;
516 z2:=trunc(p);
517 z3:=trunc((3+z1+5)/5);
518 index:=z1+z2-z3+ta-1;
519 index:=index mod 7+1;
520 end; (* of procedure *)
521
522 procedure datumsingabe (var z1,z2,z3:integer);
523 begin integer;
524 string(1..10);
525 write ('#');
526 write ('# Datum eingeben ');
527 write ('#');
528 write ('#');
529 write ('#');
530 end;
531
532 function schluess (art:integer):char;
533 begin
534   if art=1 then begin
535     pose z1y,z3, writes;
536     write ('# schluess y (j/m) #');
537     until (schluess='j') or (schluess='p');
538     writeln ('#');
539     end; (* of if *)
540   else schluess := 'j';
541   end; (* of function *)
542
543 procedure eingabe;
544 begin
545   write ('#');
546   write ('# Eingabe des... termin ');
547   write ('#');
548   write ('#');
549   writeln ('#');
550   termin:=clauf; (* string(1..37) );
551 end; (* of procedure *)
552
553 procedure weiter;
554 begin
555   schaltjahr (sj);
556   if schaltjahr then st:=st+1;
557   else begin
558     if sj=st then begin
559       st:=st+1;
560     end; (* of if *)
561   else begin
562     sj:=sj+1; st:=1;
563     end; (* of else *)
564   end; (* of procedure *)
565 end; (* of procedure *)
566
567 procedure wochentag (ta,no,ja:integer; var index:integer);
568 var k,l,o,z1,z2,z3,z4,z:integer;
569 p:real;
570 integer;
571 i:=trunc((61+no)/7);
572 o:=no-12*k;
573 p:=1/180;
574 z1:=p div 4;
575 z2:=trunc(p);
576 z3:=trunc((3+z1+5)/5);
577 index:=z1+z2-z3+ta-1;
578 index:=index mod 7+1;
579 end; (* of procedure *)
580
581 procedure datumsingabe (var z1,z2,z3:integer);
582 begin integer;
583 string(1..10);
584 write ('#');
585 write ('# Datum eingeben ');
586 write ('#');
587 write ('#');
588 write ('#');
589 end;
590
591 function schluess (art:integer):char;
592 begin
593   if art=1 then begin
594     pose z1y,z3, writes;
595     write ('# schluess y (j/m) #');
596     until (schluess='j') or (schluess='p');
597     writeln ('#');
598     end; (* of if *)
599   else schluess := 'j';
600   end; (* of function *)
601
602 procedure eingabe;
603 begin
604   write ('#');
605   write ('# Eingabe des... termin ');
606   write ('#');
607   write ('#');
608   writeln ('#');
609   termin:=clauf; (* string(1..37) );
610 end; (* of procedure *)
611
612 procedure weiter;
613 begin
614   schaltjahr (sj);
615   if schaltjahr then st:=st+1;
616   else begin
617     if sj=st then begin
618       st:=st+1;
619     end; (* of if *)
620   else begin
621     sj:=sj+1; st:=1;
622     end; (* of else *)
623   end; (* of procedure *)
624 end; (* of procedure *)
625
626 procedure wochentag (ta,no,ja:integer; var index:integer);
627 var k,l,o,z1,z2,z3,z4,z:integer;
628 p:real;
629 integer;
630 i:=trunc((61+no)/7);
631 o:=no-12*k;
632 p:=1/180;
633 z1:=p div 4;
634 z2:=trunc(p);
635 z3:=trunc((3+z1+5)/5);
636 index:=z1+z2-z3+ta-1;
637 index:=index mod 7+1;
638 end; (* of procedure *)
639
640 procedure datumsingabe (var z1,z2,z3:integer);
641 begin integer;
642 string(1..10);
643 write ('#');
644 write ('# Datum eingeben ');
645 write ('#');
646 write ('#');
647 write ('#');
648 end;
649
650 function schluess (art:integer):char;
651 begin
652   if art=1 then begin
653     pose z1y,z3, writes;
654     write ('# schluess y (j/m) #');
655     until (schluess='j') or (schluess='p');
656     writeln ('#');
657     end; (* of if *)
658   else schluess := 'j';
659   end; (* of function *)
660
661 procedure eingabe;
662 begin
663   write ('#');
664   write ('# Eingabe des... termin ');
665   write ('#');
666   write ('#');
667   writeln ('#');
668   termin:=clauf; (* string(1..37) );
669 end; (* of procedure *)
670
671 procedure weiter;
672 begin
673   schaltjahr (sj);
674   if schaltjahr then st:=st+1;
675   else begin
676     if sj=st then begin
677       st:=st+1;
678     end; (* of if *)
679   else begin
680     sj:=sj+1; st:=1;
681     end; (* of else *)
682   end; (* of procedure *)
683 end; (* of procedure *)
684
685 procedure wochentag (ta,no,ja:integer; var index:integer);
686 var k,l,o,z1,z2,z3,z4,z:integer;
687 p:real;
688 integer;
689 i:=trunc((61+no)/7);
690 o:=no-12*k;
691 p:=1/180;
692 z1:=p div 4;
693 z2:=trunc(p);
694 z3:=trunc((3+z1+5)/5);
695 index:=z1+z2-z3+ta-1;
696 index:=index mod 7+1;
697 end; (* of procedure *)
698
699 procedure datumsingabe (var z1,z2,z3:integer);
700 begin integer;
701 string(1..10);
702 write ('#');
703 write ('# Datum eingeben ');
704 write ('#');
705 write ('#');
706 write ('#');
707 end;
708
709 function schluess (art:integer):char;
710 begin
711   if art=1 then begin
712     pose z1y,z3, writes;
713     write ('# schluess y (j/m) #');
714     until (schluess='j') or (schluess='p');
715     writeln ('#');
716     end; (* of if *)
717   else schluess := 'j';
718   end; (* of function *)
719
720 procedure eingabe;
721 begin
722   write ('#');
723   write ('# Eingabe des... termin ');
724   write ('#');
725   write ('#');
726   writeln ('#');
727   termin:=clauf; (* string(1..37) );
728 end; (* of procedure *)
729
730 procedure weiter;
731 begin
732   schaltjahr (sj);
733   if schaltjahr then st:=st+1;
734   else begin
735     if sj=st then begin
736       st:=st+1;
737     end; (* of if *)
738   else begin
739     sj:=sj+1; st:=1;
740     end; (* of else *)
741   end; (* of procedure *)
742 end; (* of procedure *)
743
744 procedure wochentag (ta,no,ja:integer; var index:integer);
745 var k,l,o,z1,z2,z3,z4,z:integer;
746 p:real;
747 integer;
748 i:=trunc((61+no)/7);
749 o:=no-12*k;
750 p:=1/180;
751 z1:=p div 4;
752 z2:=trunc(p);
753 z3:=trunc((3+z1+5)/5);
754 index:=z1+z2-z3+ta-1;
755 index:=index mod 7+1;
756 end; (* of procedure *)
757
758 procedure datumsingabe (var z1,z2,z3:integer);
759 begin integer;
760 string(1..10);
761 write ('#');
762 write ('# Datum eingeben ');
763 write ('#');
764 write ('#');
765 write ('#');
766 end;
767
768 function schluess (art:integer):char;
769 begin
770   if art=1 then begin
771     pose z1y,z3, writes;
772     write ('# schluess y (j/m) #');
773     until (schluess='j') or (schluess='p');
774     writeln ('#');
775     end; (* of if *)
776   else schluess := 'j';
777   end; (* of function *)
778
779 procedure eingabe;
780 begin
781   write ('#');
782   write ('# Eingabe des... termin ');
783   write ('#');
784   write ('#');
785   writeln ('#');
786   termin:=clauf; (* string(1..37) );
787 end; (* of procedure *)
788
789 procedure weiter;
790 begin
791   schaltjahr (sj);
792   if schaltjahr then st:=st+1;
793   else begin
794     if sj=st then begin
795       st:=st+1;
796     end; (* of if *)
797   else begin
798     sj:=sj+1; st:=1;
799     end; (* of else *)
800   end; (* of procedure *)
801 end; (* of procedure *)
802
803 procedure wochentag (ta,no,ja:integer; var index:integer);
804 var k,l,o,z1,z2,z3,z4,z:integer;
805 p:real;
806 integer;
807 i:=trunc((61+no)/7);
808 o:=no-12*k;
809 p:=1/180;
810 z1:=p div 4;
811 z2:=trunc(p);
812 z3:=trunc((3+z1+5)/5);
813 index:=z1+z2-z3+ta-1;
814 index:=index mod 7+1;
815 end; (* of procedure *)
816
817 procedure datumsingabe (var z1,z2,z3:integer);
818 begin integer;
819 string(1..10);
820 write ('#');
821 write ('# Datum eingeben ');
822 write ('#');
823 write ('#');
824 write ('#');
825 end;
826
827 function schluess (art:integer):char;
828 begin
829   if art=1 then begin
830     pose z1y,z3, writes;
831     write ('# schluess y (j/m) #');
832     until (schluess='j') or (schluess='p');
833     writeln ('#');
834     end; (* of if *)
835   else schluess := 'j';
836   end; (* of function *)
837
838 procedure eingabe;
839 begin
840   write ('#');
841   write ('# Eingabe des... termin ');
842   write ('#');
843   write ('#');
844   writeln ('#');
845   termin:=clauf; (* string(1..37) );
846 end; (* of procedure *)
847
848 procedure weiter;
849 begin
850   schaltjahr (sj);
851   if schaltjahr then st:=st+1;
852   else begin
853     if sj=st then begin
854       st:=st+1;
855     end; (* of if *)
856   else begin
857     sj:=sj+1; st:=1;
858     end; (* of else *)
859   end; (* of procedure *)
860 end; (* of procedure *)
861
862 procedure wochentag (ta,no,ja:integer; var index:integer);
863 var k,l,o,z1,z2,z3,z4,z:integer;
864 p:real;
865 integer;
866 i:=trunc((61+no)/7);
867 o:=no-12*k;
868 p:=1/180;
869 z1:=p div 4;
870 z2:=trunc(p);
871 z3:=trunc((3+z1+5)/5);
872 index:=z1+z2-z3+ta-1;
873 index:=index mod 7+1;
874 end; (* of procedure *)
875
876 procedure datumsingabe (var z1,z2,z3:integer);
877 begin integer;
878 string(1..10);
879 write ('#');
880 write ('# Datum eingeben ');
881 write ('#');
882 write ('#');
883 write ('#');
884 end;
885
886 function schluess (art:integer):char;
887 begin
888   if art=1 then begin
889     pose z1y,z3, writes;
890     write ('# schluess y (j/m) #');
891     until (schluess='j') or (schluess='p');
892     writeln ('#');
893     end; (* of if *)
894   else schluess := 'j';
895   end; (* of function *)
896
897 procedure eingabe;
898 begin
899   write ('#');
900   write ('# Eingabe des... termin ');
901   write ('#');
902   write ('#');
903   writeln ('#');
904   termin:=clauf; (* string(1..37) );
905 end; (* of procedure *)
906
907 procedure weiter;
908 begin
909   schaltjahr (sj);
910   if schaltjahr then st:=st+1;
911   else begin
912     if sj=st then begin
913       st:=st+1;
914     end; (* of if *)
915   else begin
916     sj:=sj+1; st:=1;
917     end; (* of else *)
918   end; (* of procedure *)
919 end; (* of procedure *)
920
921 procedure wochentag (ta,no,ja:integer; var index:integer);
922 var k,l,o,z1,z2,z3,z4,z:integer;
923 p:real;
924 integer;
925 i:=trunc((61+no)/7);
926 o:=no-12*k;
927 p:=1/180;
928 z1:=p div 4;
929 z2:=trunc(p);
930 z3:=trunc((3+z1+5)/5);
931 index:=z1+z2-z3+ta-1;
932 index:=index mod 7+1;
933 end; (* of procedure *)
934
935 procedure datumsingabe (var z1,z2,z3:integer);
936 begin integer;
937 string(1..10);
938 write ('#');
939 write ('# Datum eingeben ');
940 write ('#');
941 write ('#');
942 write ('#');
943 end;
944
945 function schluess (art:integer):char;
946 begin
947   if art=1 then begin
948     pose z1y,z3, writes;
949     write ('# schluess y (j/m) #');
950     until (schluess='j') or (schluess='p');
951     writeln ('#');
952     end; (* of if *)
953   else schluess := 'j';
954   end; (* of function *)
955
956 procedure eingabe;
957 begin
958   write ('#');
959   write ('# Eingabe des... termin ');
960   write ('#');
961   write ('#');
962   writeln ('#');
963   termin:=clauf; (* string(1..37) );
964 end; (* of procedure *)
965
966 procedure weiter;
967 begin
968   schaltjahr (sj);
969   if schaltjahr then st:=st+1;
970   else begin
971     if sj=st then begin
972       st:=st+1;
973     end; (* of if *)
974   else begin
975     sj:=sj+1; st:=1;
976     end; (* of else *)
977   end; (* of procedure *)
978 end; (* of procedure *)
979
980 procedure wochentag (ta,no,ja:integer; var index:integer);
981 var k,l,o,z1,z2,z3,z4,z:integer;
982 p:real;
983 integer;
984 i:=trunc((61+no)/7);
985 o:=no-12*k;
986 p:=1/180;
987 z1:=p div 4;
988 z2:=trunc(p);
989 z3:=trunc((3+z1+5)/5);
990 index:=z1+z2-z3+ta-1;
991 index:=index mod 7+1;
992 end; (* of procedure *)
993
994 procedure datumsingabe (var z1,z2,z3:integer);
995 begin integer;
996 string(1..10);
997 write ('#');
998 write ('# Datum eingeben ');
999 write ('#');
1000 write ('#');
1001 write ('#');
1002 end;

```



```

weiter;
get druckstop;
until druckstop>chr(0);
close casin;
end; (* of case 5 *)

"5"; begin
if sicher(5)="j" then begin
for lauf:= 1 to max do
case device of
-1; begin lauf:= 1;
read (casin, dateiname);
while not eof(13) do begin
readln (casin, termin, clauf3);
readln (casin, termin, bclauf3);
ll:= 0; repeat ll:= ll+1;
read (casin, termin, clauf, il2);
until ((termin, clauf, il3)=chr(13)) or (ll=37);
else (termin, clauf, il3)=chr(13) then termin:=cllauf, il3:= chr(0);
lauf:= lauf+1;
end; (* while *)
close (casin);
end; (* 1 *)
-7; begin lauf:= 1;
read (turboin, dateiname);
read (turboin, z1);
read (turboin, z2);
termin:=cllauf3:= ord(z1)+256*ord(z2);
read (turboin, z1);
read (turboin, z2);
termin:=cllauf3:= ord(z1)+256*ord(z2);
lauf:= lauf+1;
end; (* while *)
end; (* 7 *)
-8; begin
string(1..8):= dateiname;
floppystring(15..18):="a, e";
floppystring(19):= chr(0); e="";
writein ("loading", floppystring(1..14));
reset (floppy, floppystring);
lauf:= 1; eof(13) do begin
read (floppy, termin, clauf3);
lauf:= lauf+1;
end; (* of while *)
end; (* of if *)
close (floppy);
if ioerror(1) then begin end;
end; (* of case *)
end; (* of case 5 *)

"5"; begin
write ("dateiname: -> ");
stringout (13);
dateiname:= string(1..14);
if ll13 then dateiname;
ll:= ll+1; begin
for il:= 1 to 14 do dateiname(il):= chr(160);
end; (* of if *)

```

```

end; (* of case 5 *)

"7"; begin
if sicher(7)="j" then begin
for lauf:= 1 to max do
begin
if ((termin, bclauf3) and (termin, clauf3)) then
initial (lauf);
else begin
if ((termin, clauf3<c) and (termin, bclauf3)) then
initial (lauf);
end; (* of for *)
end; (* of for *)
case device of
-1; begin rewrite (casout, dateiname);
if lauf:= 1 to max do
writein (casout, termin, clauf3);
writein (casout, termin, bclauf3);
end; (* of if *)
close (casout);
-7; begin rewrite (turboout, dateiname);
end; (* 1 *)
if lauf:= 1 to max do begin
write (clauf3);
write (turboout, chr(termin, clauf) mod 256);
write (turboout, chr(termin, bclauf) mod 256);
write (turboout, chr(termin, bclauf2 div 256));
for il:= 1 to 37 do
write (turboout, termin, clauf, il3);
end; (* of if *)
close (turboout);
end; (* 7 *)
-8; begin
floppystring(1..8):="a, e";
floppystring(3..8):= dateiname;
floppystring(17..8):="a, e";
floppystring(19):= chr(0);
writein ("loading", floppystring(1..16));
rewrite (floppy, floppystring);
if not(ioerror(1)) then begin
for lauf:= 1 to max do
if termin:=clauf3>8 then
write (floppy, termin, clauf3);
close (floppy);
if ioerror(1) then begin end;
end; (* of while *)
end; (* of case *)
end; (* of if *)
end; (* of case 7 *)
-8";
if (sicher(8)="j") then menuwahl:= "q";
end;
end; (* of maincase *)
close (error);
until menuwahl="q";
readln;
goto start;
end;

```

SUPERKÄSI

Dieses Spiel hat nichts mit irgendwelchen Käsesorten zu tun, auch wenn der Name so klingen mag. Vielmehr ist SUPERKÄSI ein Spiel, das Sie sicherlich noch aus Ihrer Schulzeit kennen (Käsekästchen). Ihr Gegner ist hierbei aber der Computer.

Als Erstes sehen Sie das Hauptmenü. Hier können Sie wählen, wieviel "Striche" der Computer vorgeben, mit welcher Spielstärke gespielt werden und wer anfangen soll.

Nachdem Sie die Optionen im Hauptmenü gesetzt und das Spiel gestartet haben, sehen Sie das Spielfeld. Es ist einem karierten Blatt Papier nachempfunden. Fängt der Computer mit dem Spiel an, setzt er eine schwarze Linie. Wird auf dem Spielfeld eine Hand abgebildet, setzen Sie eine Linie.

Beispiel:

Angenommen, der Computer setzt eine waagerechte Linie, Sie setzen an diese waagerechte Linie eine senkrechte Linie und der Computer setzt seine Linie so, daß von einem Quadrat nur noch eine Seite offen bleibt.

Als nächstes könnten Sie mit einer weiteren Linie dieses Quadrat vervollkommen und Ihr Zeichen in das Quadrat eintragen. Sie hätten damit Ihren ersten Punkt zu verzeichnen.

Spielsteuerung:

Das Spiel kann mit dem Joystick in Port 2 oder der Tastatur gespielt werden. Um eine ausgewählte Linie zu markieren, müssen Sie zunächst die abgebildete Hand in ein daran angrenzendes Kästchen manövrieren. Dann drücken Sie den Feuerknopf, und die Hand wechselt Ihre Farbe. Jetzt können Sie eine der vier Linien, die das Kästchen Kästchen umgrenzen, durch erneutes Verschieben der Hand auswählen, und durch einen weiteren Tastendruck auf den Feuerknopf kennzeichnen.

Tastaturbelegung:

Unten	= M
Oben	= K
Rechts	= C
Links	= Z
Feuerknopf	= E

Mathe mit Nico

Diesmal treffen Sie auf zwei Unbekannte. Sie müssen mit Ihnen rechnen, um sie zu entlarven.

Für die Lösung von Gleichungen mit zwei Unbekannten in dieser Folge wurde die Rechenseite erweitert. Wie in den früheren Folgen, steht Ihnen innerhalb des Kurses eine Bildschirmseite zur Verfügung. Auf dieser Seite können Sie Rechnungen durchführen, wobei der C 64 Ihnen als "Taschen-Rechner" zur Seite steht.

Ab dieser Ausgabe dürfen Sie jetzt alle Rechen-Kommandos verwenden, wie Sie sie aus dem Direkt-Modus des C 64 kennen. Sogar die Verwendung der Standard-Funktionen wie SIN, EXP, INT, und so weiter sind möglich.

Sie geben beispielsweise ein:

```
8*INT(4096/100) RETURN-Taste
```

und der Rechner gibt aus:

```
= 320
```

Und nun viel Spaß bei "Mathe mit Nico"!

REASSEMBLER V7.6

Was ist ein REASSEMBLER?

Das Problem ist aus dem Umgang mit Monitorprogrammen bekannt: Sie haben ein Maschinen-Programm im Speicher oder auf Diskette und möchten sich gerne den entsprechenden Assembler-Code ansehen, um zu erfahren, mit welchen Tricks und welcher Weisheit das betreffende Programm erstellt wurde, und an welcher Stelle es am sinnvollsten geändert werden kann. Doch liefern Monitore in der Regel nur eine direkte Umwandlung sinnvoller Codes in Assembler-Befehle, sogenannte MNEMONICS.

Die Übersicht wird zum einen erheblich dadurch erschwert, daß keinerlei Label aufgeführt werden und zum anderen dadurch, daß ein Monitor keine Tabellen erkennen kann, die zum Beispiel Texte enthalten, also keinen sinnvollen OP-Code.

Hierfür gibt es nun sogenannte Reassembler. Diese Programme erzeugen aus einem Maschinen-Code einen lesbaren Assembler-Code, zumindest sollten sie das. In den meisten Fällen können die erzeugten Assembler-Listings auch auf einen Drucker ausgegeben werden, doch fehlt oft die Möglichkeit, diese Quellen-Codes auf den eigenen Datenträger

abzuspeichern und anschließend von einem Assembler nach eventuellen Änderungen assemblieren zu lassen.

Ein weiteres Problem ergibt sich, wenn das Maschinen-Programm ausgerechnet den Speicherbereich beansprucht, in dem der Reassembler selbst liegt. In diesem Fall steht der neugierige Anwender im Dunkeln. Ein Reassembler sollte auch in der Lage sein, direkt von Diskette herunter zu reassemblieren und den Source-Code (Quellen-Liste) wieder auf Diskette auszugeben. Nur so können längere Sources verarbeitet werden, ohne den Speicherbereich zu überziehen. Hierbei muß das Programm soweit abgesichert sein, daß fehlerhafte Operationen der Disketten- oder Kassetten-Station nicht zum Absturz führen.

Funktionsweise des ReAss

ReAss bearbeitet ein Objekt-File (Maschinen-Programm) in zwei Durchgängen, diese werden PASSES genannt:

PASS 1:

Hier werden alle Adressen (Operanden), die auf das Programm selbst zeigen, als Label dargestellt.

Beginnt ein Maschinen-Programm bei \$C000 und endet bei \$C500, so wird aus:

```
JSR $C123      ein   JSR LBXXX
```

wobei 'LB' die Standard-Bezeichnung für Label und 'XXX' eine fortlaufende Nummerierung darstellen. Adressen außerhalb des Programms werden nicht in Label umgewandelt. Wurde eine Symboltabelle zugeladen, und ist dieses Label in der Tabelle enthalten, wird 'LBXXX' gegen die aufgeführte Bezeichnung ausgetauscht.

Beispiel:

```
'LIESRT = $c123' wird zu JSR LIESRT
```

Sind alle Label in PASS 1 vom ReAss erkannt worden, werden sie in aufsteigender Reihenfolge sortiert, was das Auffinden im Source-Listing erheblich erleichtert.

PASS 2:

Im PASS 2 wird der eigentliche Source-Code erzeugt. Dadurch, daß in PASS 2 auch Tabellen erkannt werden und der Reassembler jede Adresse auf Label überprüfen muß, dauert Durchgang 2 natürlich länger als der vorherige!

Der ReAss erkennt Tabellen nur aus dem logischen Zusammenhang. Er simuliert das Programm nicht, wie das Relocator-Programme tun. (In der

nächsten Ausgabe werden wir so ein Programm veröffentlichen!!) Daraus folgt, daß indirekte Zeiger (zum Beispiel auf Texte) nicht erkannt werden und daß dort keine Label gesetzt werden können. Diese müßten dann vom Benutzer selbst ergänzt werden.

Steht im Maschinen-Programm:

```
C123  A9 4C 85 2C
```

so wird daraus, falls \$C123 innerhalb des Programms angesprungen wird, beispielsweise:

```
1210† LBO21  LDA #$4C
1220†      STA $2C
```

Wurde \$2C in einer Symblotabelle mit BASANH = \$2C definiert, erscheint:

```
1210† LBO21  LDA #$4C
1220†      STA BASANH
```

Sie können so durch geschickte Definitionen in einer Symboltabelle die Lesbarkeit Ihres Source-Codes noch erhöhen.

Das Zeichen '†' wird als Formatzeichen am Anfang jeder Zeile verwendet und soll den BASIC-Editor daran hindern, eventuelle folgende Leerzeichen zu ignorieren.

Die Bedienung des ReAss

Unser Reassembler sollte nicht nur für einen speziellen Assembler geeignet sein. Deshalb müssen vom Benutzer einige Definitions-Zeilen, die sich auf den verwendeten Assembler beziehen, selbst eingegeben werden. Hierbei ist zu beachten, daß ReAss nur mit Assemblern verträglich ist, die den BASIC-Editor zur Bearbeitung der Source-Codes benutzen. Assembler mit eigenen Editoren oder eigenen Zeilenformaten können nicht direkt verwendet werden.

Folgende drei Zeilen beziehen sich auf diese Anpassung:

- Definitionszeile: legt den Aufruf des jeweiligen Assemblers fest (zum

Beispiel: SYS 49847)

- Verknüpfungzeile: sorgt dafür, daß bei längeren Source-Codes (Quellen), die nicht mehr in den RAM-Speicher des C 64 passen, eine Verknüpfung der einzelnen Files hergestellt werden kann (zum Beispiel durch .FILE...)

- Endzeile: Manche Assembler benötigen eine Abschluß-Zeile der Art .END.....

Beziehungen zur Peripherie

Für ReAss existieren zwei Möglichkeiten auf ein Maschinen-Programm zuzugreifen:

1. direkt aus dem Speicher
2. von einem Daten-Träger

Objekte im Speicher

Soll der Objekt-Code eines Maschinen-Programms direkt aus dem Speicher reassembliert werden, muß das betreffende Programm vorher im Speicher stehen. Dabei darf das Programm natürlich nicht in den Bereichen liegen, die ReAss selbst benutzt. Freie Bereiche sind:

\$9100-\$9fff, \$c000-\$d000 und \$ea00-\$fffa.

Der Aufruf:

SYS 32771, Anfang, Ende, Start, Gerät, AM

Anfang, Ende und Start sind Adressen. Anfang und Ende beziehen sich auf den Bereich, in dem das Maschinen-Programm im Speicher liegt.

Start ist der Beginn des Speicherbereichs, in den das Programm später assembliert werden soll. Sprünge und Aufrufe im Maschinen-Programm, die innerhalb des Bereichs zwischen Start und dem neuen Ende liegen (Start + Programmlänge), werden mit Labeln versehen. Soll das Programm die alte Start-Adresse behalten ist Anfangs- gleich Start-Adresse.

Beispiel:

sys32771,38000,39000,40960,8,55

Das Programm im Bereich von 38000 bis 39000 wird nach 40969 (bis 41960) reassembliert. Ein JMP \$A100 (JMP 40216) würde in einen JMP LBXXX umgewandelt.

Gerät bezieht sich auf den gewünschten Datenträger für den Source-Code. Es stehen zur Verfügung:

- 1 - Kassette im Commodore-Format
- 7 - Kassette im SuperTape-Format
- 8 - Diskette im DOS-Format.

Wird 7 gewählt, muß vorher SuperTape gestartet werden. Dadurch geht der Bereich \$C000-\$C750 jedoch verloren.

AM ist ein Wert für Spezis, denn er verändert die Adresse 01 in der Zero-Page, was zu erstaunlichen Effekten führt, da hiermit unter anderem das Betriebssystem stillgelegt werden kann. Der Standard-Wert ist '55'.

ReAss verkräftet aber andere Werte, da intern dieser Wert nur zur Umschaltung zwischen ROM und RAM benutzt wird. Wer ein Programm aus dem RAM unter dem Kernal lesen möchte, muß hier den entsprechenden Wert eintragen. Zum Beispiel: '53', wodurch Bit 1 auf '0' gesetzt wird und der RAM-Bereich unter dem KERNAL- und BASIC-ROM angesprochen werden kann.

Grundsätzlich ist zu beachten, daß der Speicherbereich für den Source-Code nicht unbegrenzt ist. Ist der Source-Code zu lang (größer 30719 Byte), wird in jedem Fall auf den Datenträger ausgelagert.

Im Kassetten-Betrieb kann jedoch nur ein Source-File abgespeichert werden. Dies ist sinnvoll, da die meisten Assembler bei Kassetten-Operationen keine Verknüpfung durchführen können.

Achten Sie im Kassetten-Betrieb darauf, daß Ihr Source-Code nicht zu lang wird! Sollte ein Speicher-Überlauf entstehen, versuchen Sie, die Source zu verkürzen, indem Sie zum Beispiel die Trennzeilen verkürzen oder mehrere Befehle in einer Zeile zulassen.

Träger-Objekte

Längere Objekt-Files können nur von Diskette gelesen werden. Die gleiche Diskette dient dann auch zum Abspeichern des reassemblierten Source-Codes. Also: Denken Sie daran, daß auf der Diskette genügend freie Blöcke vorhanden sein müssen.

Zum Aufruf geben Sie ein:

```
SYS 32768
```

Im Bilde des ReAss

Haben Sie den ReAss über den entsprechenden SYS-Befehl aufgerufen, erscheint eine Bildschirm-Maske. In der oberen Hälfte werden die wichtigsten Festlegungen, die Sie getroffen haben, protokolliert.

In der 22. Zeile erscheinen die Aufforderungen des ReAss-Systems zur Bedienung Ihres Peripherie-Geräts.

In der 23. Zeile wird eine Status-Meldung im Disketten-Betrieb angezeigt.

In der 24. Zeile wird mitgeteilt, welche Eingabe erforderlich ist.

Die letzte Zeile dient der Eingabe.

Der ReAss stellt Sie vor folgende Entscheidungen:

Auf Gerät Nr. J/N (obere Zeile)

Sie entscheiden, ob der Source-Code in jedem Fall auf den entsprechenden Datenträger geschrieben werden soll.

Schreibfile (in Zeile 24)

Hierunter legen Sie den Namen fest, unter dem das Source-File auf dem Datenträger gespeichert werden soll. Diese Festlegung ist auf jeden Fall erforderlich, auch wenn Sie vorher mit 'N' geantwortet haben. Falls der Source-Code zu lang werden sollte, werden die Source-Files unter dem Namen abgespeichert, wobei dem Namen eine fortlaufende Kennziffer vorangestellt wird.

Trennen J/N (obere Zeilen)

Um Ihr Source-Listing lesbarer zu gestalten, können Sie nach 'J' in der letzten Zeile eine Zeichen-Kette eingeben, die im Listing als Trennzeile verwendet wird. Trennzeilen werden nach den Befehlen

BRK, JMP, RTS und RTI

eingesetzt.

Haben Sie 'N' gewählt, wird keine Eingabe erwartet.

Kopfzeile (Zeile 24)

In der Kopfzeile können Sie das Source-File Ihrem Assembler anpassen, indem Sie die notwendigen Start-Befehle festlegen.

Beispiel:

```
← 9*4096:.opt p:*= £
```

Diese Zeile enthält zwei Sonderzeichen, die zur Steuerung des Editors und zur automatischen Zuweisung dienen:

← ersetzt den BASIC-Befehl SYS.

£ gilt als Joker für die Start-Adresse, die entweder im SYS-Befehl angegeben wurde oder bei SYS32768 von Diskette gelesen wird. Diese Zeile erscheint im Listing beispielsweise als:

```
sys39864:.opt p:*=$c000
```

Verk.-Zeile (Zeile 24)

Die Verknüpfungs-Zeile dient dazu, die Verbindung zwischen mehreren

Source-Files für Ihren Assembler herzustellen. Sie könnte lauten:

```
.file 8,"F"
```

wobei 'F' wiederum ein Joker ist, für den im Listing der Name des Schreibfiles mit führender Kennziffer eingesetzt wird. Die Kennziffer dient zum Durchnummerieren mehrerer Files.

Endzeile (Zeile 24)

Einige Assembler erwarten als Abschluß des gesamten Source eine spezielle Endzeile. Zum Beispiel:

```
.end 8,"F"
```

Für den Joker gilt das Gleiche wie oben. Eine Verknüpfung über mehrere Files ist für den Kassetten-Betrieb nicht sinnvoll (Sie müßten nämlich hin-her-spulen) und daher nicht vorgesehen.

1 Bef. pro Zeile J/N (obere Zeile)

Sie können mit 'J' ein Listing erzeugen, in dem nur ein Befehl je Zeile aufgeführt ist. Bei 'N' werden die Befehle mit ':' verknüpft. Es werden jedoch nur Zeilen von 40 Zeichen Länge erzeugt.

Beachten Sie, daß bei '1 Befehl pro Zeile' das Source-File erheblich länger wird!

Symtab laden J/N (obere Zeile)

Für manche Assembler können eigene Symboltabellen erzeugt werden. In einer solchen Tabelle können Sie Label und Variablen mit Namen belegen. Beim Reassemblieren werden dann die entsprechenden Adressen durch die jeweiligen Namen ersetzt. Dadurch kann die Lesbarkeit eines Assembler-Programms erheblich verbessert werden.

Im Anschluß finden Sie ein Beispiel für eine Symboltabelle, die sich an den Namensgebungen in den Commodore-Veröffentlichungen orientiert.

Beachten Sie:

Symboltabellen müssen assemblierte Tabellen Ihres Assemblers sein! Sie können hier keine Source-Symboltabelle zuladen. Außerdem darf eine Symboltabelle nicht zu groß sein (kleiner als 4 Blöcke, entspricht 1023 Byte).

Objekte auf Diskette

Erwartet der ReAss das Objekt-File von Diskette, erfolgt noch die Frage:

Lesefile (Zeile 24)

Hier geben Sie den Namen ihres Objekt-Files auf Ihrer Diskette an, das Sie reassemblieren wollen.

Zusatz-Service

ReAss stellt Ihnen noch einige zusätzliche Möglichkeiten zur Verfügung:

Belegung der Funktionstasten:

F1: Senden von Kommandos an die Floppy.

F3: Ausgabe der Direktory.

F5: Anzeige des DOS-Status

F7: Abbruch Direktory-Anzeige im Disk-Modus (sonst siehe Editor-Funktion)

Die Disk-Kommandos entsprechen der String-Syntax innerhalb der PRINT# oder OPENnn,8,mm, Befehle.

Beispiel:

Betätigen Sie F1, erscheint DISK (Zeile 24)

Geben Sie jetzt beispielsweise 'UJ' ein, bewirkt dies ein Reset des DOS Ihrer Floppy. Weitere Möglichkeiten entnehmen Sie bitte Ihrem Floppy-Handbuch.

Nach der obengenannten Sequenz erscheint dort nach Druck auf F5 die Einschalt-Meldung des DOS-Betriebssystems.

Die Direktory wird in der zweiten unteren Hälfte des Bildschirm-Fensters angezeigt. Sie kann mit der Leertaste weitergeblättert oder mit F7 abgebrochen werden.

Zeilen-Editor

Während der Eingabe in der Zeile 25 können Sie mit den Cursor-Tasten für Rechts und Links Ihren eingegebenen Text korrigieren.

CLR/HOME löscht die Eingabe vollständig

INST/DEL löscht ab Cursor-Position nach rechts.

F7 schaltet den Hochkomma-Modus wieder aus.

Fortsetzung auf Seite 32

Symboltabelle Reassembler

1000	open4.4.7	1683	; be oder prozessorstack.	2330	spmul	\$d01c:spexx	\$d01d
1010	sys32768	1690	;	2340	spcp	\$d01e:spcbx	\$d01f
1020	.. \$5000	1700	bed	\$d020:bc011	\$d021	\$d022:bc011	\$d023
1030	opt p1,00	1710	..	\$d024:spm10	\$d025	\$d026:sp6c01	\$d027
1031	sym b o l t a b e l l e	1720	"	\$d028:sp2c01	\$d029	\$d02a:sp4c01	\$d02b
1032	-----	1730	;	\$d02c:sp5c01	\$d02d	\$d02e	\$d02e
1033	-----	1740	buf	\$d100:vfrcq1	\$d101	\$d102:vfrcq	\$d103
1034	diese symboltabelle kann nach	1750	fat	\$d104:vfrcq	\$d105	\$d106:vfrcq	\$d107
1035	bedarf in teilen zum assemblie-	1760	menstr	\$d108:vmem1	\$d109	\$d10a:vmem1	\$d10b
1036	ren mit 'profi-ass' zugeladen	1770	timeut	\$d10c:vmem1	\$d10d	\$d10e:vmem1	\$d10f
1037	werden, die gesamte tabelle ist	1780	qdc01	\$d10f:vmem1	\$d110	\$d111:vmem1	\$d112
1038	zu gross!	1790	max	\$d112:vmem1	\$d113	\$d114:vmem1	\$d115
1039	die bezeichnungs orientieren	1800	kount	\$d115:vmem1	\$d116	\$d117:vmem1	\$d118
1040	sich an veroeffentlichungen des	1810	shriag	\$d118:vmem1	\$d119	\$d11a:vmem1	\$d11b
1041	hauses commodore.	1820	keylog	\$d11a:vmem1	\$d11c	\$d11d:vmem1	\$d11e
1042	-----	1830	autodr	\$d11c:vmem1	\$d11d	\$d11e:vmem1	\$d11f
1043	-----	1840	m5icdr	\$d11d:vmem1	\$d11e	\$d11f:vmem1	\$d11g
1044	-----	1850	rsstat	\$d11e:vmem1	\$d11f	\$d11g:vmem1	\$d11h
1045	-----	1860	baudof	\$d11f:vmem1	\$d11g	\$d11h:vmem1	\$d11i
1046	-----	1870	rodbs	\$d11g:vmem1	\$d11h	\$d11i:vmem1	\$d11j
1047	-----	1880	lrqtmp	\$d11h:vmem1	\$d11i	\$d11j:vmem1	\$d11k
1048	-----	1890	..	\$d11i:vmem1	\$d11j	\$d11k:vmem1	\$d11l
1049	-----	1900	..	\$d11j:vmem1	\$d11k	\$d11l:vmem1	\$d11m
1050	-----	1910	..	\$d11k:vmem1	\$d11l	\$d11m:vmem1	\$d11n
1051	-----	1920	valtyp	\$d11l:vmem1	\$d11m	\$d11n:vmem1	\$d11o
1052	-----	1930	getbfi	\$d11m:vmem1	\$d11n	\$d11o:vmem1	\$d11p
1053	-----	1940	inpfig	\$d11n:vmem1	\$d11o	\$d11p:vmem1	\$d11q
1054	-----	1950	temp	\$d11o:vmem1	\$d11p	\$d11q:vmem1	\$d11r
1055	-----	1960	lastpt	\$d11p:vmem1	\$d11q	\$d11r:vmem1	\$d11s
1056	-----	1970	index	\$d11q:vmem1	\$d11r	\$d11s:vmem1	\$d11t
1057	-----	1980	extab	\$d11r:vmem1	\$d11s	\$d11t:vmem1	\$d11u
1058	-----	1990	arytab	\$d11s:vmem1	\$d11t	\$d11u:vmem1	\$d11v
1059	-----	2000	retop	\$d11t:vmem1	\$d11u	\$d11v:vmem1	\$d11w
1060	-----	2010	mem12	\$d11u:vmem1	\$d11v	\$d11w:vmem1	\$d11x
1061	-----	2020	oldlin	\$d11v:vmem1	\$d11w	\$d11x:vmem1	\$d11y
1062	-----	2030	datlin	\$d11w:vmem1	\$d11x	\$d11y:vmem1	\$d11z
1063	-----	2040	inptr	\$d11x:vmem1	\$d11y	\$d11z:vmem1	\$d120
1064	-----	2050	varnas	\$d11y:vmem1	\$d11z	\$d120:vmem1	\$d121
1065	-----	2060	temp	\$d11z:vmem1	\$d120	\$d121:vmem1	\$d122
1066	-----	2070	factp	\$d120:vmem1	\$d121	\$d122:vmem1	\$d123
1067	-----	2080	facto	\$d121:vmem1	\$d122	\$d123:vmem1	\$d124
1068	-----	2090	facto	\$d122:vmem1	\$d123	\$d124:vmem1	\$d125
1069	-----	2100	facto	\$d123:vmem1	\$d124	\$d125:vmem1	\$d126
1070	-----	2110	facto	\$d124:vmem1	\$d125	\$d126:vmem1	\$d127
1071	-----	2120	facto	\$d125:vmem1	\$d126	\$d127:vmem1	\$d128
1072	-----	2130	facto	\$d126:vmem1	\$d127	\$d128:vmem1	\$d129
1073	-----	2140	facto	\$d127:vmem1	\$d128	\$d129:vmem1	\$d12a
1074	-----	2150	facto	\$d128:vmem1	\$d129	\$d12a:vmem1	\$d12b
1075	-----	2160	facto	\$d129:vmem1	\$d12a	\$d12b:vmem1	\$d12c
1076	-----	2170	facto	\$d12a:vmem1	\$d12b	\$d12c:vmem1	\$d12d
1077	-----	2180	facto	\$d12b:vmem1	\$d12c	\$d12d:vmem1	\$d12e
1078	-----	2190	facto	\$d12c:vmem1	\$d12d	\$d12e:vmem1	\$d12f
1079	-----	2200	facto	\$d12d:vmem1	\$d12e	\$d12f:vmem1	\$d12g
1080	-----	2210	facto	\$d12e:vmem1	\$d12f	\$d12g:vmem1	\$d12h
1081	-----	2220	facto	\$d12f:vmem1	\$d12g	\$d12h:vmem1	\$d12i
1082	-----	2230	facto	\$d12g:vmem1	\$d12h	\$d12i:vmem1	\$d12j
1083	-----	2240	facto	\$d12h:vmem1	\$d12i	\$d12j:vmem1	\$d12k
1084	-----	2250	facto	\$d12i:vmem1	\$d12j	\$d12k:vmem1	\$d12l
1085	-----	2260	facto	\$d12j:vmem1	\$d12k	\$d12l:vmem1	\$d12m
1086	-----	2270	facto	\$d12k:vmem1	\$d12l	\$d12m:vmem1	\$d12n
1087	-----	2280	facto	\$d12l:vmem1	\$d12m	\$d12n:vmem1	\$d12o
1088	-----	2290	facto	\$d12m:vmem1	\$d12n	\$d12o:vmem1	\$d12p
1089	-----	2300	facto	\$d12n:vmem1	\$d12o	\$d12p:vmem1	\$d12q
1090	-----	2310	facto	\$d12o:vmem1	\$d12p	\$d12q:vmem1	\$d12r
1091	-----	2320	facto	\$d12p:vmem1	\$d12q	\$d12r:vmem1	\$d12s
1092	-----	2330	facto	\$d12q:vmem1	\$d12r	\$d12s:vmem1	\$d12t
1093	-----	2340	facto	\$d12r:vmem1	\$d12s	\$d12t:vmem1	\$d12u
1094	-----	2350	facto	\$d12s:vmem1	\$d12t	\$d12u:vmem1	\$d12v
1095	-----	2360	facto	\$d12t:vmem1	\$d12u	\$d12v:vmem1	\$d12w
1096	-----	2370	facto	\$d12u:vmem1	\$d12v	\$d12w:vmem1	\$d12x
1097	-----	2380	facto	\$d12v:vmem1	\$d12w	\$d12x:vmem1	\$d12y
1098	-----	2390	facto	\$d12w:vmem1	\$d12x	\$d12y:vmem1	\$d12z
1099	-----	2400	facto	\$d12x:vmem1	\$d12y	\$d12z:vmem1	\$d130
1100	-----	2410	facto	\$d12y:vmem1	\$d12z	\$d130:vmem1	\$d131
1101	-----	2420	facto	\$d12z:vmem1	\$d130	\$d131:vmem1	\$d132
1102	-----	2430	facto	\$d130:vmem1	\$d131	\$d132:vmem1	\$d133
1103	-----	2440	facto	\$d131:vmem1	\$d132	\$d133:vmem1	\$d134
1104	-----	2450	facto	\$d132:vmem1	\$d133	\$d134:vmem1	\$d135
1105	-----	2460	facto	\$d133:vmem1	\$d134	\$d135:vmem1	\$d136
1106	-----	2470	facto	\$d134:vmem1	\$d135	\$d136:vmem1	\$d137
1107	-----	2480	facto	\$d135:vmem1	\$d136	\$d137:vmem1	\$d138
1108	-----	2490	facto	\$d136:vmem1	\$d137	\$d138:vmem1	\$d139
1109	-----	2500	facto	\$d137:vmem1	\$d138	\$d139:vmem1	\$d13a
1110	-----	2510	facto	\$d138:vmem1	\$d139	\$d13a:vmem1	\$d13b
1111	-----	2520	facto	\$d139:vmem1	\$d13a	\$d13b:vmem1	\$d13c
1112	-----	2530	facto	\$d13a:vmem1	\$d13b	\$d13c:vmem1	\$d13d
1113	-----	2540	facto	\$d13b:vmem1	\$d13c	\$d13d:vmem1	\$d13e
1114	-----	2550	facto	\$d13c:vmem1	\$d13d	\$d13e:vmem1	\$d13f
1115	-----	2560	facto	\$d13d:vmem1	\$d13e	\$d13f:vmem1	\$d13g
1116	-----	2570	facto	\$d13e:vmem1	\$d13f	\$d13g:vmem1	\$d13h
1117	-----	2580	facto	\$d13f:vmem1	\$d13g	\$d13h:vmem1	\$d13i
1118	-----	2590	facto	\$d13g:vmem1	\$d13h	\$d13i:vmem1	\$d13j
1119	-----	2600	facto	\$d13h:vmem1	\$d13i	\$d13j:vmem1	\$d13k
1120	-----	2610	facto	\$d13i:vmem1	\$d13j	\$d13k:vmem1	\$d13l
1121	-----	2620	facto	\$d13j:vmem1	\$d13k	\$d13l:vmem1	\$d13m
1122	-----	2630	facto	\$d13k:vmem1	\$d13l	\$d13m:vmem1	\$d13n
1123	-----	2640	facto	\$d13l:vmem1	\$d13m	\$d13n:vmem1	\$d13o
1124	-----	2650	facto	\$d13m:vmem1	\$d13n	\$d13o:vmem1	\$d13p
1125	-----	2660	facto	\$d13n:vmem1	\$d13o	\$d13p:vmem1	\$d13q
1126	-----	2670	facto	\$d13o:vmem1	\$d13p	\$d13q:vmem1	\$d13r
1127	-----	2680	facto	\$d13p:vmem1	\$d13q	\$d13r:vmem1	\$d13s
1128	-----	2690	facto	\$d13q:vmem1	\$d13r	\$d13s:vmem1	\$d13t
1129	-----	2700	facto	\$d13r:vmem1	\$d13s	\$d13t:vmem1	\$d13u
1130	-----	2710	facto	\$d13s:vmem1	\$d13t	\$d13u:vmem1	\$d13v
1131	-----	2720	facto	\$d13t:vmem1	\$d13u	\$d13v:vmem1	\$d13w
1132	-----	2730	facto	\$d13u:vmem1	\$d13v	\$d13w:vmem1	\$d13x
1133	-----	2740	facto	\$d13v:vmem1	\$d13w	\$d13x:vmem1	\$d13y
1134	-----	2750	facto	\$d13w:vmem1	\$d13x	\$d13y:vmem1	\$d13z
1135	-----	2760	facto	\$d13x:vmem1	\$d13y	\$d13z:vmem1	\$d140
1136	-----	2770	facto	\$d13y:vmem1	\$d13z	\$d140:vmem1	\$d141
1137	-----	2780	facto	\$d13z:vmem1	\$d140	\$d141:vmem1	\$d142
1138	-----	2790	facto	\$d140:vmem1	\$d141	\$d142:vmem1	\$d143
1139	-----	2800	facto	\$d141:vmem1	\$d142	\$d143:vmem1	\$d144
1140	-----	2810	facto	\$d142:vmem1	\$d143	\$d144:vmem1	\$d145
1141	-----	2820	facto	\$d143:vmem1	\$d144	\$d145:vmem1	\$d146
1142	-----	2830	facto	\$d144:vmem1	\$d145	\$d146:vmem1	\$d147
1143	-----	2840	facto	\$d145:vmem1	\$d146	\$d147:vmem1	\$d148
1144	-----	2850	facto	\$d146:vmem1	\$d147	\$d148:vmem1	\$d149
1145	-----	2860	facto	\$d147:vmem1	\$d148	\$d149:vmem1	\$d14a
1146	-----	2870	facto	\$d148:vmem1	\$d149	\$d14a:vmem1	\$d14b
1147	-----	2880	facto	\$d149:vmem1	\$d14a	\$d14b:vmem1	\$d14c
1148	-----	2890	facto	\$d14a:vmem1	\$d14b	\$d14c:vmem1	\$d14d
1149	-----	2900	facto	\$d14b:vmem1	\$d14c	\$d14d:vmem1	\$d14e
1150	-----	2910	facto	\$d14c:vmem1	\$d14d	\$d14e:vmem1	\$d14f
1151	-----	2920	facto	\$d14d:vmem1	\$d14e	\$d14f:vmem1	\$d14g
1152	-----	2930	facto	\$d14e:vmem1	\$d14f	\$d14g:vmem1	\$d14h
1153	-----	2940	facto	\$d14f:vmem1	\$d14g	\$d14h:vmem1	\$d14i
1154	-----	2950	facto	\$d14g:vmem1	\$d14h	\$d14i:vmem1	\$d14j
1155	-----	2960	facto	\$d14h:vmem1	\$d14i	\$d14j:vmem1	\$d14k
1156	-----	2970	facto	\$d14i:vmem1	\$d14j	\$d14k:vmem1	\$d14l
1157	-----	2980	facto	\$d14j:vmem1	\$d14k	\$d14l:vmem1	\$d14m
1158	-----	2990	facto	\$d14k:vmem1	\$d14l	\$d14m:vmem1	\$d14n
1159	-----	3000	facto	\$d14l:vmem1	\$d14m	\$d14n:vmem1	\$d14o
1160	-----	3010	facto	\$d14m:vmem1	\$d14n	\$d14o:vmem1	\$d14p
1161	-----	3020	facto	\$d14n:vmem1	\$d14o	\$d14p:vmem1	\$d14q
1162	-----	3030	facto	\$d14o:vmem1	\$d14p	\$d14q:vmem1	\$d14r
1163	-----	3040	facto	\$d14p:vmem1	\$d14q	\$d14r:vmem1	\$d14s
1164	-----	3050	facto	\$d14q:vmem1	\$d14r	\$d14s:vmem1	\$d14t
1165	-----	3060	facto	\$d14r:vmem1	\$d14s	\$d14t:vmem1	\$d14u
1166	-----	3070	facto	\$d14s:vmem1	\$d14t	\$d14u:vmem1	\$d14v
1167	-----	3080	facto	\$d14t:vmem1	\$d14u	\$d14v:vmem1	\$d14w
1168	-----	3090	facto	\$d14u:vmem1	\$d14v	\$d14w:vmem1	\$d14x
1169	-----	3100	facto	\$d14v:vmem1	\$d14w	\$d14x:vmem1	\$d14y
1170	-----	3110	facto	\$d14w:vmem1	\$d14x	\$d14y:vmem1	\$d14z
1171	-----	3120	facto	\$d14x:vmem1	\$d14y	\$d14z:vmem1	\$d150
1172	-----	3130	facto	\$d14y:vmem1	\$d14z	\$d150:vmem1	\$d151
1173	-----	3140	facto	\$d14z:vmem1	\$d150	\$d151:vmem1	\$d152
1174	-----	3150	facto	\$d150:vmem1	\$d151	\$d152:vmem1	\$d153
1175	-----	3160	facto	\$d151:vmem1	\$d152	\$d153:vmem1	\$d154
1176	-----	3170	facto	\$d152:vmem1	\$d153	\$d154:vmem1	\$d155
1177	-----	3180	facto	\$d153:vmem1	\$d154	\$d155:vmem1	\$d156
1178	-----	3190	facto	\$d154:vmem1	\$d155	\$d156:vmem1	\$d157
1179	-----	3200	facto	\$d155:vmem1	\$d156	\$d157:vmem1	\$d158

Was bringen 80-Zeichen-Karten?

Für Textverarbeitung oder eine umfangreiche Tabellenkalkulation ist der Bildschirm des C-64 mit seinen 40 Zeichen pro Zeile oft zu klein. Um die endgültige Gestaltung eines Schriftstückes beurteilen zu können, wäre eine Schreibbreite, die der des Druckers entspricht, optimal. Für den Commodore 64 werden verschiedene 80-Zeichen-Karten angeboten. Können Sie hier Abhilfe schaffen? Die Problematik soll am Beispiel der von der Firma ROOS gelieferten Karte geschildert werden.

Man erhält eine Steckplatine, die in den Erweiterungsport des C-64 gesteckt wird und von der nunmehr das dem Monitor zugeführte Video-Signal abgegriffen wird. Dazu enthält diese Karte mehrere Ausgänge, eine Cinch-Buchse für das 80-Zeichen Composite-Video-Signal und zwei DIN-Buchsen, eine davon Ausgang für einen Chrominanz/Luminanz-Farbmonitor. Die dritte Buchse ist ein Eingang; ihr kann das bisherige 40-Zeichen-Bild zugeführt werden. Doch dazu unten mehr. Die Platine wird 'nackt', das heißt ohne Gehäuse, ausgeliefert; viele der eingelöteten IC's (16 an der Zahl) sind säuberlich abgeschliffen: da ist man im Störfalle bedingungslos auf den Hersteller angewiesen. Mechanischen Halt findet die Platine auf zwei Schrauben, die mit einer Isolierhülle überzogen sind, damit wenigstens der Arbeitstisch nicht verkratzt.

Softwareseitig enthält die Karte einige Eigenschaften, die die Möglichkeiten des C-64 nicht unbeträchtlich erweitern:

Da ist zunächst die angesprochene Mischmöglichkeit des C-64 Standardbildschirms mit dem 80-Zeichen-Bildschirm, wodurch es nun gelingt, Texte und Grafikdarstellungen einfachst zu mischen und miteinander zu kombinieren.

Weiterhin kann man sich den vertikalen Zeilenabstand von 1 bis zu 8 Pixel einstellen und damit die Lesbarkeit mancher Texte und Listings bedeutend erhöhen. Weiter gibt es eine Feststellmöglichkeit für die obere (erste) Bildschirmzeile, in der man zum Beispiel vermerken kann, woran gerade gearbeitet wird und - last not least - ist die Einblendung einer Uhrzeit möglich. Sie wird nicht aus dem ungenauen TI\$ gewonnen, sondern aus den CIA-Realtime-Registern, die mit Netzfrequenz getriggert werden und genauer als TI oder TI\$ sind.

Da das bisherige Bildschirm-RAM nicht mehr benötigt wird (denn die 80-Zeichen-Karte verfügt über ein eigenes RAM), kann man den freien BASIC-Speicher durch POKE 642,4:SYS 58260 auf 39935 freie Bytes erweitern.

Eine eigene Stromversorgung ist für die 80-Zeichen-Karte nicht erforderlich. Nach allen Prüfungen hat sich die Karte als voll kompatibel zum Commodore-Zeichensatz und zum Commodore-Bildschirmeditor gezeigt.

Daß der Karte eine Kassette bzw. Diskette mit einem deutschen Textverarbeitungsprogramm beiliegt, ist sowohl notwendig als auch erfreulich, denn es ist nicht möglich, eine der gängigen Tabellenkalkulationen (Multiplan, Calc Result...), Textverarbeitungen (Easy Script, Wordcraft, Wordpro, Vizawrite..) oder sonstigen Pakete mit dieser Karte zu betreiben, da sich alle Programme auf das 40-spaltige Format sowie die Originallage des Bildschirmspeichers beziehen. Das allerdings trübt die Freude an einem sonst so nützlichen Zubehörteil gewaltig, denn zumindest von meiner eingefahrenen und bestens bewährten Textverarbeitung möchte ich nicht herunter. ROOS umgeht das Problem elegant: im Gegensatz zu anderen Anbietern, bei denen man auf eventuelle eigene Phantasie angewiesen ist, werden zwei Listings für Konversionsprogramme von Vizawrite- und Easyscript-Files auf den ROOS-BASIC-Textverarbeiter mitgeliefert.

Bleiben noch einige Anmerkungen zur Zeichendarstellung. Der Original-C-64-Zeichensatz ist mit doppelten Pixelreihen aufgebaut und reichlich fett, da nur dies ein auf dem gewöhnlichen Fernseher lesbares Bild ergibt. Wer seinen Rechner an einem Monitor betreibt, wird sicher bald auf einen "schmalen" Zeichensatz umsteigen wollen, dabei allerdings feststellen, daß gleichzeitig die Anforderungen an den Monitor erheblich steigen. Dies ist bei den logischerweise noch schmaleren 80-Spalten-Zeichen noch ausgeprägter. Die ROOS-Karte liefert ein durchaus brauchbares und stabiles Bild. Ohne einen exzellenten Monitor ist der Betrieb einer solchen Karte allerdings von vorneherein ein sinnloses Unterfangen. Den Preis für die Anschaffung eines solchen Monitors müssen Sie also gleich mit einkalkulieren.

ES

Bezug:

DM 286,50
ROOS elektronik,
Kleiner Markt 7
4190 Kleve

Magazin für elektronisch
elrad

**Verlag Heinz Heise
GmbH
Postfach 27 46
3000 Hannover 1**

ist Deutschlands Hobby-Elektronik-Magazin, das die gesamte Zielgruppe der Freizeit-Elektroniker anspricht, vom Elektronik-Anfänger bis zum Bastelprofi.

ELRAD bringt Bauanleitungen und Grundlagenartikel für die Bereiche Audio, Amateurfunk, Modellbau u. Computer.

Verständlich geschrieben, leicht nachvollziehbar.

W5444753 F

3000 Mark wa



Hier r
Sie kö

W56161



und na



W



Wichtig: Werfen Sie einen
damit Ihr Programm auch



De

Warten auf den Gewinner!

Der Wettbewerb geht weiter.

noch einmal kurz die Bedingungen:
können einsenden:

- Grafikprogramme
- Musikprogramme
- Spiele-
- Lernprogramme
- Anwenderprogramme

natürlich völlig neue Programmideen.

Blick in das Kapitel "Hinweise für Autoren",
innerhalb von INPUT 64 lauffähig ist.

Der Rechtsweg ist wie immer ausgeschlossen.

Fortsetzung von Seite 6

1. ABFRAGE: Nach der Eingabe des Datums errechnet das Programm den Wochentag und gibt eventuell vorhandene Termine aus. Mit einem Tastendruck können Sie weiterblättern, falls eine Bildschirmseite nicht ausreicht. Antworten Sie mit 'N' auf die Abfrage 'Schluß', wird der nächste Tag ausgegeben.

2. EINGABE: Sie können zuerst zwischen verschiedenen Terminarten wählen:

1. NORMALE TERMINE
2. DAUERTERMINE/JAHR
3. DAUERTERMINE/WOCHE

Nach der Festlegung geben Sie das Datum (1 und 2) oder den Wochentag ein (3). Anschließend geben Sie Ihren Termin-Text ein (max 37 Zeichen). Antworten Sie mit 'N' auf die 'Schluß-'Abfrage, bleiben Sie im Eingabemodus, bei 'J' kehrt das Programm ins Menue zurück

3. LOESCHEN: Sie beginnen wie oben unter 2. EINGABE. Wenn Sie auf die Abfrage 'Löschen' mit 'J' reagieren, wird der angezeigte Termin gelöscht und sein Speicherplatz freigegeben.

4. DRUCKEN: Sie können Ihren Terminkalender auf Device 4 ausgeben. Ist der Drucker nicht angeschlossen, erfolgt die Ausgabe auf den Bildschirm im gleichen Format. Die STOP-Taste bricht den Druckvorgang ab.

5. LADEN: Lädt eine Termin-Datei mit Namen (siehe 6) von Ihrem Datenträger. Der Speicher wird dabei überschrieben.

6. DATEINAME: Sie dürfen einen Namen mit maximal 14 Zeichen festlegen.

7. SAVEN: Nach der Sicherheitsabfrage werden die Termine aus dem Speicher auf dem Datenträger unter dem Namen (siehe 6) gesichert. Gleichnamige Dateien werden überschrieben.

8. NEUSTART: Sie könne mit einem neuen Tagesdatum beginnen.

@ DEVICE: Beim Start steht hier '1' (Kassette C=-Format). Sie können sich aber auch für 8 (Diskette) und 7 (Kassette SuperTape-Format) entscheiden. Bei 7 muß SuperTape bringen.

Kurs komplett

Jetzt als Sampler:

Die Serie BITS & BYTES IM VIDEO-CHIP

Alle Folgen des Kurses aus den Ausgaben eins bis fünf sind ab sofort gesammelt auf Kassette und Diskette erhältlich. Eine grundlegende Einführung in die Programmierung des Video-Chips, 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

jeweils inclusive Porto und Verpackung.

Bestellungen bitte direkt an den Verlag Heinz Heise GmbH
- Vertrieb
Bissendorfer Str.8

3 Hannover 61

(Verrechnungsscheck oder quittierten Einzahlungsbeleg beilegen!)

64'er Tips

Wie angekündigt: Diesmal geht's rund um die INT-Funktion. Dabei können Sie sogar mit dem Zufall rechnen oder Pfennig-Fuchser um "halbe" Pfennige bringen.

Bitte abtrennen und im Briefumschlag versenden!

Kurs Komplett-Bestellschein

Ich bestelle den gesammelten Kurs BITS & BYTES IM VIDEO-CHIP

auf Kassette zum Preis von 17,80 DM

auf Diskette zum Preis von 24,80 DM

(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

Unterschrift _____

(Ein Verrechnungsscheck/quittierter Zahlungsbeleg liegt bei)

Cave Climber

Bei diesem Spiel befinden Sie sich in einer Höhle. Nun müssen Sie versuchen, sich zum Ausgang am oberen Bildschirmrand durchzuhangeln. Dabei dürfen Sie nicht vergessen, die goldenen Bälle aufzusammeln.

Nur ist das nicht ganz einfach, denn Sie müssen beim Aufsammeln der Bälle den Höhlenbewohnern, Vögeln, Quarks und abgeworfenen Eiern ausweichen.

Gesteuert wird das Spiel mit dem Joystick in Port 2 oder der Tastatur.

Tastaturbelegung:

; = Rechts
: = Links
H = Oben
N = Unten
Space = Feuerknopf

Wenn Sie den Joystick nach oben drücken, wird, falls Sie festen Boden treffen, ein Seil gespannt (nur in Laufrichtung). Gehen Sie unter das Seil und drücken den Joystick nochmals nach oben, so können Sie sich am Seil entlanghangeln. Drücken Sie den Joystick nach unten, wird das Seil wieder losgelassen. Die Lianen, die von den einzelnen Plateaus herunterhängen, können Sie nur von oben erreichen.

Sie haben auch nur ein Seil zur Verfügung, deshalb wird das alte wieder eingezogen.

Möchten Sie gern eine Kaffeepause machen, können Sie das Spiel mit der Funktionstaste F7 unterbrechen. Durch Drücken des Feuerknopfes oder der Space-Taste, wird die Pause wieder aufgehoben.

Bemerkung:

- es gibt vier verschiedene Bilder, wobei das Spielen bis zum Level 25 immer mehr erschwert wird.
- nach jeder zweiten überstandenen Runde erhalten Sie ein "Extra-Leben".
- mit jedem aufgesammelten goldenen Ball wächst Ihre Energie, die Sie gut gebrauchen können.

MANAGER

In "meinem" Verein möchte ich einmal das Sagen haben!! Zuerst würde ich den Trainer rausschmeißen, dann würde ich andere und bessere Spieler verpflichten und überhaupt diese Eintrittspreise...

Die Spielidee

Jetzt können Sie - wenigstens auf der Simulationsebene - Ihren Lieblingsverein für die Saison 85/86 leiten. Die Rahmenbedingungen (alle Spielpaarungen, das Fassungsvermögen der Stadien und ein Anfangskapital) sind im Programm enthalten.

Sie können aber für "Ihren" Verein die folgenden Faktoren beeinflussen:

1. Spielstärke

Durch Kauf oder Verkauf von Spielern steigt oder sinkt die Spielstärke. (Beachten Sie aber beim Einkauf auch die Bilanz!)

2. Eintrittspreise

Der von Ihnen festgelegte Eintrittspreis (vom Programm wird 10,-DM vorgegeben), schlägt sich nicht nur - in Verbindung mit den Zuschauerzahlen - als Einnahme nieder, sondern hat auch indirekt Einfluß auf die Zahl der Zuschauer. (Wenn die Eintrittskarten zu teuer werden, kommen auch weniger Zuschauer)

3. Trainerwechsel

Insgesamt dreimal können Sie den Trainer wechseln. Ob aber dadurch die Spielstärke größer oder geringer wird, hängt vom Glück (oder anders ausgedrückt, von der RND-Funktion) ab.

Nicht beeinflussen können Sie:

1. Spielpaarungen

Wie schon erwähnt sind alle Spielpaarungen der 1. Fußball-Bundesliga für die Saison 85/86 als Daten im Programm enthalten. Da auch die Reihenfolge der Spieltage beachtet wurde, ergibt sich beim Spiel ein realistischer Saison-Durchlauf.

2. Formstärke

In welcher Tages-Form "Ihre" Mannschaft in ein Spiel geht, wird vom Programm berechnet. Hier kann es - wie in der Realität - ganz schöne Überraschungen geben.

3. Spielergebnisse

Natürlich können Sie die Ergebnisse nicht direkt eingeben; das wäre ja noch schöner. Das Spielergebnis "Ihrer" - und aller anderen - Mannschaften, wird durch sehr komplexe Berechnungen bestimmt. In die Berechnung gehen unter anderem die Spielstärke, die Formstärke, der Heimvorteil, der Tabellenplatz und das unberechenbare Glück ein.

4. Tabellenstand

Der Tabellenstand wird nach jedem Spieltag berechnet und richtet sich selbstverständlich nach den Ergebnissen aller bis dahin durchgeführten Spiele.

Der Spielablauf

Sie werden zuerst aufgefordert, aus den 18 Mannschaften der 1. Bundesliga den Verein zu wählen, den Sie managen möchten. Diese Festlegung ist für die Saison nicht mehr änderbar. Sie können also nicht, nachdem Sie einen Verein fast ruiniert haben, sich schnell ein neues "Opfer" aussuchen.

Es erscheinen dann alle Spielpaarungen für den 1. Spieltag. Durch Drücken einer beliebigen Taste gelangen Sie in das Hauptmenue.

Sie finden hier Ihren Gegner und die aktuellen Spiel- und Formstärken sowie ein Auswahlmenue, um die veränderbaren Größen zu beeinflussen oder mit Hilfe diverser Tabellen den Überblick zu wahren. Wenn Sie die Optionen des Menues nicht in Anspruch nehmen wollen, können Sie auch gleich mit "W" zum Spiel übergehen.

Sie gelangen auf die eigentliche Spielseite. Hier läuft Ihr erstes Spiel. Für die nächsten 90 Minuten (selbstverständlich gerafft) sitzen Sie auf der harten Bank am Spielfeld und können nur noch hoffen und fluchen.

Nach dem Schlußpfiff können Sie sich auch die anderen Ergebnisse ansehen. Als letzte Seite folgt die Tabelle.

Nach erneutem Drücken einer Taste gelangen Sie wieder auf die Bildschirmseite, auf der die Paarungen des nunmehr zweiten Spieltages erscheinen.

Der Kreis ist nun geschlossen und Sie können - sicherlich mit wachsender Spannung - alle 34 Spieltage durchspielen.

Da Sie als Manager den großen Überblick behalten müssen, haben Sie noch die Möglichkeit (nach dem ersten Spieltag), vom Hauptmenue aus einen Rückblick anzuwählen. Hier können Sie die sportliche Entwicklung "Ihres" Vereins analysieren.

ReAss bei der Arbeit

Nach erfolgten Eingaben, erscheint die Meldung:

Programm Start-Adresse - End-Adresse

Anschließend durchläuft ReAss den PASS 1 und den PASS 2. Erfolgt beim Ablegen des Source-Codes im Speicher ein Überlauf, wird der Inhalt auf dem entsprechenden Datenträger gesichert.

Wurde direkt aus dem Speicher reassembliert, erscheint bei:

Diskette: 'Disk einlegen'. (Was mit Return bestätigt wird)

Kassette: 'Press Record and Play'.

Denken Sie daran, daß im Kassetten-Betrieb nur ein File gerettet werden kann.

Zum Abschluß verabschiedet sich ReAss mit den Informationen über Programm-Anfang und -Ende und einer Meldung bei eventuellem Speicher-Überlauf. Der Rechner wird in den BASIC-Modus zurückgeschaltet.

Sie können mit der üblichen Sequenz SAVE"Name",8 ihr Source-File abspeichern, falls dies nicht gewählt wurde oder automatisch geschah. Mit LIST können Sie sich das erzeugte Source-Listing ansehen.

... die Zeitschrift mit Durchblick!

ct *magazin für
computer
technik*

die Herausforderung für Insider,
der Einstieg für Einsteiger,
ein neuer Anfang für alle.

SID-Kurs Teil 4

Die Programmierung mehrstimmiger Musikstücke ist Thema von Teil 4 der Serie über den Soundchip im C64. Die zu den unten abgedruckten Listings gehörigen Programme können Sie aus dem Kurs durch die übliche Tastenkombination CTRL und s abspeichern.

Geben Sie ab Zeile 500 eigene Notenwerte in DATA-Zeilen ein.

Listing 1

```
5 rem beispielprogramm 1 teil 4
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+10
20 poke54277+7,01*16+04:poke54278+7,6*16+10:poke54272+24,15
100 readl1,h1:poke54276,32:ifh1=0then120
110 poke54276,33:poke54272,11:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then140
130 poke54276+7,33:poke54272+7,12:poke54273+7,h2
140 forw=1to200:nextw:ifl1<>-1then100
500 rem datas
510 data 180,8,180,8,-1,0,0,0
```

Listing 2

```
5 rem beispielprogramm 2 teil 4
10 forx=0to24:poke54272+x,0:nextx:poke54277,1*16+4:poke54278,6*16+10
20 poke54277+7,1*16+4:poke54278+7,6*16+10:poke54272+24,15
25 poke54277+14,1*16+4:poke54278+14,6*16+10
100 readl1,h1:poke54276,32:ifh1=0then120
110 poke54276,33:poke54272,11:poke54273,h1
120 readl2,h2:poke54283,32:ifh2=0then133
130 poke54276+7,33:poke54272+7,12:poke54273+7,h2
133 readl3,h3:poke54290,32:ifh3=0then140
135 poke54276+14,33:poke54272+14,13:poke54273+14,h3
140 forw=1to200:nextw:ifl1<>-1then100
500 rem datas
510 data 0,8,0,32,0,128,-1,0,0,0,0,0
```

HINWEISE FÜR AUTOREN

Falls Sie uns ein Programm zur Veröffentlichung anbieten wollen, beachten Sie bitte folgende Hinweise:

Daß Ihre Programme lauffähig und absturzsicher sein müssen, versteht sich von selbst. Im einzelnen heißt das:

Kein Programmabbruch durch Fehlermeldungen, alle möglichen Eingabefehler werden abgefangen, die Bildschirmmaske wird nicht zerstört und so weiter.

Das Programm sollte in C-64-BASIC oder in 6502/6510-Assembler geschrieben sein. Als Hilfsmittel können die bisher in INPUT 64 erschienenen Tools (Hiresspeed, Sprite-Befehle usw.) benutzt werden. Sie können allerdings auch eine andere Hochsprache (zum Beispiel PASCAL) verwenden, für die es möglich ist, selbstständig lauffähige Compile zu erzeugen. Näheres erfragen Sie bitte bei der Redaktion.

Ihr Programm sollte nicht länger als 100 Blöcke (25 KByte) sein.

Alle Programme müssen auch ohne Floppy lauffähig sein. Floppy-Betrieb optional ist erlaubt und gewünscht.

Senden Sie uns Ihre Programme bitte auf Kasette oder Diskette mit kommentiertem Listing und Beschreibung.

Sowohl Autostart als auch Listschutz erschweren uns nur die Arbeit! Wir werden deshalb Programme, deren Analyse absichtlich erschwert wurde, zukünftig ungeprüft zurücksenden.

Wichtig: Sie müssen im Besitz der vollen Urheberrechte an Ihrem Programm sein und überlassen es uns zur Erstveröffentlichung.

Außerdem gibt es einige, durch das INPUT 64-Betriebssystem bedingte, programmiertechnische Erfordernisse:

1. Belegen Sie nur den Bereich des normalen BASIC-RAM (\$0800-\$9FFF) und unter dem BASIC-ROM (\$A000-\$BFFF).
2. Jede Benutzung von Zero-Page-Adressen, Veränderung der Betriebssystem-Vektoren (Interrupt, Tastatur, etc.) muß genau dokumentiert sein.
3. Die Programme müssen als BASIC-File zu laden und mit RUN zu starten sein.
4. Die CTRL-Taste darf nicht benutzt werden.

Und geben Sie bitte auf Listings, Kassetten, Disks und so weiter den Programmnamen sowie Ihre Anschrift an.

Hilfsprogramme: Sound-Erweiterung

Diese Tools ermöglichen dem BASIC-Programmierer eine einfache Nutzung der Sound-Möglichkeiten ohne umständliche POKE-Befehle.

Die Befehle:

Version 1 (\$C800)	Version 2 (\$0900)	
SYS 51200,vo	SYS 2304,vo	Lautstärke vo wird eingestellt.
SYS 51314,st,we(,pu)	SYS 2418,st,we(,pu)	Für Stimme st (0,1,2) wird Wellenform we (16,32,64,128 oder Mischformen) eingestellt. Ist 64 (Rechteck) enthalten, muss noch pu angegeben werden. Dies ist die Pulsbreite in Prozent.
SYS 51210,st,a,d,s,r	SYS 2314,st,a,d,s,r	Für Stimme st wird die Hüllkurve eingestellt.
SYS 51404	SYS 2508	Ein voreingestellter Ton wird gespielt. (BEEP)
SYS 51404,st,freq,da	SYS 2508,st,freq,da	Stimme st wird mit Frequenz freq gespielt. Nach da Interrupts (=1/60sek.) wird RELEASE eingeleitet. (PLAY)
SYS 51404,st,"note",da	SYS 2508,st,"note",da	dto., aber statt Frequenz kann eine Note angegeben werden.

Für A,D,S,R und VO können Werte von 0 bis 15 eingesetzt werden. Für die Pulsbreite (PU) sind Werte von 0 bis 99 zulässig, für DA werden Werte von 0 bis 255 akzeptiert.

Der Wert ST gibt die angesprochene Stimme des SID an, dabei werden Werte von 0 bis 2 ausgewertet.

Die Wellenformen entsprechen den üblichen Werten (16,32,64,128) oder Mischformen daraus. Die Frequenz (FREQ) beim PLAY-Befehl kann Werte von 0 bis 4095 annehmen.

Wird eine Note eingegeben, muß dies natürlich ein String sein; er kann eine maximale Länge von drei Zeichen haben. "C1" spielt das eingestrichene C, "C#4" spielt das Cis der vierten Oktave, die anderen Noten werden nach dem selben Format angegeben. Bei nicht zulässigen Angaben wird ein ILLEGAL DIRECT ERROR ausgegeben.

Dabei ist noch anzumerken, daß auch enharmonische Verwechslungen richtig gespielt werden.

Um eine gespielte Note zu unterbrechen, kann man POKE 51820+ST,1 bzw. POKE 2924+ST,1 eingeben, und beim nächsten IRQ wird die Release-Phase der Stimme eingeleitet.

Man kann auch überprüfen, ob diese schon eingeleitet ist, dazu sind die oben angeführten Adressen mit PEEK auszulesen und zu überprüfen, ob die Speicherstelle schon auf Null gesetzt war. Ist dies der Fall, befindet sich diese Stimme in der Release-Phase.

Um festzustellen, ob für alle Stimmen die Release-Phase eingeleitet wurde, kann man abfragen, ob Adresse 788 den Wert 49 beinhaltet. Wenn dies der Fall ist, stehen die Gate-Bits aller drei Stimmen auf Null.

Bitte beachten Sie, daß vor dem Aufruf der jeweiligen Stimme (SYS 2508,st,freq,da / SYS 51404,st,freq,da) die Parameter für Lautstärke, Wellenform und Hüllkurve durch die entsprechenden SYS-Aufrufe übergeben werden müssen. Dies gilt natürlich nicht für den voreingestellten Ton (BEEP).

Für die einwandfreie Funktion der Sound-Erweiterung darf der Interrupt-Vektor nicht für andere Zwecke verwendet werden, da dieser vom PLAY-Befehl intensiv genutzt wird.

Im Demo-Programm kann zwischen zwei Versionen ausgewählt werden. Die erste verschiebt nach einem RUN das Maschinenprogramm nach \$C800 (verträglich mit Supertape) und löscht sich dann selbst. Das heißt, der BASIC-Speicherbereich wird nicht beeinflusst.

Die zweite Version legt nach dem Starten den BASIC-Anfang nach \$0C00 (3072 dez.) und führt dann ein RUN aus. Man kann anschließend ein BASIC-Programm laden (mit Sekundäradresse 0) oder neu editieren. Vor dem Speichern müssen Sie POKE 44,8 eingeben, um die Sound-Tools mit auf Ihren Datenträger zu überspielen. Dann steht die Erweiterung, wenn das Programm später geladen und gestartet wird, sofort zur Verfügung und das BASIC-Programm wird abgearbeitet.

Listing-Sound-Erweiterung

00010	LSAUBERHÄLTUNGS (V, 2, 8, 1985)	06550	CFX #16	10000	10000	10000	10000
00020	VON SVEN HESSEL BUCH	06600	BCS I1,0A	06700	06750	06800	06850
00030		06900	LDX #14	06950	07000	07050	07100
00040		07200	JSR GETBYTE	07300	07350	07400	07450
00050		07500	BCS I1,0U	07600	07650	07700	07750
00060		07800	CFX #7	07900	07950	08000	08050
00070		08100	BCS I1,0U	08200	08250	08300	08350
00080		08400	CFX #0	08500	08550	08600	08650
00090		08700	BEO VOICED	08800	08850	08900	08950
00100		08950	CFX #7	09000	09050	09100	09150
00110		09200	CFX #7	09300	09350	09400	09450
00120		09500	CFX #7	09600	09650	09700	09750
00130		09800	LDX #12	09900	09950	10000	10050
00140		10100	LDX #4	10200	10250	10300	10350
00150		10400	LDX #14	10450	10500	10550	10600
00160		10650	RTS	10700	10750	10800	10850
00170		10900	RTS	10950	11000	11050	11100
00180		11150	RTS	11200	11250	11300	11350
00190		11400	RTS	11450	11500	11550	11600
00200		11650	RTS	11700	11750	11800	11850
00210		11900	RTS	11950	12000	12050	12100
00220		12150	RTS	12200	12250	12300	12350
00230		12400	RTS	12450	12500	12550	12600
00240		12650	RTS	12700	12750	12800	12850
00250		12900	RTS	12950	13000	13050	13100
00260		13150	RTS	13200	13250	13300	13350
00270		13400	RTS	13450	13500	13550	13600
00280		13650	RTS	13700	13750	13800	13850
00290		13900	RTS	13950	14000	14050	14100
00300		14150	RTS	14200	14250	14300	14350
00310		14400	RTS	14450	14500	14550	14600
00320		14650	RTS	14700	14750	14800	14850
00330		14900	RTS	14950	15000	15050	15100
00340		15150	RTS	15200	15250	15300	15350
00350		15400	RTS	15450	15500	15550	15600
00360		15650	RTS	15700	15750	15800	15850
00370		15900	RTS	15950	16000	16050	16100
00380		16150	RTS	16200	16250	16300	16350
00390		16400	RTS	16450	16500	16550	16600
00400		16650	RTS	16700	16750	16800	16850
00410		16900	RTS	16950	17000	17050	17100
00420		17150	RTS	17200	17250	17300	17350
00430		17400	RTS	17450	17500	17550	17600
00440		17650	RTS	17700	17750	17800	17850
00450		17900	RTS	17950	18000	18050	18100
00460		18150	RTS	18200	18250	18300	18350
00470		18400	RTS	18450	18500	18550	18600
00480		18650	RTS	18700	18750	18800	18850
00490		18900	RTS	18950	19000	19050	19100
00500		19150	RTS	19200	19250	19300	19350
00510		19400	RTS	19450	19500	19550	19600
00520		19650	RTS	19700	19750	19800	19850
00530		19900	RTS	19950	20000	20050	20100
00540		20150	RTS	20200	20250	20300	20350
00550		20400	RTS	20450	20500	20550	20600
00560		20650	RTS	20700	20750	20800	20850
00570		20900	RTS	20950	21000	21050	21100
00580		21150	RTS	21200	21250	21300	21350
00590		21400	RTS	21450	21500	21550	21600
00600		21650	RTS	21700	21750	21800	21850
00610		21900	RTS	21950	22000	22050	22100
00620		22150	RTS	22200	22250	22300	22350
00630		22400	RTS	22450	22500	22550	22600
00640		22650	RTS	22700	22750	22800	22850
00650		22900	RTS	22950	23000	23050	23100
00660		23150	RTS	23200	23250	23300	23350
00670		23400	RTS	23450	23500	23550	23600
00680		23650	RTS	23700	23750	23800	23850
00690		23900	RTS	23950	24000	24050	24100
00700		24150	RTS	24200	24250	24300	24350
00710		24400	RTS	24450	24500	24550	24600
00720		24650	RTS	24700	24750	24800	24850
00730		24900	RTS	24950	25000	25050	25100
00740		25150	RTS	25200	25250	25300	25350
00750		25400	RTS	25450	25500	25550	25600
00760		25650	RTS	25700	25750	25800	25850
00770		25900	RTS	25950	26000	26050	26100
00780		26150	RTS	26200	26250	26300	26350
00790		26400	RTS	26450	26500	26550	26600
00800		26650	RTS	26700	26750	26800	26850
00810		26900	RTS	26950	27000	27050	27100
00820		27150	RTS	27200	27250	27300	27350
00830		27400	RTS	27450	27500	27550	27600
00840		27650	RTS	27700	27750	27800	27850
00850		27900	RTS	27950	28000	28050	28100
00860		28150	RTS	28200	28250	28300	28350
00870		28400	RTS	28450	28500	28550	28600
00880		28650	RTS	28700	28750	28800	28850
00890		28900	RTS	28950	29000	29050	29100
00900		29150	RTS	29200	29250	29300	29350
00910		29400	RTS	29450	29500	29550	29600
00920		29650	RTS	29700	29750	29800	29850
00930		29900	RTS	29950	30000	30050	30100
00940		30150	RTS	30200	30250	30300	30350
00950		30400	RTS	30450	30500	30550	30600
00960		30650	RTS	30700	30750	30800	30850
00970		30900	RTS	30950	31000	31050	31100
00980		31150	RTS	31200	31250	31300	31350
00990		31400	RTS	31450	31500	31550	31600
01000		31650	RTS	31700	31750	31800	31850
01010		31900	RTS	31950	32000	32050	32100
01020		32150	RTS	32200	32250	32300	32350
01030		32400	RTS	32450	32500	32550	32600
01040		32650	RTS	32700	32750	32800	32850
01050		32900	RTS	32950	33000	33050	33100
01060		33150	RTS	33200	33250	33300	33350
01070		33400	RTS	33450	33500	33550	33600
01080		33650	RTS	33700	33750	33800	33850
01090		33900	RTS	33950	34000	34050	34100
01100		34150	RTS	34200	34250	34300	34350
01110		34400	RTS	34450	34500	34550	34600
01120		34650	RTS	34700	34750	34800	34850
01130		34900	RTS	34950	35000	35050	35100
01140		35150	RTS	35200	35250	35300	35350
01150		35400	RTS	35450	35500	35550	35600
01160		35650	RTS	35700	35750	35800	35850
01170		35900	RTS	35950	36000	36050	36100
01180		36150	RTS	36200	36250	36300	36350
01190		36400	RTS	36450	36500	36550	36600
01200		36650	RTS	36700	36750	36800	36850
01210		36900	RTS	36950	37000	37050	37100
01220		37150	RTS	37200	37250	37300	37350
01230		37400	RTS	37450	37500	37550	37600
01240		37650	RTS	37700	37750	37800	37850
01250		37900	RTS	37950	38000	38050	38100
01260		38150	RTS	38200	38250	38300	38350
01270		38400	RTS	38450	38500	38550	38600
01280		38650	RTS	38700	38750	38800	38850
01290		38900	RTS	38950	39000	39050	39100
01300		39150	RTS	39200	39250	39300	39350
01310		39400	RTS	39450	39500	39550	39600
01320		39650	RTS	39700	39750	39800	39850
01330		39900	RTS	39950	40000	40050	40100
01340		40150	RTS	40200	40250	40300	40350
01350		40400	RTS	40450	40500	40550	40600
01360		40650	RTS	40700	40750	40800	40850
01370		40900	RTS	40950	41000	41050	41100
01380		41150	RTS	41200	41250	41300	41350
01390		41400	RTS	41450	41500	41550	41600
01400		41650	RTS	41700	41750	41800	41850
01410		41900	RTS	41950	42000	42050	42100
01420		42150	RTS	42200	42250	42300	42350
01430							

Für Nachzügler und "Spätzügler"

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. Richten Sie Ihre Bestellung bitte direkt an den Verlag; die Lieferung erfolgt nur gegen Vorkasse (Verrechnungsscheck).

Kassettenversion: DM 12,80;
Diskettenversion: DM 19,80.
(incl. Porto und Verpackung)



INPUT 64
Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

ÜBRIGENS: Abonnenten sparen den Preis von zwei INPUT 64 - Ausgaben pro Jahr (Diskettenversion). Und zwar ohne den üblichen "Pferdefuß", denn das INPUT 64 - Abo ist jederzeit mit Wirkung ab der übernächsten Ausgabe kündbar!

Aus dem Inhalt: Ausgabe 1 - Dateiverwaltung, drei (!) Spiele * Ausgabe 2 - Textprogramm, Zeichengenerator * Ausgabe 3 - Spriteeditor, Maschinensprache-Monitor * Ausgabe 4 - SuperTape DII, Grafikhilfe, Urlaubskalender * Ausgabe 5 - Mathe mit Nico, Talk to me (Dialogsimulation), Hintergrundmonitor * Ausgabe 6 - Haushaltsbuchführung, Recorder-Justage, Textadventure, BASIC-Compactor * Ausgabe 7 - Scroll Editor, TapeCopy, HiFi-Boxenberechnung.

Rätselecke

Wir freuen uns immer wieder über die Vielzahl der eingesandten Lösungen. Auch diesmal haben viele Leser einen Lösungsvorschlag zu unserer Aufgabe eingesandt, ein vorgegebenes Datenfeld möglichst schnell zu sortieren.

Den Hauptpreis hat ein Leser aus dem deutschsprachigen Ausland gewonnen:

Christof Mader aus Graz in Österreich

Alle Gewinner werden von uns schriftlich benachrichtigt.

Bei der Auswertung dieses Rätsels haben es uns die Einsender sehr schwer gemacht, die ersten zehn Gewinner aus den Einsendungen herauszufinden.

Einige Programme waren zwar etwas schneller als das von Christof Mader (sie brauchten zum Sortieren so um die fünf bis fünfzehn Sekunden), aber entweder wurden zusätzliche Felder definiert oder benutzt - einige Programmierer übersahen einfach, daß die Feld-Variable F\$(0) noch zur Benutzung frei war - oder es wurde nur nach dem ersten Buchstaben sortiert, wie zum Beispiel hier:

APPS V.
ALBRECHT P.

Diese beiden Namen müßten selbstverständlich getauscht werden.

In manchen Fällen wurde auch das Ende-Kennzeichen (*) nicht berücksichtigt und so weiter und so weiter.

Rätsellösung

```
300 AN=I-1:S=INT(AN/1.39)
310 FORI=1TOAN-S
320 IFF$(I)>F$(I+S)THENH$=F$(I+S):F$(I+S)=F$(I):F$(I)=H$
330 NEXT:IFS<>2THENS=INT(S/1.39):GOTO310
340 FORI=2TOAN:IFF$(I)>=F$(I-1)THEN390
350 H$=F$(I):FORE=I-1TO1STEP-1:F$(E+1)=F$(E):IFH$<=F$(E-1)THEN380
370 F$(E)=H$:GOTO390
380 NEXTE
390 NEXTI
```

Oder so:

Platz zwei - entwickelt von Peter Firmenich

```
300 N=I-1:A=49152:B=49302:C=1:D=N:I=1
310 E=C:F=D
320 IFF$(E)<=F$(F)THEN420
330 Z$=F$(E):F$(E)=F$(F):F$(F)=Z$
360 E=E+1:IFE=FTHEN440
380 IFF$(E)<=F$(F)THEN360
390 Z$=F$(E):F$(E)=F$(F):F$(F)=Z$
420 F=F-1:IFF<>ETHEN320
440 F=F+1:IFF>=DTHEN480
460 POKEA+I,F:POKEB+I,D:I=I+1
480 E=E-1:IFC>=ETHEN520
500 D=E:GOTO310
520 I=I-1:C=PEEK(A+I):D=PEEK(B+I):IFI>0THEN310
```

Platz drei - entwickelt von Bernd Leps

```
300 N=I-1:S=820:B=20:POKES,1:POKE840,N:L=1:R=N
340 I=L:J=R:A$=F$((L+R)/2)
360 IFA$<F$(J)THENJ=J-1:GOTO360
370 FORI=ITON:IFF$(I)<ASTHENNEXT
380 IFI>JTHEN450
390 H$=F$(I):F$(I)=F$(J):F$(J)=H$:I=I+1:J=J-1:IFI<=JTHEN360
450 IFI<RTHENS=S+1:POKES,I:POKEB+S,R
460 R=J:IFL<RTHEN340
470 IFS=820THEN998
490 L=PEEK(S):R=PEEK(B+S):S=S-1:IFL<RTHEN340
520 IFS=820THEN490
```

Wenn man sich diese drei Programm-Listings ansieht, merkt man, daß sie gar nicht direkt miteinander verglichen werden können.

Jeder der drei Erstplatzierten hat das Problem auf eine andere Art und Weise gelöst.

Alle drei Programme könnten aber noch so verändert werden, daß sie noch schneller laufen. Zum Beispiel durch Ausschalten des Bildschirms (Bit vier in Register 53265 löschen) oder Ausschalten des Interrupts (Bit

eins im Register 56334 löschen). Wobei zu beachten ist, daß nach dem Programm-Durchlauf der alte Wert im Register 56334 wieder herzustellen ist.

Viele Leser haben nicht nur die Lösung zum Rätsel eingeschickt, sondern auch einen Brief dazu geschrieben. Hier sind zwei Kostproben:

...Die Beschäftigung mit dem gestellten Problem hat Freude gemacht, auch falls vielleicht nichts dabei herauskommt.

J. Rätsch

...es ist eine gute Idee, statt eines Rätsels eine Programmaufgabe zu stellen, da anzunehmen ist, daß in jedem INPUT 64-Bezieher noch soviel Energie und Freude an der Sache steckt, daß eine gestellte Aufgabe als persönliche Herausforderung angesehen und deren Lösung sofort in Angriff genommen wird.

In meinem Falle war das ebenso und dabei habe ich auch noch neue Erkenntnisse gewonnen und Spass gehabt.

R. Strauch

Diesen Meinungen haben wir uns angeschlossen. Wir stellen Ihnen deshalb eine neue Aufgabe, die auch wieder aus einer Mischung von Rätsel und Programmierung besteht.

Jetzt geht es los!

Wir kommen zu unserer neuen Aufgabe. Im Mittelpunkt steht dabei der Satz: Send more Money

Übersetzt lautet er: Schick mehr Geld. Formatiert man den Satz etwas anders, dann kann man daraus eine tolle Knobelei machen. Für diese drei Wörter, aus denen der Satz besteht, sollen drei Zahlen eingesetzt werden, wobei jeder Buchstabe eine Ziffer darstellt.

SEND
+ MORE

MONEY

Stellen Sie sich vor, daß jeder Buchstabe eine Ziffer darstellt. Die Wörter 'SEND', 'MORE' und 'MONEY' entsprechen dann drei verschiedenen Zahlen. Wenn Sie die Zahlen-Codes der Wörter 'SEND' und 'MORE' addiert haben, sollten Sie die entsprechende Zahlen-Kombination des Wortes 'MONEY' erhalten.

Das von Ihnen zu erstellende Programm sollte in der Lage sein, diese und ähnliche Aufgaben lösen zu können. Auch dürfte das Programm nicht nur für jeden Buchstaben alle Ziffern durchprobieren (das kann ja jeder), sondern sich durch eigene "Intelligenz" auszeichnen.

Außerdem möchten wir von Ihnen einen ähnlichen, sinnvollen Spruch wie oben haben, der ebenfalls von Ihrem Programm enträtselt werden muß.

Für das letzte Rätsel verlangten wir ausschließlich eine reine BASIC-Lösung. Diesmal erweitern wir den Sprachschatz und lassen noch eine weitere Programmier-Sprache zu, nämlich PASCAL. Für die Lösungen in PASCAL sind die gleichen Preise ausgesetzt wie für die BASIC-Lösung.

Das von Ihnen eingesandte Programm soll nach dem Starten die gestellte Aufgabe in möglichst kurzer Zeit lösen und zum Schluß die drei Zahlen untereinander stehend auf dem Bildschirm anzeigen.

Wenn Sie an diesem Rätsel teilnehmen wollen, schicken Sie uns Ihre Lösung als BASIC- oder PASCAL-Listing zu. Bitte notieren Sie auch auf dem Lösungsblatt die Zeit, die Ihr Programm zum Erstellen der Lösung benötigt (Bei PASCAL-Programmen den verwendeten PASCAL-Compiler).

Der Einsendeschluß für Ihr Listing ist der 1. Oktober 1985 (Datum des Poststempels). Der Rechtsweg ist wie immer ausgeschlossen.

Die Anschrift entnehmen Sie bitte dem Impressum.

Gewinnen können Sie selbstverständlich auch.

Der Autor, der das schnellste Programm zur Lösung der gestellten Aufgabe entwickelt hat, erhält ein Jahresabonnement INPUT 64 (wahlweise Kassetten- oder Diskettenversion). Außerdem warten wieder neun Buchpreise auf einen Empfänger.

Wir wünschen Ihnen viel Erfolg beim Entwickeln des Programmes. Bitte vergessen Sie nicht, auf dem Listing die Laufzeit des Programmes anzugeben.

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. Sie können dazu eine Reinigungskassette verwenden, gründlicher und besser ist es aber, ein Wattestäbchen und Reinigungsflüssigkeit zu verwenden. Die genaue Vorgehensweise ist im Handbuch der Data-sette beschrieben.

Führt auch dies nicht zum Erfolg, ist wahrscheinlich der Tonkopf Ihres Gerätes verstellt. Dieser Fehler tritt leider auch bei fabrikneuen Geräten auf.

TONKOPF SELBST JUSTIEREN

Wir haben 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 des Bandes befindet. Gehen Sie bitte folgendermaßen vor:

Nehmen Sie sich einen kleinen Schraubenzieher und werfen Sie einen Blick auf Ihre Data-sette. Über der REWIND-Taste, in etwa 0,5 cm Abstand vom Kassettenfach, befindet sich ein kleines Loch. Wenn Sie die PLAY-Taste drücken und durch dieses Loch schauen, sehen Sie den Kopf der Justierschraube für die Spurlage.

Legen Sie nun die zweite Seite von INPUT-64 ein, und spulen Sie zum Bandanfang. Drücken Sie jetzt die PLAY-Taste, lassen Sie das Band 45 Sekunden laufen, dann stoppen und umdrehen. Das Band steht jetzt kurz vor dem Synchro-Signal.

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.

Geschieht dies nicht, dann verdrehen Sie die oben beschriebene Einstellschraube. Markieren Sie sich vorher die alte Stellung der Schraube, sonst kann es bei grob verstelltem Tonkopf passieren, daß Sie mit dejustiertem Tonkopf geschriebene Kassetten nicht mehr lesen können. Aber

Vorsicht: ganz langsam drehen, ohne dabei Druck auszuüben! Verdrehen Sie die Schraube nicht mehr als eine Umdrehung in jede Richtung. Nach etwas Ausprobieren wird der Bildschirm gleichmäßig die Farbe wechseln. Soweit die Grobeinstellung.

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

Besitzer der Ausgabe 6/85 können es sich einfacher machen. Das dort veröffentlichte Programm RECORDER-JUSTAGE macht die Einstellung des Daten-Recorders zum Kinderspiel.

Listing justage

```
800 fori=49199to49410:readd:ps=ps+d:pokei.d:next
900 ifps<>24716thenprint"falsch abgetippt - fehler korrigieren!":end
950 print"o.k."
970 sys12*16+13+11*16+10
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.

Ab 21. Oktober an Ihrem Kiosk: INPUT 64 Ausgabe 10/85

Wir bringen unter anderem:

— JETFLIGHT

Geschrieben von einem Berufspiloten: ein Flugsimulator mit den Daten der wichtigsten westdeutschen Flughäfen. Simuliert wird das Verhalten einer großen Verkehrsmaschine (DC 10, Boeing 747) im Instrumentenflug. Fast eine Übung für den Pilotenschein!

— ReLoc

Nachdem wir in dieser Ausgabe einen Re-Assembler vorgestellt haben, folgt gleich ein Relocator. Dieses Programm verschiebt Maschinenprogramme im Speicher und führt die notwendigen Änderungen durch. So ist zum Beispiel die Anpassung von SuperTape an jeden beliebigen Adressbereich möglich!

— und außerdem:

Hilfsprogramme * Serien: SID-Kurs, Mathe mit Nico, 64er-Tips * News * Spiele * und so weiter.

c't-Magazin für Computertechnik

c't 10/85 - jetzt am Kiosk

Prüfstand: Commodore 128, Schneider CPC6128, Robotext-Modul für C64, Operator II * Serien: Einsteigen in CP/M, Des Schneiders Kern, Compiler, Dr. Osborne Kit * Applikation: V-Chips von NEC ersetzen 8080 und 8086 * Software-Know-how: Einführung in Prolog * Projekte: Universelle Busmonitore für Z80 und 68000, Unicard, I/O-Karte für Apple u.v.a.m

elrad-Magazin für Elektronik

Heft 10/85 - ab 30.9. am Kiosk

* Technische Daten komplett: Modularer Vorverstärker 4 * Universal-Steuerplatine für Synthesizer * Elektronisches Kunstwerk: Perpetuum Pendulum * VCA mit OTA: Spannungsgesteuerter Verstärker * Wie kommt das Bild aufs Band: Video-Aufzeichnung * u.v.a.m

(Änderungen vorbehalten)

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) 5352-0

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 Mohle
Karl-Friedrich Probst
Jurgen Seeger

Ständige Mitarbeiter:

Peter S. Berk
Irene Heinen
Peter Sager
Hajo Schulz
Peter Seeliger
Eckart Steffens

Vertrieb: Anita Kreutzer

Redaktion, Anzeigenverwaltung, Abonnementsverwaltung:

Verlag Heinz Heise GmbH
Postfach 610407
3000 Hannover 61
Tel.: (05 11) 5352-0

Grafische Gestaltung:

Wolfgang Ulber, Dirk Wollschläger

Herstellung: Heiner Niens

Lithografie:

Köhler & Lippmann, Braunschweig.

Druck:

Leunisman GmbH, Hannover
Hahn-Druckerei, Hannover

Konfektionierung:

Lettershop Brendler, Hannover

Kassettenherstellung:

SONOPRESS GMBH, Gütersloh

INPUT 64 erscheint monatlich.

Einzelpreis DM 12,80
Jahresabonnement Inland Kassette DM 140,-
Diskette DM 196,-
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

Eine Verantwortung für die Richtigkeit der Veröffentlichungen und die Lauffähigkeit der Programme kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden.

Die gewerbliche Nutzung ist ebenso wie die private Weitergabe von Kopien aus INPUT 64 nur mit schriftlicher Genehmigung des Herausgebers zulässig. Die Zustimmung kann an Bedingungen geknüpft sein. Bei unerlaubter Weitergabe von Kopien wird vom Herausgeber - unbeschadet zivilrechtlicher Schritte - Strafantrag gestellt.

Honorierte Arbeiten gehen in das Verfügungsrecht des Verlages über. Nachdruck nur mit Genehmigung des Verlages. Mit der Übergabe der Programme und Manuskripte an die Redaktion erteilt der Verfasser dem Verlag das Exklusivrecht zur Veröffentlichung. Für unverlangt eingesandte Manuskripte und Programme kann keine Haftung übernommen werden.

Sämtliche Veröffentlichungen in **INPUT 64** erfolgen ohne Berücksichtigung eines eventuellen Patentschutzes. Warennamen werden ohne Gewährleistung einer freien Verwendung benutzt.

Printed in Germany

© Copyright 1985 by Verlag Heinz Heise GmbH

ISSN 0177-3771

Titelidee: **INPUT 64**

Titelfoto: Artreferenz

Titelmusik: **INPUT 64**

Programmierung: H. Gehrman

INPUT 64-Abonnement

Abruf-Coupon

Ja, übersenden Sie mir bis auf Widerruf alle künftigen INPUT 64-Ausgaben ab Monat
(Kündigung ist jederzeit mit Wirkung ab der jeweils übernächsten Ausgabe möglich. Überzahlte Abonnementgebühren 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/Nichtanzuführendes streichen.)

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (A = ae, O = oe, B = ue)

Vorname/Zuname

Beruf/Funktion

Straße/Nr.

Pl. Z.

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.

hier abtrennen

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

Bankleitzahl

Konto-Nr.

Geldinstitut

Ort des Geldinstituts

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



INPUT 64

Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

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



Heise