

500 Mark

für formatierte Eingabe

Schreiben Sie ein Programm, dann besteht es in der Regel aus drei Teilen: Eingabe, Verarbeitung und Ausgabe von Daten. Es sind also diese Teile, die immer wieder programmiert werden müssen. Und deshalb hat dieses Mal ein Unterprogramm den Wettbewerb gewonnen, das eine wichtige und universelle Eingabe-Routine zur Verfügung stellt.

Jeder Programmierer steht bei einem neuen Programm vor der gleichen Frage: Wie soll meine Eingabe aussehen? Begnügt man sich mit dem vorhandenen Wortschatz des Commodore-Basic, ist es schwierig, eine sichere Eingabe zu erhalten. Andere Basic-Versionen haben entsprechende Befehle. Was also liegt näher, als sich eine Eingabe-Routine selbst zu programmieren, und zwar eine, die man immer wieder verwenden kann? Doch da beginnt auch schon das Problem. Es ist gar nicht so einfach, Standard-Unterprogramme zu entwickeln. So ein Programm muß folgende Forderungen erfüllen:

1. Es muß fehlerfrei sein. Das heißt, ganz gleich, welche Taste bei einer Eingabe gedrückt wird, das Programm darf niemals mit einer Fehlermeldung (oder noch schlimmer: ohne Meldung) abstürzen.
2. Es muß flexibel sein. Man

muß das Unterprogramm in jedem möglichen Programm einsetzen können, unabhängig von der sonstigen Aufgabe. Die wichtigsten Eingaben sind: nur numerische, also alle Ziffern von 0 bis 9; für kaufmännische und ähnliche Probleme 'muß wahlweise zusätzlich noch die Eingabe eines Dezimalpunktes möglich sein; und Text, also alle Buchstaben, Sonderzeichen und Ziffern. Oft wird eine bestimmte Eingabelänge gefordert und darf nicht überschritten werden, auch das ist wichtig. Und damit die einzugebenden Werte auch an der richtigen Stelle stehen, muß der Cursor positioniert werden können.

Das folgende Unterprogramm erfüllt diese Forderungen und läßt sich in jedem Basic-Programm einsetzen. Die Bedienung des Programms können Sie den REM-Zeilen des Listings entnehmen.

(Rolf Hilchner/gk)

```

1 REM"
2 REM" | UP - FORMATIERTE EINGABE |
3 REM" |
4 REM" |
5 REM" | Q1 = ZEILE DER EINGABE |
6 REM" | Q2 = SPALTE DER EINGABE |
7 REM" | Q3 = MAX. LAENGE D. EING. |
8 REM" | Q4 = 1 ← LEEREINGABE NICHT |
9 REM" | ERLAUBT |
10 REM" | Q5 = 1 ← NUR NUMERISCHE |
11 REM" | EINGABE ERLAUBT |
12 REM" | Q6 = 1 ← BEI NUMERISCHER |
13 REM" | EINGABE IST EIN |
14 REM" | PUNKT ERLAUBT |
15 REM" |
16 REM" | Y3$ = MUSS EINMAL AM ANFANG |
17 REM" | DES HAUPTPROGRAMMES |
18 REM" | DEFINIERT WERDEN. EIN |
19 REM" | PUNKTESTRING, DER DIE |
20 REM" | LAENGE DER LAENGSTEN |
21 REM" | EINGABE IM HAUPTPROG. |
22 REM" | HAT. ZUSAETZLICH MUSS |
23 REM" | AM ENDE DES STRINGS |
24 REM" | DAS ZEICHEN '←' |
25 REM" | STEHEN. BEISPIEL : |
26 REM" | Y3$=' .....←' |
27 REM" |
28 REM" | Y4$ = WIE 'Y3$', JEDOCH |
29 REM" | WERDEN STATT PUNKTE |
30 REM" | SPACES GESETZT. DAS |
31 REM" | ZEICHEN '+ ' ENTFAELLT |
32 REM" | BEISPIEL : |
33 REM" | Y4$=' ' ' |
34 REM" |
35 REM" |
36 REM" | II = LAUFVARIABLE |
37 REM" | JJ = LAUFVARIABLE |
38 REM" | Q7 = ZAEHLER, DER ANGIBT, |

```

```

39 REM" | AN WELCHER STELLE EIN |
40 REM" | PUNKT GESETZT WURDE |
41 REM" |
42 REM" | Y1$ = INHALT DER EINGABE |
43 REM" | WIRD VOM UNTERPROG. |
44 REM" | AN DAS HAUPTPROG. |
45 REM" | UEBERGEHEN. |
46 REM" |
47 REM" | Y2$ = ENTHAELT DAS JEWEILS |
48 REM" | EINGEGEBENE ZEICHEN |
49 REM" |
50 REM" |
51 REM" | FORMATIERTE EINGABE, BEI DER |
52 REM" | JEDES EINGEGEBENE ZEICHEN |
53 REM" | UEBERPRUEFT UND FALLS NOETIG |
54 REM" | ZURUECKGEWIESEN WERDEN KANN. |
55 REM" | DIE ANGEGEBENEN PARAMETER |
56 REM" | MUESSEN ZUM TEIL (Q1,Q2,Q3, |
57 REM" | Y3$,Y4$) EINGEGEBEN WERDEN, |
58 REM" | ODER KOENNEN BEI BEDARF |
59 REM" | UEBERGEHEN WERDEN (Q4,Q5,Q6) |
60 REM" | VOR DEM RUECKSPRUNG INS |
61 REM" | HAUPTPROGRAMM WERDEN ALLE |
62 REM" | PARAMETER AUF NULL GESETZT. |
63 REM" | DAS ERGEBNIS DER EINGABE- |
64 REM" | ROUTINE, DIE EINGABE WIRD IN |
65 REM" | DER VARIABLEN 'Y1$' AN DAS |
66 REM" | HAUPTPROGRAMM UEBERGEHEN. |
67 REM" |
68 REM" |
69 REM" |
70 REM" |
100 REM" | ***** |
102 REM" | *** FORMATIERTE EINGABE *** |
104 REM" | *** VON *** |
106 REM" | *** ROLF HILCHNER *** |
108 REM" | *** RHEYDTER STR.48 *** |
109 REM" | *** 4040 NEUSS 1 *** |

```

```

110 REM *****
120 POKE 650,255:Y1$="":Q2=Q2-1:POKE 214,Q1
:POKE 211,Q2:PRINT" (UP)T";RIGHT$(Y3$,Q3)
<071>
130 FOR II=1 TO Q3+1 <123>
140 GET Y2$:IF Y2$=""THEN 140 <130>
150 IF ASC(Y2$)=20 AND II>1 THEN Y1$=LEFT$(Y1$,
LEN(Y1$)-1):II=II-2:GOTO 250 <253>
160 IF ASC(Y2$)=13 AND II=1 AND Q4=1 THEN GOSUB
280:GOTO 140 <104>
170 IF ASC(Y2$)<>13 AND II=Q3+1 THEN GOSUB 280
:GOTO 140 <059>
180 IF ASC(Y2$)=13 GOTO 260 <060>
190 IF ASC(Y2$)<32 OR ASC(Y2$)>93 THEN GOSUB 280
:GOTO 140 <224>
200 IF Q5=1 AND ASC(Y2$)=45 AND II=1 GOTO 240
<143>
210 IF II<=Q7 THEN Q7=0:Q6=1 <145>
220 IF Q5=1 AND ASC(Y2$)=46 AND Q6=1 THEN Q6=0:
:Q7=II:GOTO 240 <035>
230 IF Q5=1 AND ASC(Y2$)<48 OR Q5=1 AND ASC(Y2$
)>57 THEN GOSUB 280:GOTO 140 <063>
240 Y1$=Y1$+Y2$ <087>
250 POKE 214,Q1:POKE 211,Q2:PRINT" (UP)";
LEFT$(Y1$+"T"+Y3$,Q3)+"T";:NEXT II <004>
260 POKE 214,Q1:POKE 211,Q2:PRINT" (UP)";
Y1$+LEFT$(Y4$,Q3-LEN(Y1$)+1) <242>
270 Q1=0:Q2=0:Q3=0:Q4=0:Q5=0:Q6=0:Q7=0
:POKE 650,0:RETURN:'<--- AUSGANG AUS UP <117>
280 POKE 54296,15:POKE 54277,6:POKE 54278,0
:POKE 54275,8:POKE 54274,0 <046>
290 POKE 54273,92:POKE 54272,237:POKE 54276,65
:FOR JJ=1 TO 150:NEXT JJ:POKE 54276,0 <084>
300 RETURN:'<--- AUSGANG AUS PIEP-UP <061>
310 REM"
320 REM" BEISPIEL 1 FUER DEN AUFRUF :

```

```

330 REM"
340 REM" 10 Y3$=' .....<'':Y4$=' '
350 REM"
360 REM" 20 PRINT' '':POKE214,5:POKE
370 REM" 211,15:PRINT' 'NAME : '
380 REM" 30 Q1=5:Q2=23:Q3=10:GOSUB 100
390 REM" 40 PRINT:PRINT' 'SIE HEISSEN'
400 REM" ;Y1$:END
410 REM"
420 REM" BEISPIEL 2 FUER DEN AUFRUF :
430 REM"
440 REM" 10 Y3$=' .....<'':Y4$=' '
460 REM" 20 PRINT' '':POKE214,5:POKE
470 REM" 211,15:PRINT' 'ALTER : '
480 REM" 30 Q1=5:Q2=24:Q3=10:Q4=1:
490 REM" Q5=1:GOSUB 100
500 REM" 40 PRINT:PRINT' 'SIE SIND' ;
510 REM" VAL(Y1$);' 'JAHRE ALT'':END
520 REM"
530 REM" HINWEIS :
540 REM"
550 REM" POKE 214,Y BEWIRKT, DASS DER
560 REM" CURSOR IN DIE ZEILE Y SPRINGT
570 REM"
580 REM" POKE 211,X BEWIRKT, DASS DER
590 REM" CURSOR AN D. SPALTE X SPRINGT
600 REM"

```

READY.

Formatierte Eingabe

## Einzeiler-Wettbewerb: Die nächsten 14

Wieder haben wir die interessantesten Einzeiler für Sie herausgesucht. Darunter sind sowohl nützliche als auch witzige oder lehrreiche. Sogar ein Adventure ist darunter. Alle Veröffentlichungen werden mit 50 Mark und einer Diskette mit allen Programmen dieser Ausgabe belohnt.

Wenn man einigen Einsendern Glauben schenken will — und es besteht kein Grund, das nicht zu tun —, so sind ihre Einzeiler keineswegs schnell dahingeschriebene Programme, sondern oft das Ergebnis von tage-, ja wochenlangen Experimenten und Versuchen. Bei manchen anderen allerdings ist die programmiertechnische Umsetzung einer Idee nicht das Entscheidungskriterium gewesen, sondern vielmehr die Idee selbst. Sie werden auch sehen, daß bei einem Einzeiler weder das Programm noch die Idee besonders originell ist. Aber seine Programmbeschreibung ist so überzeugend, daß sie Ihnen nicht vorenthalten sein soll. Doch nun viel Spaß und hoffentlich einige »Aha«-Erlebnisse.

### Umwandlung beliebiger Zahlensysteme (VC 20/C 64)

Die beiden Einzeiler dienen zum Umrechnen zwischen Dezimalzahlen und Zahlen beliebiger Basis. Kombiniert bilden beide eine Umwandlung zwischen verschiedensten Zahlensystemen. Beide Zeilen können sowohl als Unterroutine (mit RETURN) wie auch als Teil eines größeren Programms stehen.

#### a) Umwandlung dezimal/beliebig

Die Routine wandelt eine Dezimalzahl beliebiger Größe in der Variablen D in eine Zahl der Basis um, die in der Variablen B angegeben ist. Das Ergebnis steht in Z\$. Zuerst wird

Z\$ gelöscht. Dann wird eine Dummy-Schleife eröffnet, die nur einen einzigen Durchlauf zu haben scheint (0 bis 0). Der Befehl wird dazu genutzt, später wieder mitten in die Zeile einspringen zu können. Bei jeder Stelle wird D durch die Basis B geteilt und dadurch die unterste Ziffer abgeschnitten. Die jeweils niederwertigste Stelle ist der ganzzahlige Rest dieser Division und steht in S. S wird nun in den ASCII-Code umgerechnet, indem zur Zahl S 45 addiert wird. Ist S eine Ziffer von 0 bis 9, so nimmt der Term (S < 10) den Wert -1 an und es wird 7 subtrahiert (siehe ASCII-Tabellen). Der Code wird durch den CHR\$-Befehl in einen String gewandelt und vorne an Z\$ angehängt. Die letzte (höchstwertige) Ziffer ist erreicht, wenn D < 1 ist (nächste Stelle = 0). Die Schleifenvariable P wird auf -D gesetzt. Beim darauffolgenden NEXT-Kommando wird P um den STEP (hier 1) erhöht und mit dem Endwert des FOR-Befehls (hier 0) verglichen. Wenn D noch >= 1 ist, ist P+1 <= 0, die Endbedingung ist noch nicht erreicht und es wird zu dem Statement nach dem FOR-Befehl gesprungen.

#### b) Umwandlung beliebig/dezimal

Diese Routine wandelt eine beliebige große Zahl der von der Variablen B angegebenen Basis in eine Dezimalzahl. Die zu wandelnde Zahl muß in Z\$ stehen; die Variable D enthält das Ergebnis. Am Anfang wird die Variable D auf Null gesetzt. Die Schleifenvariable S des darauffolgenden FOR-Befehls dient als Zeiger auf die einzelnen Stellen von Z\$. Diese werden nun in den ASCII-Code gewandelt, der Code für Null (48) wird subtrahiert und das Ergebnis in H zwischengespeichert. Die schon umgewandelten Stellen (in D) werden zunächst durch Multiplikation mit B um eine Potenz dieser Basis erhöht, um dann die aktuelle Stelle (H) zu addieren. Ist