

PSG – Der Prüfsummengenerator

464

664

6128

Dieses Programm soll allen dabei helfen, die Listings des Schneider Magazins möglichst ohne Tippfehler in ihren CPC einzugeben. Vor allem bei langen Listings tauchen oft durch Ermüdung der Augen Fehler auf, die nur schwer auffindig zu machen sind. Wie schnell hat man da ein I mit einer 1, eine 0 mit einem O oder eine 8 mit einem B verwechselt.

Da die Betriebssysteme des 664 und 6128 von dem des 464 abweichen, waren zwei Versionen des PSG notwendig. Dabei erhielt der Prüfsummengenerator für den 464 auch gleich noch eine erweiterte AUTO-Funktion. Sie ist auch ohne PSG einsetzbar. Der normale AUTO-Befehl des 464 gibt, wenn er auf eine bereits vorhandene Zeilennummer stößt, nur ein Sternchen aus. Im Basic 1.1 der 6er Reihe ist dies besser gelöst. Hier wird die vollständige Zeile angezeigt, und man kann alles, einschließlich der Zeilennummer, verändern. Die abgedruckte AUTO-Funktion gibt jetzt auch die Zeile aus. Allerdings ist nur ihr Text editierbar. Zum Löschen einer Zeile muß dies zunächst mit CLR oder DEL für alle Zeichen durchgeführt werden. Mit SPACE, gefolgt von ENTER, läßt sich die Zeile aus dem Listing entfernen. Dies gilt für alle drei CPCs. Der PSG ist natürlich auch bei der Direkteingabe (mit Zeilennummer) und dem EDIT-Kommando aktiv.

Damit der Prüfsummengenerator einwandfrei funktioniert, ist es unbedingt erforderlich, daß das abzutippende Programm vor dem ersten Probelauf vollständig eingegeben und abgespeichert wird. Ansonsten ist es möglich, daß eventuell vorhandene Pokes oder nachgeladene MC-Routinen dieses Programms den PSG teilweise oder ganz zerstören. Im Zweifelsfall ist er neu zu installieren. Andererseits kann er für einen Probelauf mit CTRL + CLR abgeschaltet werden. Die Symboltabelle ist freigegeben. Nach SYMBOL AFTER 256 sollte HIMEM einen Wert von &A4FF (42239) ergeben. Unterhalb der Adresse &A500 können Sie also weiterhin machen, was Sie wollen. Mit SHIFT + CLR läßt sich der PSG nach einem Break wieder aktivieren.

Das hier abgedruckte Programm ist natürlich nur mit den Listings des Schneider Magazins kompatibel. In dieser und den folgenden Ausgaben enthalten sie vor den Zeilennummern eine hexadezimale Zahl, die sich aus der Quersumme aller Zeichen einer eingegebenen Zeile bildet, nachdem diese in das interne Format des Basic übersetzt wurde. Was eine hexadezimale Zahl ist, muß man zur korrekten Eingabe eines Programms nicht unbedingt wissen. Wichtig ist nur, daß diese Zahl nicht abgetippt wird! Zur besseren Kennzeichnung ist sie dazu in zwei doppelte eckige Klammern gesetzt, die nicht auf der Tastatur des CPC zu finden sind. Diese Zeichen sehen so « » und nicht etwa so << >> aus. Zusätzlich läßt sich dadurch der Beginn einer neuen Zeile jetzt leichter erkennen.

Jetzt können Sie zunächst einmal den für Ihr Gerät passenden PSG abtippen, zuerst den Starter und dann, nachdem Sie diesen abgespeichert haben, den MC-Generator. Besitzer eines 464 sollten diesen nicht hinter dem Starter, sondern auf einer anderen Cassette ablegen. Dann wird

der MC-Generator gestartet. Er schreibt die Daten in das RAM und speichert sie dann automatisch auf einem Datenträger ab. Beim 464 mit Cassettenrecorder ist bei der Aufforderung "Press REC and PLAY then any key" wieder die Cassette mit dem Starter einzulegen, und zwar so, daß der MC hinter dem Starter abgelegt wird. Bei der Eingabe des erweiterten AUTO-Befehls ist genauso vorzugehen.

Die Listings sind weiterhin in einer Breite von 40 Zeichen abgedruckt. Das entspricht MODE 1 des CPC. Sollte in diesem Mode auf dem Bildschirm das äußerste rechte Zeichen einer Zeile (das 40.) nicht dem der Zeile des abgedruckten Listings entsprechen, steht bereits fest, daß irgendwo etwas vergessen oder zuviel eingegeben wurde. Man tippt also wie bisher eine Zeile ab. Mit einem kleinen Stück Pappe oder einem Lineal geht man nun Zeile für Zeile durch das Listing. Sobald eine vollständig eingegeben ist, drückt man ENTER (beim CPC 6128 heißt diese Taste RETURN), und die Prüfsumme wird in einem Kasten ausgegeben. Sollte sie nicht mit der im Schneider Magazin angegebenen übereinstimmen, muß die Zeile korrigiert werden. Geben Sie die nächste erst dann ein, wenn die momentan in Arbeit befindliche korrekt ist. Oft wurde ja nur ein Leerzeichen zuviel geschrieben, oder Sie haben in einem String einen Groß- mit einem Kleinbuchstaben verwechselt. Auch die Eingabe von ' für REM oder Klein- statt Großschreibung in einer REM-Zeile ergeben andere Werte.

Der PSG erkennt überflüssige Leerzeichen am Ende der Zeile und schneidet sie einfach ab. Gerade bei langen Listings führen solche überzähligen Spaces oft zu MEMORY FULL oder einer allzu häufigen Garbage Collection, da sie unnötig Speicherplatz belegen. Wenn man mit COPY eine Zeile dupliziert, saust der Copy-Cursor besonders gerne einmal vier oder fünf Stellen über das tatsächliche Zeilenende hinaus. Das sind dann gleich wieder ein paar Byte Speicherplatz weniger. Die erweiterte AUTO-Funktion für den 464 verfügt natürlich über dasselbe Feature.

Der PSG wurde ohne Prüfsumme abgedruckt, da ja noch niemand über ihn verfügt. Und solange er nicht fehlerfrei läuft, nützen auch die schönsten Prüfsummen nichts. Nehmen Sie also noch einmal Ihren ganzen Mut zusammen. Besonders lang sind die Listings zum Glück ja nicht. Der PSG ist zudem auf den Fingerschonend-Datenträgern ab dieser Ausgabe zu finden. Sollten Sie dennoch nicht zu recht kommen oder andere Probleme mit Ihrem CPC haben, so sei hier nochmals auf die User-Sprechstunde (dienstags und donnerstags, von 13 bis 16 Uhr) hingewiesen.

H. H. Fischer

PSG 464 Starter

```
10 LOAD"psg464.mc":CALL &A400:NEW
```

PSG 464 MC-Generator

```
1 : 'MC-Generator: psg464.ldr
2 :
3 : 'erzeugt : psg464.mc
4 :
5 : 'Copyright : hhf (CPC 464)
6 :-
```

```

100 DATA 3E10068DCD2DBB3E10068ECD33&0478
101 DATA BB068D0E0B2151A4CD0FBB068E&04A8
102 DATA 0E0B215CA4CD0FBB3E01CD0EBC&04A7
103 DATA 2167A4CDECA5CD7ABCCD8FBC21&07C6
104 DATA 00002294B2217FA4227BAE227D&0496
105 DATA AE2A96B02296B211F00002180A4&05CE
106 DATA C3ABBB43414C4C202641353030&0461
107 DATA 0D43414C4C2026413530430050&02B5
108 DATA 5320322E3020A4313938372068&0328
109 DATA 2E682E666973636865720D0A0A&03C9
110 DATA 414E203A203C5348946543E3C&033D
111 DATA 434C523E0D0A4155533A203C43&02F8
112 DATA 54524C3E3C434C523E0D0A0A00&02AC
113 DATA 000000000000000000000000&0000
114 DATA 000000000000000000000000&0000
115 DATA 000000000000000000000000&0000
116 DATA 000000000000000000000000&0000
117 DATA 000000000000000000000000&0000
118 DATA 000000000000000000000000&0000
119 DATA 000000000000000000000000&0139
120 DATA 00A597215FA6180AC9973200A5&04BB
121 DATA 3EC9216DA6320CA5113ABD0103&042A
122 DATA 00EDB0C3ECA5F3E32245A6E3FB&08B2
123 DATA C5D5E5972A45A6011CC1ED427C&06B4
124 DATA B53244A62019CDA3E7301401A4&054A
125 DATA AC110000232323237EB72805CD&0378
126 DATA 96E118F702E1D1C1CD6DA6CD20&07C8
127 DATA A6C8F5C5D5E521A4ACED5B1DC&0864
128 DATA 3A44A6B72811CD61DDB72827C&05F2
129 DATA 04EE30227EFE20200123CDBBDE&058A
130 DATA E5CD61DDB7E12811E56A160062&0688
131 DATA 1959195819D1CD0BA6CDD9A5E1&0677
132 DATA D1C1F1C9505320322E3020A431&0594
133 DATA 39383720682E682E6669736368&0401
134 DATA 6572206261686E686F66737472&0526
135 DATA 2E31302037353030206B61726C&0345
136 DATA 7372756865E52147A6CDECA5E1&0759
137 DATA 7CCDF5A57DCDF5A52153A67EB7&0816
138 DATA 8CD5ABB2318F7F50F0F0F0FCD&05DA
139 DATA FEA5F1E60FFE0A3802C607C630&068E
140 DATA C35ABB5C0100001A134F09E32B&0431
141 DATA 7CB5E320F5C101040009C9F5C&067B
142 DATA D5E521A4AC06007EB728040423&04B9
143 DATA 18F878B7280A2B7EFE20200436&0492
144 DATA 0010F6E1D1C1F1C9FF00000D0A&0649
145 DATA 969A9A9A9A9C0D0A9500950D0A&04F2
146 DATA 939A9A9A9A990D0A00C324A504&053B
147 DATA 01505320616E0D0A0A00CF98AA&03C5
148 DATA 04015053206175730D0A0A00&0232
149 DATA EOF
150 :
151 MEMORY &A3FF
152 z= 100:0= 1:a=&A400
153 READ d$:IF d$="EOF" GOTO 163
154 PRINT "Zeile:" z ;
155 FOR i=1 TO INSTR(d$,"&")-2 STEP 2
156   b%=VAL("&"MIDS(d$,i,2))
157   POKE a,b%:s=s+b%:a=a+1
158 NEXT
159 cs=VAL(MIDS(d$,i))
160 PRINT CHR$(1)CHR$(6-(168*(cs<>s)))
161 s=0:z=z+o
162 GOTO 153
163 SAVE "psg464.mc",b,&A400,&27C
164 END

```

PSG 664/6128 Starter

10 LOAD "psg664.mc":CALL &A400:NEW

PSG 664/6128 MC-Generator

```

1 : 'MC-Generator: PSG664.ldr
2 :
3 :
4 : 'erzeuge : PSG664.mc
5 :
6 :
7 : 'Copyright : hhf (CPC 464)
8 :
9 :
100 DATA F33A4FBBF70281E215EBD221A&0563
101 DATA AS214ADDE2233A521CFEE2239A5&05C9
102 DATA 21A4DF2244A5214DDE2248A5FB&0605
103 DATA 3E10068DCD2DBB3E10068ECD33&0478
104 DATA BB068D0E0B2172A4CD0FBB068E&04C9
105 DATA 0E0B217DA4CD0FBB3E01CD0EBC&04C8
106 DATA 2188A4CDACA5CD7ABCCD8FBC21&07A7
107 DATA 00002234B7217FA4225EAE2260&0401
108 DATA AE11F0002180A4C3ABBB43414C&05ED
109 DATA 4C2026413530300D43414C4C20&02B1
110 DATA 26413530430D505320322E3020&028F
111 DATA A43139383720682E682E666973&040B
112 DATA 636865720D0A0A414E203A203C&0308
113 DATA 53484946543E3C434C523E0D0A&032E
114 DATA 4155533A203C4354524C3E3C43&0371
115 DATA 4C523E0D0A0A000000000000&00FD
116 DATA 000000000000000000000000&0000
117 DATA 000000000000000000000000&0000
118 DATA 000000000000000000000000&0000
119 DATA 00000000000000000000003EC9&32&0139
120 DATA 00A597211CA6180AC9973200A5&0478
121 DATA 3EC9212BA6320CA5115BBD0103&0409
122 DATA 00EDB0C3ACA5CD2BA6CDE0A5C8&0869
123 DATA F5C5D5E5218AACCD52DEB72827&07CE
124 DATA CDD4EE30227EFE20200123CDA9&0637
125 DATA DFE5C52DEB7E12811E56A1600&06F7
126 DATA 621959195819D1CDCBA5CD99A5&0677
127 DATA E1D1C1F1C9505320322E3020A4&0644
128 DATA 3139383720682E682E66697363&03CA
129 DATA 866572206261686E686F667374&051C
130 DATA 722E31302037353030206B6172&034B
131 DATA 6C7372756865E52104A6CDACA5&0661
132 DATA E17CCDB5A57DCDB5A52110A67E&077D
133 DATA B7C8CD5ABB2318F7F50F0F0F0&05C4
134 DATA CDBEA5F1E60FFE0A3802C607C6&06EB
135 DATA 30C35ABB5C0100001A134F09E3&0436
136 DATA 2B7CB5E320F5C101040009C9F5&05E1
137 DATA C5D5E5218AAC06007EB7280404&0541
138 DATA 2318F878B7280A2B7EFE202004&047F
139 DATA 360010F6E1D1C1F1C9D0D0A969A&06B0
140 DATA 9A9A9A9C0D0A9500950D0A939A&04E8
141 DATA 9A9A9A990D0A00C324A5040150&045F
142 DATA 5320322E3020616E00CF02AC04&0373
143 DATA 01505320322E302061757300&02BD
144 DATA EOF
145 :
146 MEMORY &A3FF
147 zeile= 100:schritt= 1:adr=&A400
148 READ byte$:IF byte$="EOF" GOTO 162
149 PRINT "Zeile:" zeile ;
150 l=INSTR(byte$,"&")-2
151 FOR i=1 TO l STEP 2
152   b%=MIDS(byte$,i,2)
153   POKE adr,VAL("&"&b$)

```

```

154      sum = sum + PEEK(adr)
155      adr = adr + 1
156 NEXT
157 checksum = VAL(MID$(byte$,i))
158 IF sum=checksum THEN v=6 ELSE v=174
159 PRINT CHR$(1)CHR$(v)
160 sum=0:zeile=zeile+schritt
161 GOTO 148
162 SAVE"PSG664.mc",b,&A400,&23B
163 END

```

AUTO 464 Starter

```
<0654> 10 LOAD"XAUTO.mc":CALL &A500:NEW
```

AUTO 464 MC-Generator

```

<08E3> 1  : 'MC-Generator: XAUTO.ldr
<004B> 2  :
<07B4> 3  : 'erzeugt      : XAUTO.mc
<004D> 4  :
<08DA> 5  : 'Copyright   : hhf (CPC 464)
<004F> 6  :
<0814> 100 DATA 3E10068DCD2DBB3E10068ECD33&0478
<082B> 101 DATA BB068D0E0B2151A5CD0FBB068E&04A9
<084E> 102 DATA 0E0B215CA5CD0FBB3E10CD0EBC&04A8
<0860> 103 DATA 2167A5CD6CA6CD7ABCCD8FBC21&0748
<07D1> 104 DATA 00002294B2217FA5227BAE227D&0497
<07E9> 105 DATA AE2A96B02296B211F0002180A5&05CF
<07C3> 106 DATA C3ABBB43414C4C202641363030&0462
<07BD> 107 DATA 0D43414C4C2026413630430D58&02BE
<07A4> 108 DATA 4155544F20312E3020A4313938&034E
<07BC> 109 DATA 3720682E682E66697363686572&0467
<07BE> 110 DATA 0D0A0A414E203A203C53484946&0290
<07D7> 111 DATA 543E3C434C523E0D0A4155533A&0327
<07DA> 112 DATA 203C4354524C3E3C434C523E0D&0337
<07CD> 113 DATA 0A0A006261686E686F66737472&0443
<0791> 114 DATA 2E31302037353030206B61726C&0345
<074E> 115 DATA 737275686500000000000000&0227
<070C> 116 DATA 000000000000000000000000&0000
<070D> 117 DATA 000000000000000000000000&0000
<070E> 118 DATA 000000000000000000000000&0000
<0755> 119 DATA 00000000000000000000003EC932&0139
<07F1> 120 DATA 00A697219BA6180AC9973200A6&04F9
<080D> 121 DATA 3EC921ACA6320CA6113ABD0103&046A
<0848> 122 DATA 00EDB0C36CA6F3E32299A63FB&0887
<082B> 123 DATA C5D5E5972A99A6011CC1ED427C&0708
<07FD> 124 DATA B52019CDA3E7301401A4AC1100&04EB
<07DB> 125 DATA 00232323237EB72805CD96E118&04AA
<082D> 126 DATA F702E1D1C1CDACA61819584155&06AA
<07D7> 127 DATA 544F312E30A431393837686866&03E5
<086F> 128 DATA 7EB7C8CD5ABB2318F7F5C5D5E5&0885
<07E8> 129 DATA 21A4AC06007EB72804042318F8&040F
<07D8> 130 DATA 78B7280A2B7EFE202004360010&0392
<07FE> 131 DATA F6E1DC1F1C90000C324A60401&06B5
<07EB> 132 DATA 584155544F20616E00DA0A00&0CF60370
<07BC> 133 DATA 98AA0401584155544F20617573&0441
<03E5> 134 DATA 0D0A0A00&0021
<0217> 135 DATA EOF
<008F> 136 :
<031C> 137 MEMORY &A4FF
<07F5> 138 z = 100:0 = 1:a=&A500
<07BC> 139 READ d$:IF d$="EOF" GOTO 149
<0559> 140 PRINT"Zeile:" z ;
<0A09> 141 FOR i=1 TO INSTR(d$,"&")-2 STEP 2
<0912> 142   b%=VAL("&"+MID$(d$,i,2))
<0CB1> 143   POKE a,b%:s=s+b%:a=a+1
<0146> 144 NEXT
<0771> 145 cs=VAL(MID$(d$,i))
<0B00> 146 PRINT CHR$(1)CHR$(6-(168*(cs<s)))
<0798> 147 s=0:z=z+0
<0207> 148 GOTO 149
<075F> 149 SAVE"XAUTO.mc",b,&A500,&1BE
<0134> 150 END

```

Update für das Programm Girokontoführung aus Heft 9/87

464

664

6128

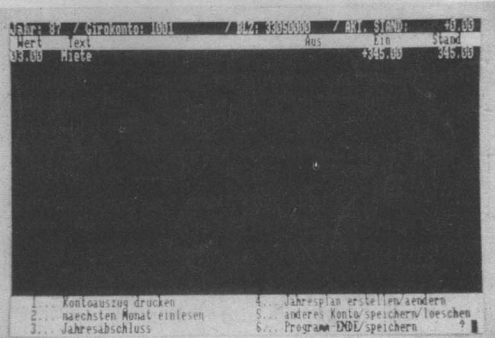
Beim Anpassen des Programms an den CPC 464 unterliefs uns leider ein kleiner Fehler. Abgedruckt wurde das Listing mit dem ersten (schwachen) Patch. Das GOSUB in Zeile 1300 ist natürlich nicht richtig.

In Zeile 1145 stand ON ERROR GOTO 3210. Damit wurde beim CPC 664/6128 das Nichtvorhandensein einer Datei abgefragt und dann nach "Datei neu einrichten J/N ?" gefragt. Das Problem war nur die Zeile 3210. Hier stand nämlich folgendes:

```
3210 CLS#5: IF ERR <> 32 AND DERR <> 146 THEN
PRINT#5, CHR$(7); "Fehler Nr.:"; ERR;
" in Zeile ",ERL;: CLS: END
```

Das funktioniert wegen DERR natürlich nur auf der 6er Serie. Beim CPC 464 kommt grundsätzlich ein Programmabbruch, wenn die Datei nicht gefunden wird. Außerdem läßt sich die Zeile, in der DERR steht, nicht mehr listen. Also, ganz einfach ein GOSUB eingebaut und vor dem Eröffnen die obligate Frage stellen. Leider wurde das RETURN nicht mit abgespeichert und somit auch nicht gedruckt. Im zugehörigen Kasten finden Sie nun die richtige Form.

H.H. Fischer



Update für GIRO.GO

```

<079E> 1290 REM /// GIROKONTO EINLESEN ///
<00E1> 1300 CLS#5
<0B92> 1301 PRINT#5,CHR$(7);"Konto neu anlegen
? (J/N)"
<0913> 1302 d=UPPER$(INKEYS):IF d="" THEN 1302
<0485> 1303 IF d="N" GOTO 1310
<0531> 1304 IF d="J" GOTO 3270
<011F> 1305 GOTO 1300
<0698> 1310 OPENIN "KTO"+c1+".DAT"

```