



Als erstes fiel mir auf, daß eine Dokumentation von Simons-Basic-Programmen nicht möglich war. Hier die vorläufige Lösung: Folgende Zeilen müssen eingefügt werden:

```
495 if x=100 then 330
805 if x=34 then y=2
806 if x=100 then y=3
835 if y=3 and (x=49 or x=50 or x=51) then y=0
836 if y=3 goto 750
Folgende Zeilen müssen geändert werden:
770 if x=34 or x=100 or x=131 or x=143 then z=0:
```

```
    goto 800
820 if x=34 and y=2 goto 750
920 ...:va$(i)=v1$+" :"
```

Des weiteren empfand ich es als einigermaßen lästig, bei vielen Variablen ständig die Frage »Wollen Sie Bemerkungen eingeben?« beantworten zu müssen. Nach folgenden Änderungen kann man die Variablen beliebig durchsuchen und mit Text versehen, um danach durch Eingabe von »E« wie Ende zur Ausgabe zu gelangen. Außerdem kann man nun bei Variablen, die sich über mehrere Druckzeilen erstrecken, seine Bemerkungen auf alle zugehörigen Zeilen verteilen. Folgende Zeilen sind einzufügen:

```
1105 if a$ ="e" goto 1180
1135 print "{home}{9down}"le$
Folgende Zeilen müssen geändert werden:
1080 ...a$ < > chr$(13) and a$ < > "e" goto 1080
1020 open 1,0
1160 ...: goto 1070
```

Die folgenden Zeilen sind zu löschen:

```
1140
1150
1170
```

Bei der Ausgabe ohne Zeilennummer genügt es im allgemeinen, wenn eine Variable, auch wenn sie sich über mehrere Zeilen erstreckt, nur einmal ausgegeben wird. Die folgende Lösung berücksichtigt dabei die Bemerkungen, die sich über mehrere Zeilen erstrecken. Es sind folgende Zeilen zu ändern:

```
1272 ...:goto 1275
1273 if left$(va$(i+1),4)=left$(va$(i),4) then 1275
1274 print #1,left$(va$(i), 8)
1275 next i
```

Die Abfrage »Nochmal/Weiter« läßt sich bei einer Ausgabe auf den Drucker durch Einfügen folgender Zeilen vermeiden:

```
685 if dv=4 then 712
1245 if dv=4 then 1269
```

Meine letzte Anmerkung betrifft die leidige Garbage-Collection (siehe hierzu auch den Kurs »Effektives Programmieren«, 64'er, Ausgabe 1/85), die der Computer durchführt, wenn sein Stringspeicher voll ist, und er weiteren Platz braucht. Bei meinen längeren Programmen lag der Computer spätestens bei der Eingabe von Bemerkungen für längere Zeit stumm und schweigend vor mir. Folgende Änderungen können hier Verbesserung schaffen:

1) Die Variable OB in Zeile 150 verkleinern. Sie gibt die Größe der Stringfelder an. Die Dauer der Garbage-Collection hängt aber in etwa vom Quadrat der Variablenzahl ab. Ich halte eine Angabe von OB=200 im Normalfall für völlig ausreichend. Ansonsten erscheint während des Programmlaufs eine entsprechende Fehlermeldung.

2) Das Ersetzen der speicherplatzverschwendenden Sortierprozedur durch die SWAP-Routine aus der 64'er, Ausgabe 1/85, Seite 123. Hierzu

```
1) Zeile 550 ersetzen durch:
550 sys 6000(ps$(i), ps$(j))
2) Zeile 1000 ersetzen durch:
1000 sys 6000(va$(i),va$(j))
3) Eingabe im Direktmodus:
```

```
print (peek(45)+256*peek(46))
Den erhaltenen Wert aufschreiben!
```

4) In den Zeilen 550 und 1000 den Wert 6000 durch den aufgeschriebenen Wert ersetzen.

5) Das Programm mit SAVE"XREF1.0",8 abspeichern.

6) Den Basic-Lader der SWAP-Routine laden und starten.

7) Auf die Frage nach der Startadresse den aufgeschriebenen Wert eingeben.

8) Das Dokumentationsprogramm mit LOAD"XREF1.0",8 laden.

9) Im Direktmodus eingeben (für »W« dabei den aufgeschriebenen Wert einsetzen):

```
POKE 45,(W+50)AND255:POKE 46,(W+50)/256
POKE174,(W+50)AND255:POKE175,(W+50)/256
```

Das Programm kann nun mit SAVE"XREF2.0",8 abgespeichert werden.

Achtung! Es dürfen keine Veränderungen mehr vorgenommen werden, die die Länge des Programms beeinflussen, da das Maschinenprogramm direkt hinter dem Basic-Programm steht! Das Zwischenprodukt "XREF1.0" kann jetzt gelöscht werden.

(Lutz Kleinostendarp/tr)

## Anpassung von Cross-Ref auf Datasette:

Bedingt durch ein anderes Speicherformat auf Datasette gegenüber der Floppy ist Cross-Ref mit den in 64'er, Ausgabe 6/86 genannten Änderungen für Kassettenbetrieb trotzdem nicht lauffähig. Auch die Abfrage der Status-Variablen ist nicht korrekt.

Um Cross-Ref trotzdem auch für unsere Datasettenbesitzer zugänglich zu machen, ist folgendes zu beachten:

Das zu bearbeitende Programm muß in einem »Pseudo-Floppy-Format« gespeichert werden. Dazu das Programm laden und mit folgenden Befehlen im Direktmodus (also ohne Zeilennummern) neu aufs Band schreiben.

```
OPEN 1,1,"name" [RETURN]
PRINT #1,CHR$(PEEK(43));CHR$(PEEK(44)) [RETURN]
FOR I=PEEK(43)+256*PEEK(44) TO PEEK(45)+256*PEEK(46):PRINT #1,CHR$(PEEK(I));NEXT [RETURN]
CLOSE 1 [RETURN]
```

Das File hat zwar jetzt das richtige Format, aber die Statusabfrage in Zeile 90 ergibt immer noch falsche Werte. Auf Diskette zeigt ein ST=64 nämlich das Ende eines Files an, auf Kassette dagegen bedeutet ST=64, daß ein CHR\$(0) folgt! Die folgenden Änderungen berücksichtigen dies:

```
90 GET #1,A$ : X=ASC(A$+N$) : IF X < > 0 THEN
XX=0 : RETURN
95 XX=XX+1 : IF XX<3 THEN RETURN
290 OPEN 1,1,0,NA$ : GOSUB 120 : IF X=2049 THEN
310
```

```
730 OPEN 1,1,0,NA$ : GOSUB 120
```

Die Zeilen 1290 bis 1310 entfallen.

(Stefan Becker/tr)

Anmerkung der Redaktion:

Auf der Leserservice-Diskette dieser Ausgabe befindet sich »Cross-Ref« in der verbesserten Version von Gerhard Löber und die Simons Basic-Version von Lutz Kleinostendarp. Auch ist die Löber-Version zusätzlich kompiliert worden. Diese Version können wir allen Lesern wegen der gewaltigen Geschwindigkeitsvorteile nur empfehlen.