

# SMU

## Der Maskengenerator

**Programmieren Sie gerne Datenverarbeitungsprogramme? Dann brauchen Sie diese Erweiterung. Mit ihr können Bildschirmmasken mit numerischen und alphanumerischen Feldern erstellt und anschließend in eigene Basic-Programme eingebunden werden.**

Das Programmpaket »SMU« (Screen Management Utility) dient zur Erstellung von Bildschirmmasken und deren einfache Handhabung durch ein erweitertes Basic.

Angelehnt an professionelle Bildschirm-Maskengeneratoren kann auf einfache Weise eine Bildschirmmaske erstellt und auf Diskette gespeichert werden. Wenn Sie jetzt denken, schon wieder ein Maskengenerator, dann haben Sie weit gefehlt. Dieser Maskengenerator erstellt nicht stur eine Maske durch Erzeugen von PRINT-Zeilen, wie es andere Maskengeneratoren machen, sondern es lassen sich natürlich zusammen mit der Maske numerische und alphanumerische Felder definieren. Mit der Basic-Erweiterung können die so erstellten Masken in eigene Programme eingebunden werden. Der Vorteil dieses Programms liegt auf der Hand. Es ist nicht mehr erforderlich, sich über die Programmierung von Eingabefeldern und Masken in selbstgeschriebenen Dateiverwaltungsprogrammen Gedanken zu machen. Die Programmierung übernimmt ab jetzt der Maskengenerator. Sie müssen sich nur noch überlegen, wo die Eingabefelder stehen und wie sie aussehen sollen. Das Erstellen der Masken ist denkbar einfach. Wird

der Maskengenerator »SMU« geladen und mit dem Basic-Befehl RUN gestartet, muß man sich zuerst entscheiden, ob eine neue Maske erstellt oder eine alte bearbeitet werden soll. Hat man sich für das Erstellen einer neuen Maske entschieden, wird der Bildschirm gelöscht und ein Cursor erscheint. Wie gewohnt, läßt sich dieser Cursor mit den entsprechenden Steuertasten auf dem Bildschirm bewegen. Überall dort, wo ein Eingabefeld definiert werden soll, dient das »!«-Zeichen als Platzhalter für alphanumerische und das »#«-Zeichen als Platzhalter für numerische Zeichen. Aber Vorsicht; diese beiden Zeichen dürfen im erklärenden Text nicht erscheinen. Um später die Länge des Eingabefeldes zu erkennen, empfiehlt es sich, den Anfang und das Ende mit einem beliebigen Zeichen zu markieren. Ist die Maske erstellt, kann sie durch zweimaliges Drücken der »F1«-Taste auf einem zuvor bestimmten Gerät gespeichert werden. Mit der Basic-Erweiterung »SMUBU« läßt sich die so erstellte und gespeicherte Maske in eigene Basic-Programme einbinden. Ein Demoprogramm (LIT.VERW.) zeigt, wie einfach und dennoch effizient die Handhabung solcher Masken sein kann. (Georg Krcal/ah)



Eine Maske wird definiert



So sieht die Maske in Aktion aus (Demoprogramm LIT.VERW.)

### Lebenslauf

Am 3.2.1965 in Wien geboren, kam ich erst während meiner Ausbildung zum Chemiker mit der Computerei (soweit man diesen Anfang so nennen kann) in Berührung. Wie alles einmal klein und zaghaft beginnt, so auch meine Programmierkünste. Diesen bescheidenen Anfang machte ein programmierbarer UPN-Taschenrechner mit 49 Programmierschritten. Allerdings lernte ich bald darauf Basic im Selbststudium, frei nach dem Motto: Probieren geht über studieren. Meine Basic-Kenntnisse konnte ich bei einem Bekannten an einem Commodore 8032 vertiefen. Kurze Zeit später entstand auf einem P2000-Computer ein Textverarbeitungsprogramm.

1983 entschloß ich mich dann zum Kauf eines C 64. Vom langsamen Basic wenig angetan, wurde fortan in Maschinensprache programmiert. Die Konfrontation mit bildschirmgesteuerten Programmen weckte in mir den Wunsch, ähnliche Möglichkeiten auf dem C 64 zu realisieren, zumal der bedienungsunfreundliche INPUT-Befehl des Basic-Interpreters häufig mehr Schaden anrichtet als er nutzt. Wer hat es schon gern, wenn seine mühsam generierten Bildschirm-Layouts durch ein versehentliches CLR zerstört werden oder der Interpreter nach einer vermeintlich richtigen Eingabe »extra ignored« meldet? Ein Maskengenerator und entsprechende Funktionen waren also notwendig. Aber wo beginnen? Schützenhilfe gab es kaum. Die meisten Anwender waren gerade noch dabei, die Grafikfähigkeiten zu erkunden und Soundeffekte zu programmieren. Auf Fragen, wie der Computer String-Arrays anlegt, wußte ich keine Antwort. Diese Unwissenheit war jedoch kein Hindernis, die 2,5 KByte große Befehlserweiterung (SMUBU), an der ich etwa ein halbes Jahr gearbeitet habe, zu entwickeln. Hoffentlich macht SMU, wie ich meinen Maskengenerator getauft habe, all jenen Freude, die mit dem C 64 nicht nur spielen, sondern ihn auch sinnvoll einsetzen wollen.

(Georg Krcal)

Fortsetzung von Seite 59

```

$)+1
5010 IF A$=MID$(A1$,A,LEN(A$)) THEN 5040
5020 NEXT A
5030 PRINT"DIESES JON IST NICHT ENTHALTEN"
:GOTO 440
5040 B9$=MID$(A2$,A+LEN(A$),1)
5050 IF B9$="" THEN B9$=" "
5060 IF ASC(B9$)<=161 OR ASC(B9$)>=191 THE
N B9=1:RETURN
5070 B9=ASC(B9$)-160
5080 RETURN
9000 REM *** MASCHINENPROGRAMM ***
10000 DATA 173,14,220,41,254,141,14,220,16
5,1,41,251,133,1,169,0,133,34,133
10010 DATA 36,169,216,133,35,169,8,133,37,
162,8,160,0,177,34,145,36,136,208
10020 DATA 249,230,35,230,37,202,208,240,1
65,1,9,4,133,1,173,24,208,41,240,9
10030 DATA 2,141,24,208,173,14,220,9,1,141
,14,220,96,255
10040 REM *** DATA KLEINE ZAHLEN ***
10050 DATA 0,0,126,102,102,126,102,126
10060 DATA 0,0,126,96,96,126,6,126
10070 DATA 0,0,6,14,102,6,6,6
10080 DATA 0,0,126,6,6,126,6,126
10090 DATA 0,0,102,102,102,126,6,6
10100 DATA 0,0,24,102,6,24,96,126
10110 DATA 0,0,126,96,96,126,102,126
10120 DATA 0,0,126,6,6,12,12,12
10130 DATA 0,0,126,102,102,126,6,126
10140 DATA 0,0,126,102,102,102,102,126
10990 REM *** MASCHINENPROGRAMM ***
11000 DATA 165,45,133,34,165,46,133,35,160
,3,177,34,133,36,200,177,34,133,37
11010 DATA 160,0,177,36,201,65,144,9,201,9
0,176,5,169,75,145,36,96,162,0,201
11020 DATA 171,208,4,234,234,162,49,201,17
9,208,2,162,50,201,177,208,2,162,51
11030 DATA 201,178,208,2,162,52,201,163,20
8,2,162,53,201,183,208,2,162,54,201
11040 DATA 184,208,2,162,55,201,162,208,2,
162,56,201,185,208,2,162,57
11045 DATA 201,175,208,2,162,48,224,0
11050 DATA 240,4,138,145,36,96,201,40,208,
2,162,65,201,41,208,2,162,66,201,91
11060 DATA 208,2,162,67,201,93,208,2,162,6
8,224,0,240,4,138,145,36,96,169,71
11070 DATA 145,36,96,0
31000 REM *** PERIODENSYSTHEM ***
32000 DATA"1",1,"HE",4,"LI",6,94,"BE",9,01
,"B",10,81,"C",12,"M",14,"D",16
32010 DATA"E",19,"NE",20,18,"NA",22,99,"HG
",24,31,"HL",26,98,"SI",28,09
32020 DATA"2",30,97,"S",32,06,"CL",35,45,"
BR",39,95,"L",39,1,"GA",40,08
32030 DATA"SC",44,96,"II",47,9,"V",50,94,"
GR",52,"HN",54,94,"EE",55,85
32040 DATA"OD",58,93,"NI",58,71,"OU",63,54
,"ZN",65,37,"BA",69,72,"GE",72,59
32050 DATA"BS",74,92,"SE",78,96,"BR",79,01
,"JR",83,8,"RB",85,47,"SR",87,62
32060 DATA"Y",88,91,"ZR",91,22,"NB",92,91,
,"MD",95,94,"IC",97,"BU",101,07
32070 DATA"RH",102,9,"PD",106,4,"BS",107,8
7,"CD",112,41,"IN",114,82
32080 DATA"SN",118,69,"SB",121,75,"IE",127
,6,"I",126,9,"XE",131,3
32090 DATA"SS",132,9,"BA",137,33,"LA",138,
91,"GE",140,12,"PR",140,91
32100 DATA"ND",144,24,"PM",145,"SM",150,35
,"EU",151,96,"OD",157,25
32110 DATA"IB",158,92,"BY",162,5,"HD",164,
93,"ER",167,26,"IM",168,93
32120 DATA"YB",173,04,"LU",174,97,"HF",178
,94,"JA",180,95,"H",183,85
32130 DATA"KE",182,2,"GS",190,2,"IR",192,2
,"ET",195,05,"BU",196,97,"HG",200,59
32140 DATA"J",204,37,"PB",207,19,"BI",208
,99,"PD",209,"AT",210
32150 DATA"BN",222,"ER",223,"BA",226,"BC",
,227,"IH",232,04,"PA",231
32160 DATA"U",238,03

```

© 64'er

Listing zum »Chemie-Assistent« (Schluß)

# SMU - der Masken- generator

»Screen Management Utility« dient zur professionellen Erstellung von Bildschirmmasken unter Berücksichtigung von alphanumerischen und numerischen Feldern. Diese Masken lassen sich durch ein erweitertes Basic in eigene Programme einbinden.

SMU (Listing 1) ist für alle jene verwendbar, die ihren Basic-Programmen einen professionellen Anstrich geben möchten. SMU zielt auf die Verwaltung von Bildschirmmasken ab, wie sie auch bei kommerziellen Programmen verwendet werden. Mit SMU können Bildschirmmasken auf einfachem Wege erstellt und anschließend mit Basic-Befehlen verwaltet werden. SMU besteht aus drei Programteilen dem Maskengenerator »SMU« (Listing 1), einem Maschinenprogramm (Listing 2) und einer Befehlserweiterung »SMUBU« (Listing 3). Sind alle drei Programme abgetippt, wird der Maskengenerator mit LOAD "SMU",8 geladen und mit RUN gestartet. Die Maschinenprogramme werden automatisch nachgeladen. Zunächst einige Worte zu dem Prinzip einer Bildschirmmaske. Eine Bildschirmmaske ist aus Feldern und Informationsbegriffen aufgebaut. Wem das zu wenig anschaulich ist, der kann eine Bildschirmmaske (kurz Maske) mit einem unausgefüllten Formular vergleichen. Der Informationsbegriff (IB) wäre dabei der Teil, der dem Ausfüllenden erklärt, was auszufüllen ist; das Feld, der Freiraum hinter der Angabebezeichnung. Somit ergibt sich, daß die Informationsbegriffe Literale sind, die mit der Maske immer ausgegeben werden. Die Felder einer Maske vermögen nun Informationen aufzunehmen, die entweder vom Bediener oder vom Programm kommen. Im ersten Fall füllt der Bediener die Maske aus, die Feldinhalte werden anschließend Variablen zugewiesen, im zweiten Fall werden Variableninhalte in die Felder übertragen.

Die Art und Weise, wie ein Bediener solch eine Maske ausfüllt, läßt sich in zwei Gruppen unterteilen:

1. Die Zwangseingabe; der Benutzer muß die Felder so ausfüllen, wie das Programm es vorschreibt.
2. Die wahlfreie Eingabe; der Benutzer trifft selbst die Wahl, welches Feld er ausfüllt.

Bei SMU wurde eine Verknüpfung beider Methoden gewählt, nach einem ausgefüllten Feld springt der Cursor in das nächste, mit den Funktionstasten kann aber stets ein beliebiges Feld der Maske angesprungen werden, genauere Erklärungen dazu folgen noch.

Welchen Vorteil bieten nun Masken gegenüber den herkömmlichen Eingabeverfahren (wie zum Beispiel INPUT)? Der Hauptvorteil liegt in der Fehlerkorrektur, die Feldinhalte können vom Benutzer so lange verändert werden, bis alle Eingaben richtig sind. Ein weiterer Pluspunkt ist die feste Länge eines Feldinhaltes, ein alphanumerischer Feldinhalt wird der Stringvariablen mit einer fixen Länge übergeben. Letztlich

kann der Bediener keine Gestaltungsversuche am Bildschirm mehr machen, die die meisten CBM-Computer (so auch der C 64) ziemlich oft aus der Fassung bringen (redo from start, extra ignored).

## Der Maskengenerator

Beginnen wir mit der Beschreibung des Maskengenerators und mit der Erstellung einer Bildschirmmaske mit dem Programm SMU.

Wie bereits erklärt, besteht eine Maske aus Literalen und Feldern, für den Maskengenerator ist alles literal, was nicht Feld ist. Das mag vorerst nicht viel aussagen, aber nachdem festgelegt worden ist, was ein Feld ist, versteht man auch, wie man eine SMU-Maske zu erstellen hat. Der Maskengenerator unterscheidet zwei Feldarten:

1. Alphanumerische Felder
2. Numerische Felder

Diese Feldarten werden bei der Erstellung der Maske mit zwei eigens ausgewählten Zeichen gekennzeichnet. Dabei markiert das Rufzeichen (!) ein alphanumerisches Feld, das Channelzeichen (#) das numerische Feld. Überall dort, wo am Bildschirm diese Zeichen stehen, erkennt der Maskengenerator ein Feld. Eine Besonderheit dieses Maskengenerators ist, daß eine Folge mehrerer solcher Zeichen, als ein einziges Feld interpretiert wird. Zum Beispiel ist die Folge »!!!!« für den Maskengenerator ein einziges alphanumerisches Feld, mit einer Feldlänge von fünf Zeichen. SMU kann bis zu 255 Felder verwalten. Ein Feld darf eine maximale Länge von 255 Zeichen nicht überschreiten. Eine Verknüpfung von einem alphanumerischen Feld und einem numerischen Feld zu einem einzigen Feld ist nicht möglich. So ist die Felddefinitionsfolge »!!! # #« lediglich die unmittelbare Folge eines numerischen Feldes auf ein alphanumerisches Feld von drei Zeichen Länge. Eine SMU-Maske umfaßt einen Bildschirm von 23 Zeilen (0 bis 22), die restlichen zwei Zeilen sind während der Maskenerstellung nicht erreichbar und werden vom Maskengenerator auch nicht analysiert. Welche Wirkung haben nun diese Feldmarkierungen? Überall dort, wo bei der Maskenerstellung diese Feldbezeichner gestanden haben, werden später (siehe Maskenaufruf durch den SCREEN-Befehl) Leerzeichen ausgegeben. Hier kann der Benutzer seine Eingaben machen, den Cursor bewegen, Zeichen einfügen oder löschen. Numerische Felder erlauben nur die Eingabe von Zahlen (0 bis 9), Punkt (.), Plus (+), Minus (-) und dem Exponentenzeichen (e). Bei einem alphanumerischen Feld sind alle Zeichen (mit Ausnahme von Hochkomma) für die Eingabe zugelassen. Beachtenswert bei den numerischen Feldern ist noch, daß Zahlen, die eine Bereichsüberschreitung des Computers zur Folge haben würden, von SMU zurückgewiesen werden. Das heißt allerdings, daß ein numerisches Feld mit zum Beispiel 255 Zeichen nicht sinnvoll ist, der C64 kann solch eine große Zahl ja nicht mehr verwalten (unter der Voraussetzung, daß das Feld vollständig ausgefüllt wird).

## Die Basic-Erweiterung SMUBU

Zehn neue Befehle sind nach der Initialisierung der Erweiterung SMUBU (Listing 3) mit SYS 9\*4096 zugänglich. Vor jedem SMUBU-Befehl steht ein Rufzeichen (!).

Nachdem die Erweiterung mit SYS 9\*4096 gestartet wurde, stehen die Befehle SETIN, CLEIN, MASK, SCREEN, WMASK, IMASK, IOMASK, -, INLINE und DIER zur Verfügung.

### ! SETIN:

Dieser Befehl sperrt die Wirkung der Tasten STOP und

STOP/RESTORE. Das laufende Basic-Programm kann vom Benutzer nicht mehr unterbrochen werden.

### ! CLEIN:

Dieser Befehl hebt die Wirkung von SETIN wieder auf.

### ! MASK name,geräteadresse,stringarray(x):

Damit wird von der Diskette eine Maske geladen und die Daten für die Maske in das Feld ab dem String mit dem Index x gespeichert. Es bedeutet:

**name:** kann eine Stringvariable sein, die den Namen enthält, oder direkt der Name, eingefaßt in Hochkommata.

**geräteadresse:** ist die Geräteadresse der Floppy (8, 9 und so weiter).

**stringarray (x):** ist ein eindimensionales Feld. x ist der Index, ab dem die Maskendaten im Feld gespeichert werden sollten. Zum Beispiel: 10 DIM M\$(14)

```
20 !MASK "MENU",8,M$(0)
```

Es wird von dem Laufwerk mit der Geräteadresse 8 die Maske mit dem Namen »menu« geladen. Die Maskendaten befinden sich ab M\$(0) im Speicher. Die Daten einer Maske benötigen einen Platz von 15 Strings zu je 255 Zeichen, das heißt, daß die Maskendaten im obigen Beispiel in den Speichern M\$(0) bis einschließlich M\$(14) zu finden sind.

### ! SCREEN Stringarray(x):

Wie bereits angekündigt, wird durch diesen Befehl eine Maske aufgerufen. Dieser Befehl bringt eine Maske auf den Bildschirm. Gleichzeitig werden die Daten der Felder (Position, Länge und Feldart) SMUBU übermittelt, alle weiteren Befehle, die mit einer Maske operieren, greifen darauf zu. Dies ist selbst dann der Fall, wenn der Bildschirm durch CHR\$(142) gelöscht worden ist. SMUBU greift bei Maskenoperationen immer auf die Daten der mit SCREEN aufgerufenen Maske zurück.

**stringarray(x):** ist ein eindimensionales Stringarray. x ist der Index, ab dem die Daten einer SMU-Maske gespeichert sind. Zum Beispiel: 30 !SCREEN M\$(0)

Es würde die im ersten Beispiel geladene Maske auf den Bildschirm geschrieben werden.

Wichtig ist, daß durch x auch wirklich der Beginn der Maske indiziert wird, so hätte »!SCREEN M\$(1)« im Beispiel zu einer Fehlermeldung geführt (siehe SMUBU - Fehlermeldungen).

### ! WMASK var1,var2,var3,... :

Dieser Befehl ähnelt dem Basic-Befehl PRINT. Er schreibt allerdings die Inhalte der Variablen (String und Zahlenvariable) in die Felder der Maske. Zuvor werden die Felder der Maske gelöscht. Bei WMASK ist zu beachten:

1. Die Übergabe der Variableninhalte an die Felder erfolgt sequentiell, das heißt 1. Variable nach Feld1, 2. Variable nach Feld2, und so weiter.
2. Es dürfen Zahlenvariable nur numerischen Feldern, Strings nur alphanumerischen Feldern zugeordnet werden. Ist dies nicht der Fall, folgt die Fehlermeldung: type mismatch error.
3. Leerstrings bewirken keine Ausgabe, Zahlenvariable ohne Inhalt die Ausgabe von 0.
4. Zahlen werden linksbündig ausgegeben.
5. Überschreitet die Länge einer Variablen (Zahl, String) die Feldlänge, wird nur bis zum Feldende ausgegeben.
6. Der Befehl WMASK fordert eine Variablenliste (analog dem Basic-Befehl INPUT), Stringoperationen und dergleichen sind nicht zulässig.

var1,var2,var3,... stellt die Variablenliste dar.

### ! IMASK übergabe,var1,var2,var3,... :

Dieser Befehl veranlaßt die Eingabe von der Maske, die zuletzt mit dem SCREEN-Befehl auf den Schirm geschrieben wurde. Es werden alle Felder der Maske gelöscht und der Cursor auf das erste Zeichen des ersten Feldes gesetzt. Jetzt kann der Bediener die einzelnen Felder der Maske ausfüllen.

Die Cursortasten, INST, DEL, CLR, HOME und RETURN wirken nur auf die Felder der aktuellen Maske.

Die Sondertasten (F1 bis F8) haben eine Übergabefunktion, ähnlich dem RETURN im herkömmlichen INPUT-Befehl. Durch das Drücken einer dieser acht Tasten, beendet der Bediener die Maskeneingabe. Welche dieser Tasten wirksam werden darf, legt der Programmierer in

**Übergabe** fest. Übergabe ist ein String oder eine Zeichenkette eingeschlossen in Hochkomma, die nur die Zeichenkodes von F1 bis F8 enthalten darf. In diesem String oder mit dieser Zeichenkette legt der Programmierer fest, welche der acht Funktionstasten (F1 ... F8) die Beendigung der Eingabe bewirken können.

**var1,var2,var3,...** sind Variable. Sie müssen in der gleichen Reihenfolge wie die Felder der aufgerufenen Maske stehen, es gilt das gleiche, was bereits für WMASK gesagt wurde, nur daß hier die Feldinhalte in die Variablen übertragen werden. Dabei ist zu beachten, daß bei einem alphanumerischen Feld stets die gesamte Feldlänge in die Variable übertragen wird, gleichgültig, wie viele Zeichen der Bediener eingegeben hat. Zum Beispiel umfaßt ein Feld 20 Zeichen, der Bediener trägt aber nur »Peter« ein, dann werden dem String zusätzlich 15 Leerzeichen mitgegeben.

Mit welcher Taste der Bediener die Maskeneingabe abgeschlossen hat, kann durch Auslesen der Speicherstelle 2 festgestellt werden. Dabei entspricht der Inhalt der Speicherstelle der Tastennummer (1 für F1, 2 für F2, bis 8 für F8).

Beispiel: Eine Maske habe 2 Felder, ein alphanumerisches Feld mit 20 Zeichen Länge und ein numerisches Feld mit einer Länge von 5 Stellen (letzte Längenangabe ist nur der Vollständigkeit halber). Durch Drücken von F1 und F2 kann der Bediener die Maskeneingabe beenden.

```
120 AB$ = CHR$(133) + CHR$(137)
130 !IMASK AB$,F1$,F2
140 ON PEEK(2) GOTO 150,160
150 REM EINGABE MIT F1 ABGESCHLOSSEN
160 REM EINGABE MIT F2 ABGESCHLOSSEN
```

In diesem Beispiel würde SMUBU das Drücken der Tasten F3 bis F8 ignorieren. Anschließend befindet sich der Feldinhalt von Feld 1 in F1\$ und die Zahl von Feld 2 in F2\$.

**! IOMASK übergabe,var1,var2,var3,... :**

Dieser Befehl stellt die Verknüpfung von WMASK mit IMASK dar. Zunächst werden die Felder der Maske gelöscht, die Variableninhalte in die Felder übertragen (analog WMASK) und danach der Cursor auf die erste Stelle des ersten Feldes positioniert (wie bei IMASK-Befehl), nun kann der Bediener die Felder verändern. Die Bedeutung von »Übergabe« und »var1,var2,var3,...« sind der Erklärung von IMASK zu entnehmen.

**! ← var1,var2,var3,... :**

SMUBU kann 255 Felder verwalten, müßte man aber 255 Variable in einer Kette, in eine einzige Basic-Zeile stopfen, hätte man einige Schwierigkeiten. Eine Basic-Zeile darf beim C 64 ja bekanntlich nur 255 Zeichen (Token und Zeichen) beinhalten. Das daraus entstehende Dilemma umgeht dieser Befehl. Er setzt die Variablenliste fort, die durch den Befehl WMASK, IMASK oder IOMASK begonnen wurde. Allerdings muß bei der vorhergehenden Liste ein Verweis auf diesen Befehl vorhanden sein. Dieser Verweis ist der Pfeil nach links hinter einem Komma, anstelle einer Variablen. Es muß aber mindestens eine Variable vor einem solchen Verweis vorhanden sein. In der nächsten Basic-Zeile muß nun der obige Befehl stehen, dieser darf ebenfalls einen Verweis besitzen. Das Beispiel aus der Erklärung für IMASK könnte daher auch so aussehen:

```
130 !IMASK AB$,F1$, ←
135 ! ← F2
```

Wichtig ist auch noch, daß der Verweis wie ein REM-Befehl gehandhabt wird. Alle folgenden Befehle innerhalb der Zeile werden ignoriert.

**! INLINE zeile,spalte,zeichen,string :**

Dieser Befehl ermöglicht es, den Cursor an eine beliebige Position zu setzen. Ab dieser Position wird eine Eingabe erwartet. Es bedeutet:

**zeile:** die Zeilenangabe (0 bis 24), in einer Variablen oder als Festwert.

**spalte:** die Spaltenangabe (0 bis 39). Als Variable oder als Festwert.

**zeichen:** die Anzahl der Zeichen, die eingegeben werden dürfen (0 bis 255). Als Variable oder als Festwert.

**string:** eine Stringvariable, der die Eingabe übermittelt wird, sie hat danach die Länge »zeichen«.

Die Eingabe ist hier nicht so komfortabel, wie beim herkömmlichen INPUT-Befehl. DEL löscht das letzte Zeichen. Hochkommata und alle Funktionstasten (außer RETURN) werden ignoriert. Mit RETURN wird die Eingabe abgeschlossen.

**! DIER fehlernummer,fehlermeldung,tracknummer,sektornummer:**

Mit diesem Befehl wird vom Laufwerk mit der Geräteadresse 8 eine eventuelle Fehlermeldung eingelesen.

**fehlernummer:** ist eine Variable, der die Fehlernummer zugewiesen wird.

**fehlermeldung:** ist ein String, dem die Fehlermeldung im Klartext zugewiesen wird.

**tracknummer** und **sektornummer:** sind Variable, denen die Tracknummer und die Sektornummer in dem der Fehler aufgetaucht ist, zugewiesen werden.

## Das Basic-Programm SMU

Nach der Erklärung der einzelnen Befehle von SMUBU, soll noch auf die Arbeitsweise mit SMU eingegangen werden. Nach dem Programmstart erscheint eine Menüzelle:

MA-ALT MA-NEU UTILITY DRUCKEN ENDE

Durch Drücken von »j« wird der in Inverschrift dargestellte Programmteil ausgewählt. RETURN bewegt die Inverschrift weiter.

Es bedeutet:

**MA - ALT:** Laden einer Maske und deren Veränderung.

**MA - NEU:** Erstellen einer neuen Maske.

**UTILITY:** Bereitstellen von SMUBU auf der Diskette zum Nachladen.

**DRUCKEN:** Laden einer Maske und ausdrucken.

**ENDE:** SMU beenden.

**MA - NEU**

SMU fragt nach dem Maskennamen und der Geräteadresse. Danach wird der Bildschirm gelöscht und die Maske kann entworfen werden. Dabei wird mit »!« ein alphanumerisches und mit »#« ein numerisches Feld festgelegt. Der Maskengenerator faßt jedes Zeichen, das von den beiden Zeichen unterschiedlich ist, als eine Feldbegrenzung beziehungsweise als ein Literal auf.

Vorsicht! Die Tasten DEL und INST wirken nicht mehr wie gewohnt!

**DEL:** löscht die Zeile in der sich der Cursor befindet und zieht alle anderen Zeilen nach.

**INST:** fügt eine Zeile ab der Cursorposition ein und schiebt die restlichen Zeilen nach unten.

Durch Drücken von F1 beginnt der Maskengenerator seine Analyse und schreibt die Maskendaten auf Diskette. Es erfolgt die Rückkehr ins Menü.

**MA - ALT**

SMU fragt den alten Maskennamen ab und setzt wie in »MA

- NEU« beschrieben fort. Allerdings wird die geladene Maske auf den Bildschirm gebracht und der Schirm nicht gelöscht.

#### UTILITY:

Es wird unter dem Namen »SMUBU« ein Lader auf Diskette erzeugt.

#### DRUCKER

lädt die mit Namen spezifizierte Maske und druckt sie aus. Die Geräteadressen sind auf einen GP100 von Seikosha zugeschnitten, sie können durch Veränderung der Parameter in Zeile 730 leicht verändert werden.

## Fehlermeldung von SMUBU

Fehler werden immer gemacht, SMUBU enthält einige Fehlermeldungen, die dem Programmierer bei der Korrektur auf die Sprünge helfen sollen:

#### not smu file:

Es wurde versucht eine Maske mit MASK von der Diskette zu laden, die nicht SMU-Format besitzt. Dieser Fehler tritt auch auf, wenn das File nicht gefunden wurde oder eine andere Fileart als ein sequentielles File angesprochen wurde.

#### not smu data:

Es wurde versucht mit SCREEN ein Stringarray anzusprechen, das keine SMU-Maskendaten enthält. Meist tritt der Fehler deshalb auf, weil ein falscher Index gewählt wurde.

#### no smu mask:

Dieser Fehler kann nur einmal auftreten. Nur dann, wenn noch keine Maske mit SCREEN auf den Schirm gebracht wurde, aber ein Zugriff auf eine Maske mit WMASK, IMASK oder IOMASK versucht wurde.

#### array not found:

Es wurde bei den Befehlen MASK oder SCREEN auf ein Feld zugegriffen, das nicht existiert.

#### transmit error:

Dieser Fehler tritt bei IMASK oder IOMASK auf, wenn der Übergabestring:

1. nicht vorhanden ist
2. leer ist
3. ein Zeichen enthält, dessen Kode von F1 bis F8 unterschiedlich ist.

#### missing link command:

In WMASK, IMASK oder IOMASK wurde ein Verweis auf die Fortsetzung der Variablenliste gegeben, der in der folgenden Zeile nicht erfolgt ist.

#### illegal position:

Bei INLINE wurden Zeilen- beziehungsweise Spaltenangaben verwendet, die über den zulässigen Bereich hinausgehen.

Ein Fehler, der von SMUBU nicht berücksichtigt wurde: Greift man auf eine Maske (mit WMASK, IMASK oder IOMASK), die keine Felder enthält, zeigt der Interpreter die interessantesten Verhaltensweisen, sofern der Computer nicht gleich abstürzt.

## Initialisierung und Einsatz

Da SMUBU im Speicherbereich ab \$9000 abgelegt ist, muß er vor dem Zugriff von Basic her geschützt werden (der Bereich \$C000 ... \$CFFF wurde bewußt vermieden, damit der Programmierer selbst noch die Möglichkeit hat, andere Maschinenroutinen laufen zu lassen). Der Schutz geschieht am einfachsten dadurch, daß man das Basic-RAM-Ende herabsetzt (POKE 56,144). SMUBU bindet man in seine eigenen Programme am besten folgendermaßen ein: Mit UTILITY erstellt man den Objektcode von SMUBU auf einer geeigneten Diskette. Die ersten Zeilen jedes Programmes in dem SMUBU verwendet werden soll, sehen nun so aus:

```
0 IF A=0 THEN POKE 56,144:CLR:A=1LOAD"SMUBU",8,1
```

```
1 CLR:SYS 9*4096:REM INITIALISIEREN VON SMU
```

Wo kann man SMUBU nun überall einsetzen? Bei Dateiverwaltungen, Plattenarchiven, Adreßverwaltungen und vielem mehr, kurzum, überall dort, wo viel mit Eingabe und Ausgabe operiert wird, aber lediglich Daten sich verändern.

## Tips zu SMU

1. Beim Maskenerstellen mit SMU grenzt man die Felder am besten durch Graphikzeichen (zum Beispiel senkrechte Striche) ein. Dadurch sind die Felder beim späteren Aufruf der Maske klar erkennbar.
2. Da die Maske nicht den ganzen Bildschirm in Anspruch nimmt, kann in den letzten beiden Zeilen ein Kommentar untergebracht werden, zum Beispiel welche Sondertaste welche Funktion im Programm hat.
3. Dadurch, daß man bei MASK und SCREEN selbst bestimmen kann, ab welcher Position die Maskendaten im Array stehen, kann man sich lästiges Nachladen von Masken (falls in einem Programm mit mehreren Masken gleichzeitig gearbeitet werden sollte) ersparen, indem man die Masken hintereinander ablegt.
4. SMU überliest Variablen von MASK, IMASK und IOMASK die keinem Feld mehr zugeordnet werden können. Baut man seine Masken geschickt auf, kann man also mit einem WMASK-, IMASK- oder IOMASK-Befehl mehrere Masken abarbeiten.

## LIT.VERW. eine Literaturverwaltung

LIT.VERW. (Listing 4) ist ein Demoprogramm für eine SMU-Anwendung. Um mit diesem Programm arbeiten zu können, sind folgende Schritte erforderlich:

1. Programm »MASKE« (Listing 5) mit dem MSE abtippen und speichern.
  2. Programm »MASKE« mit LOAD"MASKE",8,1 laden.
  3. NEW <RETURN> eingeben.
  4. Listing 6 »SEQ.SAVE« abtippen und mit RUN starten.
- Anschließend befindet sich ein sequentielles File mit dem Namen »ZEITSCHRIFT« auf Ihrer Diskette. Bei diesem File handelt es sich um eine mit dem SMU erstellte Maske. Die Felder der Maske haben folgende Bedeutung:
- Stichwort:** Ein Stichwort, unter dem der Zeitschriftenartikel oder das Buch einzuordnen ist.
- Titel:** Der Titel des Zeitschriftenartikels oder des Buches.
- Autor:** Der Name des Autors im Format: Nachname, Vorname. Der Vorname ist im allgemeinen kürzer.
- Verlag:** Der Name des Verlages.
- Ort:** Der Erscheinungsort der Zeitschrift oder des Buches.
- Jahr:** Jahr der Veröffentlichung des Artikels oder des Buches.

**Seite:** Die Seitenangabe.

**Buch:** Wenn sich das Zitat auf eine Zeitschrift bezieht muß dieses Feld mit »n« ausgefüllt sein.

**Ausgabennummer:** Nur bei einer Zeitschrift von Bedeutung (zum Beispiel 64'er). Die Ausgabennummer der Zeitschrift.

**Zeitschriftenname:** Name der Zeitschrift

LIT.VERW. ist ein maskengesteuertes Programm, den Funktionstasten kommt folgende Bedeutung zu:

#### F1 Eingabe:

Ein Literaturzitat wird in die Datei aufgenommen und in Abhängigkeit von den Suchbegriffen eingeordnet. Da manchmal große Teile des Speichers »umgestellt« werden müssen, dauert dieser Eingabeteil mit zunehmender Größe der Datei immer länger.

**F2 Drucken:**

Der Maskeninhalte wird als Literaturzitat auf dem Drucker ausgegeben. Dieses Zitat sieht für Bücher folgendermaßen aus:  
Nachname, Vorn.:

Buchtitel.

Erscheinungsort: Verlagsname, Veröffentlichungsjahr, Seite.

Für Zeitschriften wurde folgende Zitatangabe verwendet:

Nachname, Vorn.: Artikeltitel.

Zeitschriftenname Ausgabennummer (Erscheinungsjahr) Seite.

**F3 Suchen:**

Es wird die bestehende Datei nach den angegebenen Suchkriterien durchsucht. Als Suchkriterien gelten der Name des Autors und das Stichwort. Eine Verknüpfung beider Kriterien ist dabei möglich. Alle weiteren Feldinhalte haben auf die Suchroutine keinerlei Einfluß.

**F4 Index**

Es wird ein Gesamtindex der Datei ausgegeben. Die Stichwörter sind, wie auch die Autoren alphabetisch geordnet, etwaige Abweichungen sind auf die Ordnung der ASCII-Zeichen zurückzuführen. Die Zahlen hinter dem Stichwort, beziehungsweise dem Autor verweisen auf die Angabe bei der Literaturstelle.

**F5 Weiter:**

Setzt den Suchlauf nach Suchkriterien fort. Wurde zum Bei-

spiel nach einem Autor gesucht, dieser gefunden und angezeigt, kann durch Drücken der Taste F5 der Suchlauf nach weiteren Literaturstellen des Autors fortgesetzt werden. Findet sich kein Datum in der Datei mehr, wird die Maske gelöscht.

**F6 Ende:**

beendet LITVERW. indem es neu eingegebene Literaturzitate sichert. Danach wird der Bildschirm gelöscht und das Programm abgebrochen.

Einiges zum Programm: Das Programm kann jedem Stichwort maximal 255 Verweise zuordnen. Wird dieser Bereich überschritten, erfolgt ein Programmabbruch mit der Meldung: Stichworttabellenüberlauf. Analoges gilt für die Autoren. Wenn das Programm zum erstenmal gestartet wird, das heißt wenn eine neue Datei angelegt wird, muß man die Frage:

Bestehende Datei [j/n] ?

durch Drücken der Taste »N« beantworten. Die Druckerparameter der OPEN-Sequenz in Zeile 1100, 1200 und 9310 sind für einen Seikosha GP 100 VC zugeschnitten, sie müssen für andere Druckermodelle also angepaßt werden. Weiter bewirkt CHR\$(14) in den Zeilen 1220, 1340 und 1470 die Eröffnung des Shift In-Modus, CHR\$(15) beendet den Modus.

(Georg Krcal/ah)

```

11 rem *****
12 rem *
13 rem *          s m u          *
14 rem *
15 rem * (screen management utility) *
16 rem *
17 rem *
19 rem *          georg krcal    *
20 rem * ziegelhofstrasse 36/11/9 *
21 rem * a - 1220 wien          *
22 rem * telefon 22-17-724      *
23 rem *
24 rem *****
25 rem
26 rem
27 rem
28 rem
29 ifs=0then:sc=1:load"smubu",8,1
30 ifs=1then:sc=2:load"smu.m",8,1
60 poke 56,144:clr
65 print"CHR$(14)CHR$(8);:poke53280,0
:poke53281,0
70 k$=" MA-ALT MA-NEU UTILITY DRUCKEN
ENDE "
75 for i=1 to 25:cd$=cd$+" ":fi$=fi$+"####
#####:ru$=ru$+"!!!!!!!!!!"
80 cr$=cr$+"|":next
85 cd$="|"+cd$:ru$=ru$+"!!!!!!":fi$=fi$+"
#####"
90 dim b$(14),a$(24),bc%(1024)
100 rem **** erstellen des bildschirms
110 print"##### *** ***
*** *** ";
120 print" *** *** **** ***
*** ";
130 print" *** *** ***** ***** ***
*** ";
140 print" *** *** ***** *** ***
*** ";
150 print" ***** *** ** *** ***
*** ";
160 print" *** *** *** ***
*** ";
180 print" *** *** *** ***
*** ";
190 print" *** *** *** ***
* *** ";
200 print" ***** *** *** *
***** ";
210 print"tab(6) SCREEN MANAGEMENT
UTILITY"
220 print
230 printtab(12)"Von Georg Krcal"
240 printtab(13)"Wien, Mai 1985":ifxthen
250
242 printleft$(cd$,24) BITTE T
ASTE DRUECKEN
243 poke198,0:wait198,1:poke198,0
244 sys9*4096
250 printleft$(cd$,24)k$;:x=1
390 rem kommandoentgegennahme
400 s$=left$(k$,x-1)+"|"+mid$(k$,x,8)+"|
"+right$(k$,33-x)
405 printleft$(cd$,24)s$;" ";
410 wait203,63
420 getg$:ifg$="j"then460
430 ifg$<>chr$(13)then410
440 x=x+8:ifx>=40thenx=1
450 goto400
460 printleft$(cd$,24)
" "
470 x=(x-1)/8+1:on x goto 480,500,900,67
0,471
471 print" ";:end
480 printleft$(cd$,24)"Alter Dateiname I
"
490 !inline 23,17,16,an$:h$=an$:gosub 20
00:an$=h$:ifan$=""then490
500 printleft$(cd$,24)"Neuer Dateiname I
"
510 !inline 23,17,16,nn$:h$=nn$:gosub 20
00:nn$=h$:ifnn$=""then510

```

Listing 1. Der Maskengenerator »SMU«

```

520 printleft$(cd$,24)"
      [RETURN]
530 printleft$(cd$,24)"Geraetenummer I
      [RETURN]
540 !inline 23,15,3,gn$:h$=gn$:gosub 200
0:gn$=h$:ifgn$=""then540
550 gn=val(gn$):if(gn<>1andgn<4)orgn>255
then530
555 printleft$(cd$,24)"
      [RETURN]
560 if x=2 then sys12*4096:goto640:rem n
eue maske
580 !mask an$,gn,b$(0):!screen b$(0)
590 fori=1to255:cd=peek(39445+i):cr=peek
(39700+i):fe=peek(39955+i)
600 ch=peek(40210+i):ifcd=128then630
610 h$=fi$:iffe=1thenh$=ru$
620 printleft$(cd$,cd+1)spc(cr)left$(h$,
ch):next:ifx=4thenreturn
630 ifx=4thenreturn
640 sys49162:nn$="@: "+nn$+" ,s,w"
650 sys49589,nn$,gn:rem maske analysiere
n und auf peripherie
652 ifpeek(832)=0thenprint" ";:goto100
654 h$="!":ifpeek(832)=1thenh$="#"
656 printleft$(cd$,24)"FEHLER: "h$ - Fe
ld >255 Zeichen - SPACE tasten";
658 h=len(nn$)-6:nn$=mid$(nn$,3,h)
660 getg$:ifg$=""then660
662 printleft$(cd$,24)"

664 goto640
670 printleft$(cd$,24)"Dateiname I
      [RETURN]
680 !inline 23,11,16,an$:h$=an$:gosub 20
00:an$=h$:ifan$=""then670
690 gosub 520
700 fori=0to24:a$(i)="":forj=0to39:bc=pe
ek(1024+i*40+j)
710 ifbc>127theng$="":goto725

```

```

715 ifbc<65andbc>31theng$=chr$(bc):goto7
25
717 ifbc<31orbc>96theng$=chr$(bc+64):got
o725
720 g$=chr$(bc+32)
725 a$(i)=a$(i)+g$:nextj,i
730 open4,4,7:rem drucker
740 print#4,"      0      1      2
      3"
750 print#4,"      0....5....0....5....0...
.5....0....5...."
755 print#4
760 fori=0to22:h$=str$(i):h$=right$(h$,
len(h$)-1))
770 iflen(h$)=1thenh$="0"+h$
780 a$(i)=h$+" "+a$(i)
790 print#4,a$(i):next
800 close 4:rem drucker schliessen
810 print" ";:goto100
900 printleft$(cd$,24)"Geraetenummer I
      [RETURN]
910 !inline 23,15,3,gn$:h$=gn$:gosub 200
0:gn$=h$:ifgn$=""then910
920 gn=val(gn$):if(gn<>1andgn<4)orgn>255
then900
930 printleft$(cd$,24)"
      [RETURN]
940 open8,gn,1
950 poke 183,7:poke187,169:poke188,2
960 poke 681,64:poke682,58:poke683,83:po
ke684,77:poke685,85:poke686,66
970 poke687,85:poke251,0:poke252,144
980 poke 781,23:poke782,154:poke780,251:
sys65496
1000 close8:print" ";:goto100
2000 forj=len(h$)tolstep-1:ifmid$(h$,j,1
)="" thennext
2010 h$=left$(h$,j):return
ready.

```

Listing 1. Der Maskengenerator »SMU« (Schluß)

programm : smu.m c000 c32e

```

c000 : 4c 07 c0 03 83 8d 22 20 ce
c008 : 44 e5 20 66 e5 ea a5 c6 ee
c010 : 85 cc f0 fa a5 cf f0 0c 4b
c018 : a5 ce ae 87 02 ae 00 84 ef
c020 : cf 20 13 ea 20 b4 e5 8d 7c
c028 : c0 02 a2 03 bd 03 c0 cd 85
c030 : c0 02 f0 d9 ca 10 f5 38 de
c038 : 20 f0 ff ad c0 02 c9 94 f2
c040 : f0 51 c9 14 f0 4d c9 85 79
c048 : d0 01 60 e0 15 d0 1e c9 b1
c050 : 0d f0 b7 c9 9d f0 36 c9 ca
c058 : 11 d0 08 a2 00 18 20 f0 4b
c060 : ff 90 aa c0 27 d0 06 20 bb
c068 : d2 ff 4c 0a c0 e0 00 d0 43
c070 : 1c c0 00 d0 0f c9 9d d0 5e
c078 : 0b a2 15 a0 27 18 20 f0 c3
c080 : ff 4c 0d c0 c9 91 d0 05 77
c088 : a2 15 4c 7d c0 20 d2 ff d0
c090 : 4c 0d c0 a9 00 8d c1 02 40
c098 : a9 04 8d c2 02 ad c0 02 94
c0a0 : c9 94 f0 01 e8 e0 00 f0 87
c0a8 : 14 ad c1 02 18 69 28 8d cc
c0b0 : c1 02 ad c2 02 69 00 8d bd
c0b8 : c2 02 ca 00 ec 18 ad c1 12
c0c0 : 02 8d c3 02 ad c2 02 69 86
c0c8 : d4 8d c4 02 ad c1 02 ae 22
c0d0 : c2 02 85 5f 86 60 ad c0 84
c0d8 : 02 c9 94 f0 41 38 ad c1 12
c0e0 : 02 e9 28 85 5a ad c2 02 b4
c0e8 : e9 00 85 5b a9 c0 a2 07 d7
c0f0 : 85 58 86 59 20 6f c1 ad 4e
c0f8 : c3 02 ae c4 02 85 5f 86 d7

```

```

c100 : 60 38 ad c3 02 e9 28 85 7b
c108 : 5a ad c4 02 e9 00 85 5b 16
c110 : a9 c0 a2 db 85 58 86 59 25
c118 : 20 6f c1 4c 0d c0 a9 98 98
c120 : a2 07 85 5a 86 5b a9 c0 5e
c128 : 85 58 86 59 20 bf a3 ad 90
c130 : c3 02 ae c4 02 85 5f 86 0f
c138 : 60 a9 98 a2 db 85 5a 86 48
c140 : 5b a9 c0 85 58 86 59 20 b0
c148 : bf a3 38 20 f0 ff 8e c5 c0
c150 : 02 8c c6 02 20 ff e9 a2 79
c158 : 16 a0 00 18 20 f0 ff 20 8b
c160 : ff e9 ae c5 02 ac c6 02 5d
c168 : 18 20 f0 ff 4c 0d c0 a0 3e
c170 : 00 b1 5f 91 5a c8 c0 28 92
c178 : d0 f7 18 a5 5f 69 28 85 ec
c180 : 5f a5 60 69 00 85 60 18 d5
c188 : a5 5a 69 28 85 5a a5 5b 32
c190 : 69 00 85 5b a5 5f c5 58 e3
c198 : d0 d5 a5 60 c5 59 d0 cf d2
c1a0 : 60 ff ff ff ff ff ff ff 00
c1a8 : ff ff ff ff ff ff ff ff a7
c1b0 : ff ff ff ff ff 20 fd ae 05
c1b8 : 20 9e ad 20 a3 b6 8d 3e 39
c1c0 : 03 8e 3f 03 8c 40 03 20 52
c1c8 : fd ae 20 9e b7 a9 01 a0 06
c1d0 : 02 20 ba ff ad 3e 03 ae c7
c1d8 : 3f 03 ac 40 03 20 bd ff f4
c1e0 : 20 c0 ff a2 01 20 c9 ff ed
c1e8 : a9 00 8d 3c 03 8d 3d 03 14
c1f0 : 8d 3e 03 8d 40 03 85 69 14
c1f8 : 85 6b 8d 43 03 a9 01 8d 9b
c200 : 41 03 a9 04 85 6a a9 d8 b2
c208 : 85 6c a2 00 a9 80 9d 33 ed
c210 : c3 9d 33 c4 9d 33 c5 9d cd

```

```

c218 : 33 c6 e8 d0 ef a9 de 20 0a
c220 : d2 ff a9 64 20 d2 ff a0 c3
c228 : 00 8c 3d 03 ac 3d 03 b1 42
c230 : 69 8d 42 03 20 d2 ff b1 4d
c238 : 6b 20 d2 ff ad 42 03 c9 f4
c240 : 21 d0 14 ad 40 03 f0 07 72
c248 : c9 02 f0 20 20 f4 c2 a9 5a
c250 : 02 20 15 c3 4c 6c c2 c9 e7
c258 : 23 d0 16 ad 40 03 f0 07 0d
c260 : c9 01 f0 08 20 f4 c2 a9 ef
c268 : 01 20 15 c3 ee 3e 03 f0 06
c270 : 4c ad 40 03 f0 0e ad 42 be
c278 : 03 c9 21 f0 07 c9 23 f0 f3
c280 : 03 20 f4 c2 ad 3d 03 c9 8d
c288 : 27 f0 06 ee 3d 03 4c 2c fc
c290 : c2 ee 3c 03 ad 3c 03 c9 95
c298 : 16 f0 1d 18 a5 69 69 28 0c
c2a0 : 85 69 a5 6a 69 00 85 6a 12
c2a8 : 18 a5 6b 69 28 85 6b a5 42
c2b0 : 6c 69 00 85 6c 4c 27 c2 cd
c2b8 : a9 00 8d 41 03 a9 00 8d 85
c2c0 : 3c 03 ae 3c 03 bd 33 c3 23
c2c8 : 20 d2 ff ae 3c 03 bd 33 60
c2d0 : c4 20 d2 ff ae 3c 03 bd ad
c2d8 : 33 c5 20 d2 ff ae 3c 03 bd
c2e0 : bd 33 c6 20 d2 ff ee 3c 4e
c2e8 : 03 d0 7d 20 c2 ff a9 01 c3
c2f0 : 20 c3 ff 60 ae 43 03 ad 6a
c2f8 : 3e 03 9d 33 c6 a9 00 8d 5a
c300 : 3e 03 8d 40 03 ee 43 03 e6
c308 : f0 01 60 6b 68 a9 02 8d 95
c310 : 41 03 ac eb c2 8d 40 03 03
c318 : 38 e9 01 ae 43 03 9d 33 84
c320 : c5 ad 3c 03 9d 33 c3 ad 09
c328 : 3d 03 9d 33 c4 60 00 ff 04

```

Listing 2. Das Maschinenprogramm zu »SMU«. Beachten Sie bitte die Eingabehinweise auf Seite 54

programm : smubu 9000 9a17

```

9000 : a9 fb a2 90 8d 08 03 8e a4
9008 : 09 03 a9 00 8d 10 90 60 5a
9010 : 00 37 97 00 00 00 00 00 92
9018 : 00 00 00 00 00 00 00 00 19
9020 : 00 00 00 00 00 00 43 4c c6
9028 : 45 49 4e 2e 53 45 54 49 ae
9030 : 4e 2e 4d 41 53 4b 2e 53 00
9038 : 43 52 45 45 4e 2e 57 4d ed
9040 : 41 53 4b 2e 49 4d 41 53 6e
9048 : 4b 2e 49 4f 4d 41 53 4b a9
9050 : 2e 49 4e 4c 49 4e 45 2e b8
9058 : 44 49 45 52 2e 71 91 5b 48
9060 : 91 7d 91 6b 92 35 93 d2 48
9068 : 93 22 97 37 97 07 98 85 f8
9070 : 89 86 8a 87 8b 88 8c 2c 57
9078 : 53 2c 52 8b 90 97 90 a3 37
9080 : 90 b1 90 c1 90 d0 90 d8 c9
9088 : 90 e3 90 4e 4f 54 20 53 b6
9090 : 4d 55 20 46 49 4c c5 4e 03
9098 : 4f 54 20 53 4d 55 20 44 0c
90a0 : 41 54 c1 4e 4f 54 20 53 04
90a8 : 4d 55 20 41 52 52 41 59 40
90b0 : d3 49 4c 4c 45 47 41 4c f1
90b8 : 20 50 4f 53 49 54 49 4f 3a
90c0 : ce 41 52 52 41 59 20 4e 0a
90c8 : 4f 54 20 46 4f 55 4e c4 75
90d0 : 54 52 41 4e 53 4d 49 d4 d6
90d8 : 4e 4f 20 53 4d 55 2d 4d 0f
90e0 : 41 53 cb 4d 49 53 53 49 76
90e8 : 4e 47 20 4c 49 4e 4b 20 e0
90f0 : 53 4d 55 20 43 4f 4d 4d c2
90f8 : 41 4e c4 20 73 00 c9 21 36
9100 : f0 06 20 79 00 40 c7 a7 7c
9108 : a2 ff a0 ff a5 7a 8d c0 b8
9110 : 02 a5 7b 8d c1 02 c8 c8 56
9118 : a9 00 85 59 ad c0 02 85 42
9120 : 7a ad c1 02 85 7b e8 bd 75
9128 : 26 90 f0 2c c9 2e f0 14 52
9130 : 85 02 a5 59 d0 f0 20 73 47
9138 : 00 c5 02 f0 e9 a9 01 85 b4
9140 : 59 4c 26 91 a5 59 d0 ce 81
9148 : 8b b9 5d 90 8d 11 90 b9 2d
9150 : 5e 90 8d 12 90 6c 11 90 6e
9158 : 4c 08 af a9 34 8d 14 03 cf
9160 : a9 70 a2 91 8d 18 03 8e df
9168 : 19 03 20 73 00 4c ae a7 e6
9170 : 40 a9 31 8d 14 03 58 a9 91
9178 : 47 a2 fe d0 e7 20 73 00 37
9180 : 20 9e ad 20 a3 b6 f0 04 1a
9188 : c9 11 90 03 4c 48 b2 85 3b
9190 : 02 86 59 84 5a a8 b1 59 21
9198 : 99 13 90 88 10 f8 a2 00 43
91a0 : a4 02 bd 77 90 99 13 90 e7
91a8 : e8 c8 e0 04 d0 f4 84 02 78
91b0 : 20 fd ae 20 9e b7 86 5b f7
91b8 : 20 fd ae 20 3e 98 a9 08 e6
91c0 : a6 5b a0 02 20 ba ff a5 9f
91c8 : 02 a2 13 a0 90 20 bd ff f5
91d0 : 20 c0 ff a2 08 20 c6 ff 41
91d8 : 20 cf ff c9 de d0 07 20 ea
91e0 : cf ff c9 64 f0 05 a9 00 8c
91e8 : 4c 25 98 a9 0b a2 a0 20 ab
91f0 : ae 98 a5 47 a6 48 bd 13 46
91f8 : 90 8e 14 90 a9 04 a2 ff 2c
9200 : 85 02 8e c2 02 ad c0 02 17
9208 : 85 64 ad c1 02 85 65 ad a0
9210 : c2 02 20 75 b4 a0 02 b9 56
9218 : 61 00 91 47 88 10 f8 20 f4
9220 : e4 98 20 46 99 c6 02 d0 9b
9228 : dc a9 00 8d c0 02 a9 04 55
9230 : 85 02 ad 13 90 ae 14 90 74
9238 : 85 47 86 48 a0 01 b1 47 73
9240 : 85 49 c8 b1 47 85 4a 20 dc
9248 : cf ff ac c0 02 91 49 20 6c
9250 : e4 98 c6 02 d0 e6 ee c0 f4
9258 : 02 ad c0 02 c9 ff d0 ce 1f
9260 : 20 cc ff a9 08 20 c3 ff ac
9268 : 4c ae a7 20 73 00 20 3e 2d
9270 : 98 a5 47 a6 48 bd 13 90 e9
9278 : 8e 14 90 a9 0b a2 a0 20 f2
9280 : f2 98 a9 15 20 09 99 a9 cf

9288 : 16 a2 9a 20 38 99 a9 27 df
9290 : 20 09 99 a9 15 a2 9b 20 e5
9298 : 38 99 a9 01 20 09 99 a9 2b
92a0 : 14 a2 9c 20 38 99 a9 ff 27
92a8 : 20 09 99 a9 13 a2 9d 20 e5
92b0 : 38 99 a9 80 8d 10 90 ad 26
92b8 : 13 90 ae 14 90 85 47 86 a1
92c0 : 48 a9 00 a2 04 85 57 86 08
92c8 : 58 85 59 a9 d8 85 5a a2 d7
92d0 : 0b a0 01 b1 47 85 49 c8 f9
92d8 : b1 47 85 4a a0 00 84 02 f8
92e0 : 8c c0 02 ac c0 02 b1 49 58
92e8 : c9 21 f0 04 c9 23 d0 02 fb
92f0 : a9 20 8c c0 02 a4 02 91 55
92f8 : 57 ac c0 02 c8 b1 49 8c 6e
9300 : c0 02 a4 02 91 59 e6 02 ae
9308 : ee c0 02 ac c0 02 c0 a0 cd
9310 : d0 d1 20 4a 98 18 a5 57 fd
9318 : 69 50 85 57 a5 58 69 00 b8
9320 : 85 58 18 a5 59 69 00 85 b9
9328 : 59 a5 5a 69 00 85 5a ca 43
9330 : d0 9f 4c ae a7 20 73 00 02
9338 : ad 10 90 2a b0 05 a9 06 3d
9340 : 4c 25 98 20 58 99 a9 00 42
9348 : 85 5b a6 5b bd 16 9a 48 17
9350 : bd 15 9b a8 bd 13 9d 85 8a
9358 : 02 bd 14 9c 85 60 68 aa 24
9360 : 18 20 f0 ff a5 60 d0 1f a3
9368 : 20 28 af 20 8d ad 20 dd 0f
9370 : bd a0 0f a9 01 84 59 85 57
9378 : 5a 84 5f ae 00 01 e0 20 8e
9380 : d0 17 85 59 4c 99 93 20 88
9388 : 28 af 20 a3 b6 86 59 84 12
9390 : 5a 85 5f c9 00 f0 16 a0 df
9398 : 00 b1 59 f0 10 20 ca f1 f6
93a0 : c8 c4 02 f0 08 a5 5f f0 76
93a8 : f0 c4 5f 9a 0c e6 5b a6 7c
93b0 : 5b bd 16 9a 30 09 20 fd 8b
93b8 : ae 20 d9 99 4c 4a 93 ad e1
93c0 : 10 90 2a 2a 90 03 4c 2d 95
93c8 : 94 20 06 a9 20 fb a8 4c 40
93d0 : ae a7 ad 10 90 30 05 a9 b1
93d8 : 06 4c 25 98 20 73 00 20 3f
93e0 : 9e ad 20 a3 b6 f0 04 c9 68
93e8 : 09 90 05 a9 05 4c 25 98 28
93f0 : 85 02 86 49 84 4a a0 00 5e
93f8 : b1 49 c9 85 90 ed c9 8d 2c
9400 : b0 e9 99 13 90 c8 c4 02 d4
9408 : d0 ee a9 00 99 13 90 20 6e
9410 : fd ae a5 7a a6 7b 8d c0 1b
9418 : 02 8e c1 02 ba 8e c2 02 41
9420 : ad 10 90 2a 2a 90 03 4c 0b
9428 : 43 93 20 58 99 a2 00 86 04
9430 : 5e 86 57 86 5b 86 5c bd 4f
9438 : 15 9b a8 bd 16 9a aa 18 0e
9440 : 20 f0 ff 20 82 99 c9 85 03
9448 : 90 24 c9 8d b0 20 a2 00 a5
9450 : bd 13 90 f0 ee c5 02 f0 e0
9458 : 03 e8 10 f4 a2 07 a5 02 6f
9460 : dd 6f 90 f0 03 ca d0 f8 f3
9468 : e8 86 02 4c 37 96 c9 22 31
9470 : f0 d1 c9 20 b0 03 4c ce b1
9478 : 94 c9 80 90 07 c9 a0 b0 c6
9480 : 03 4c ce 94 a6 57 bd 14 34
9488 : 9c d0 1a a5 02 c9 45 f0 2d
9490 : 14 c9 2d f0 10 c9 2e f0 dc
9498 : 0c c9 2b f0 08 c9 30 90 22
94a0 : a2 c9 3a b0 99 a9 00 85 0e
94a8 : 5b 85 5c a5 02 e6 5e 20 a3
94b0 : ca f1 a6 57 bd 13 9d c5 7e
94b8 : 5e d0 88 a9 00 85 5e e6 49
94c0 : 57 a6 57 bd 16 9a 10 03 74
94c8 : 4c 2d 94 4c 37 94 c9 93 c0
94d0 : f0 0c c9 13 f0 08 48 a9 5f
94d8 : 00 85 5c 85 5b 68 c9 0d 9d
94e0 : f0 d9 c9 11 f0 d5 c9 8d 51
94e8 : f0 d1 c9 13 d0 14 a5 5b 91
94f0 : f0 03 4c 2d 94 a9 00 85 bc
94f8 : 5e a6 57 a9 01 85 5b 4c f7
9500 : c1 94 c9 93 d0 33 a5 5c e6
9508 : f0 06 20 58 99 4c 2d 94 e8
9510 : a9 01 85 5c a6 57 bd 15 6d
9518 : 9b a8 bd 16 9a aa 18 20 d9
9520 : f0 ff a6 57 bd 13 9d aa e5
9528 : a9 20 20 ca f1 ca d0 fa f1

9530 : a9 00 85 5e a6 57 4c c1 e0
9538 : 94 c9 91 d0 1a a9 00 85 29
9540 : 5e c6 57 a6 57 e0 ff d0 ca
9548 : 0b e6 57 a6 57 bd 16 9a 62
9550 : 10 f7 c6 57 4c c1 94 c9 b1
9558 : 1d d0 05 85 02 4c a5 94 12
9560 : c9 9d d0 46 a5 5e f0 0b 1c
9568 : c6 5e c6 5e a9 9d 85 02 7c
9570 : 4c a5 94 c6 57 a6 57 e0 56
9578 : ff d0 0b e6 57 a6 57 bd 02
9580 : 16 9a 10 f7 c6 57 a6 57 57
9588 : bd 15 9b a8 bd 16 9a aa 18
9590 : 18 20 f0 ff a6 57 bd 13 37
9598 : 9d aa ca 86 5e f0 08 a9 ef
95a0 : 1d 20 ca f1 ca d0 fa 4c 76
95a8 : 43 94 c9 94 d0 3f a6 57 8a
95b0 : bd 13 9d a8 88 f0 33 98 81
95b8 : c5 5e c6 5e a9 9d 85 85 e5
95c0 : 59 20 6c e5 20 24 ea 20 10
95c8 : 9e 99 a4 59 b1 61 c9 20 15
95d0 : d0 18 88 b1 61 c8 91 61 6a
95d8 : 88 b1 63 c8 91 63 88 d0 23
95e0 : f1 a9 20 91 61 ad 8c 02 82
95e8 : 91 63 4c 43 94 c9 14 d0 30
95f0 : 43 a6 57 bd 13 9d a8 88 e6
95f8 : 84 59 d0 08 a9 20 20 ca 10
9600 : f1 4c 08 96 a5 5e f0 2c 56
9608 : a9 9d 20 ca f1 20 6c e5 7f
9610 : 20 24 ea 20 9e 99 a4 59 fd
9618 : f0 1a c8 98 38 e5 85 85 92
9620 : 60 a0 00 c8 b1 61 88 91 55
9628 : 61 c8 c4 60 d0 f5 a9 20 ce
9630 : 91 61 c6 5e 4c 43 94 ad 7c
9638 : c0 02 ae c1 02 85 7a 86 20
9640 : 7b a2 00 86 57 a6 57 bd 61
9648 : 16 9a 48 bd 15 9b a8 68 17
9650 : aa 18 20 f0 ff a6 57 bd 3a
9658 : 14 9c f0 33 20 8b b0 85 89
9660 : 49 84 4a 20 a3 b6 a6 57 bb
9668 : bd 13 9d 85 58 20 75 b4 8d
9670 : a0 02 b9 61 00 91 49 88 6f
9678 : 10 f8 20 69 99 a0 00 20 22
9680 : cf ff 91 62 c8 c4 61 d0 d9
9688 : f6 20 d0 99 4c da 96 20 2c
9690 : 8b b0 85 49 84 a4 20 8d 34
9698 : ad a6 57 bd 13 9d 85 58 0b
96a0 : 20 b9 99 a0 00 20 cf ff 57
96a8 : 99 c3 02 c8 c4 58 d0 f5 fb
96b0 : a9 00 99 c3 02 20 d0 99 d0
96b8 : a9 c3 a2 02 85 2d 86 23 f5
96c0 : a9 fe a2 96 8d 00 03 8e 66
96c8 : 01 03 20 b5 b7 a9 8b a2 46
96d0 : e3 8d 00 03 8e 01 03 20 18
96d8 : d0 bb e6 57 a6 57 bd 16 73
96e0 : 9a 30 09 20 fd ae 20 d9 62
96e8 : 99 4c 45 96 20 06 a9 20 e5
96f0 : fb a8 a9 80 8d 10 90 ae b3
96f8 : c2 02 9a 4c ae a7 8a 30 9e
9700 : cc 48 a9 8b a2 e3 8d 00 4c
9708 : 03 8e 01 03 68 c9 0f f0 e6
9710 : 03 4c 8e e3 68 68 68 68 96
9718 : 68 a9 00 85 5e a6 57 4c 17
9720 : 37 94 ad 10 90 30 05 a9 01
9728 : 06 4c 25 98 ad 10 90 09 60
9730 : 40 8d 10 90 4c d2 93 20 37
9738 : 73 00 20 9e b7 e0 19 90 8f
9740 : 05 a9 03 4c 25 98 86 57 44
9748 : 20 fd ae 20 9e b7 e0 28 92
9750 : b0 ef 86 59 20 fd ae 20 b2
9758 : 9e b7 86 5f e0 00 d0 03 b7
9760 : 4c 48 b2 20 fd ae a9 00 7d
9768 : 85 5e a6 57 a4 59 18 20 67
9770 : f0 ff 20 82 99 c9 0d f0 b6
9778 : 56 c9 14 d0 18 a6 5e f0 e4
9780 : f1 c6 5e a9 9d 20 ca f1 8b
9788 : a9 20 20 ca f1 a9 9d 20 c6
9790 : ca f1 4c 72 97 c9 22 f0 e6
9798 : d9 c9 20 90 d5 c9 80 90 3f
97a0 : 1f c9 a0 90 cd c9 c0 90 2d
97a8 : 17 c9 c2 90 c5 c9 ea 90 de
97b0 : 0f c9 e0 90 bd c9 fe 90 35
97b8 : 07 c9 ff f0 03 4c 72 97 df
97c0 : a6 5e a4 5f f0 ac e8 86 4d
97c8 : 5e 20 ca f1 4c 72 97 a5 29
97d0 : 5f 48 a6 57 a4 59 18 20 9e

```

Listing 3. Mit dem Programm »SMUBU« lassen sich durch 10 neue Basic-Befehle Masken in eigene Programme einbinden

```

97d8 : f0 ff 20 8b b0 85 49 84 a7
97e0 : 4a 20 a3 b6 68 20 75 b4 c1
97e8 : a0 02 b9 61 00 91 49 88 e7
97f0 : 10 f8 20 b9 99 a0 00 20 9a
97f8 : cf ff 91 62 c8 c4 61 d0 51
9800 : f6 20 d0 99 4c ae a7 a9 9a
9808 : 80 a2 08 a0 0f 20 ba ff cc
9810 : a9 00 20 bd ff 20 c0 ff 7d
9818 : 20 73 00 a2 80 20 ad ab 5d
9820 : 4c ae a7 ea ea 0a 48 20 6b
9828 : cc ff a9 08 20 c3 ff 68 50
9830 : aa bd 7b 90 85 22 bd 7c 03
9838 : 90 85 23 4c 4a a4 20 8b 3f
9840 : b0 20 8f ad a5 64 8d c0 cf
9848 : 02 a5 65 8d c1 02 a6 2f 4d
9850 : a5 30 86 5c 85 5d c5 32 f9
9858 : d0 04 e4 31 f0 1d a0 00 04
9860 : b1 5c c8 c5 45 d0 06 a5 68
9868 : 46 d1 5c f0 13 c8 b1 5c c3
9870 : 18 65 5c aa c8 b1 5c 65 fe
9878 : 5d 90 d7 a9 04 4c 25 98 b1
9880 : c8 b1 5c 18 65 5c 85 5e 47
9888 : c8 b1 5c 65 5d 85 5f 18 9c
9890 : a5 47 69 2d 85 60 a5 48 5b
9898 : 69 00 85 59 a5 5f c5 59 ad
98a0 : 90 08 d0 09 a5 5e c5 60 af
98a8 : b0 03 4c 45 b2 60 85 02 de
98b0 : 8e c2 02 ad c0 02 85 64 d1
98b8 : ad c1 02 85 65 ad c2 02 4a
98c0 : 20 75 b4 a0 02 b9 61 00 4f
98c8 : 91 47 88 10 f8 a0 00 20 f6
98d0 : cf ff 91 62 c8 c4 61 d0 29
98d8 : f6 20 e4 98 20 46 99 c6 53
98e0 : 02 d0 d0 60 18 a5 47 69 29
98e8 : 03 85 47 a5 48 69 00 85 0f
98f0 : 48 60 85 02 86 59 a0 00 c0
98f8 : b1 47 c5 59 f0 03 4c 13 68
9900 : 99 20 e4 98 c6 02 d0 f0 97
9908 : 60 85 02 a0 00 b1 47 c9 fe
9910 : ff f0 05 a9 01 4c 25 98 36
9918 : c8 b1 47 99 44 00 c0 02 09
9920 : d0 f6 a0 fe b1 45 c0 09 8b
9928 : a5 02 d1 45 b0 03 4c 13 66
9930 : 99 88 10 f0 20 e4 98 60 7c
9938 : 85 49 86 4a a0 00 b1 45 a8
9940 : 91 49 c8 d0 f9 60 18 ad 20
9948 : c0 02 69 01 8d c0 02 ad c6
9950 : c1 02 69 00 8d c1 02 60 1c
9958 : a9 00 85 02 a6 02 bd 16 41
9960 : 9a 30 1e 48 bd 15 9b a8 e7
9968 : bd 13 9d 85 59 68 aa 18 7b
9970 : 20 f0 ff a9 20 20 ca f1 4f
9978 : c6 59 d0 f9 e6 02 4c 5c c6
9980 : 99 60 a5 c6 85 cc f0 fa 04
9988 : a5 cf f0 0c a5 ce ae 87 6d
9990 : 02 a0 00 84 cf 20 13 ea 93
9998 : 20 b4 e5 85 02 60 18 a5 0b
99a0 : d1 65 d3 85 61 a5 d2 69 2b
99a8 : 00 85 62 18 a5 f3 65 d3 3e
99b0 : 85 63 a5 f4 69 00 85 64 64
99b8 : 60 a9 03 a2 03 a0 03 20 83
99c0 : ba ff a9 00 20 bd ff 20 14
99c8 : c0 ff a2 03 20 c6 ff 60 8a
99d0 : 20 cc ff a9 03 20 c3 ff cc
99d8 : 60 c9 5f f0 01 60 20 09 b8
99e0 : a9 20 fb a8 20 09 a9 a9 f1
99e8 : fb a2 99 8d 08 03 8e 09 31
99f0 : 03 4c b1 a7 68 68 a9 07 f9
99f8 : 4c 25 98 a9 fb a2 90 8d 64
9a00 : 08 03 8e 09 03 20 73 00 4e
9a08 : c9 21 d0 e8 20 73 00 c9 e4
9a10 : 5f d0 e1 4c 73 00 00 ff 10

```

Listing 3. »SMUBU« (Schluß)

```

10 rem *****
15 rem *
20 rem * literatur verwaltung *
25 rem *
30 rem * demoprogramm *
35 rem * zu smu *
40 rem *
45 rem * autor: georg krcal *
50 rem *
60 rem * ziegelhofstrasse 36/11/9 *
65 rem * a-1220 wien *
70 rem * telefon: 22-17-724 *
75 rem *
80 rem *****
100 ifa=0thenpoke56,144:clr:a=1:load"smu
bu",8,1
110 clr:sys9*4096
120 printchr$(147)chr$(8)chr$(14)chr$(30
);
130 poke53280,0:poke53281,0
140 dim m$(14),h$(20)
150 rem
160 !mask "zeitschrift",8,m$(0)
170 fori=1to40:cd$=cd$+chr$(17)
180 cr$=cr$+chr$(29):bl$=bl$+" ":next
190 cd$=chr$(18)+cd$:x=fre(0)
200 ri$=chr$(18):ro$=chr$(146)
210 bl$=ri$+"1"+ro$+" eingeben "+ri$+"3"
+ro$+" suchen "+ri$+"5"+ro$
220 bl$=bl$+" weiter "
230 b2$=ri$+"2"+ro$+" drucken "
240 b2$=b2$+ri$+"4"+ro$+" index "+ri$
+"6"+ro$+" prg-ende"
300 print"Diskette formatieren [j/n]";w
ait 203,63
310 getg$:ifg$<>"j"then380
320 printg$"a":print"Diskettenname:"
330 print"Disketten - ID":!inline 1,15,1
6,dn$
340 !inline2,16,2,di$:ifdi$=" "then340
350 h$="n:"+dn$+",""+di$:print:print" Di
skette einlegen - Taste druecken"
360 getg$:ifg$=" "then360
370 rem "open 1,8,15,h$:close1"
380 printchr$(147)"Bestehende Datei [j/n
] ?"
390 wait203,63:getg$:ifg$="j"then500
400 ifg$<>"n"then390
410 open1,8,2,"literatur,1,"+chr$(202)
420 gosub10000
440 open2,8,15
450 print#2,"p"+chr$(2)+chr$(44)+chr$(1)
+chr$(1)
460 print#1,chr$(255)
470 print#2,"p"+chr$(2)+chr$(1)+chr$(1)+
chr$(1)
480 print#1,1,chr$(13);1;chr$(13);1
490 close2:close1
500 gosub10000:open1,8,2,"literatur,1,"+
chr$(202)
510 open2,8,15
520 print#2,"p"+chr$(2)+chr$(1)+chr$(1)+
chr$(1)
530 input#2,en,en$,t,s:ifenthenclose1:cl
ose2:gosub10070:goto500
540 input#1,lr,se,ae:close2:close1
550 ts=se+100:ta=ae+100
580 dims$(se+100),au$(ae+100),sl$(se+100
),sh$(se+100),al$(ae+100),ah$(ae+100)
590 ifse=1then680
600 open1,8,2,"stichwort,s,r":open2,8,15
:input#2,en,en$,t,s
610 ifenthenclose1:gosub10070:goto600
620 fori=1tose-1:input#1,s$(i)
625 gosub7300:sl$(i)=h$:sh$(i)=h1$
630 nexti:close2:close1
640 open1,8,2,"autor,s,r"
650 rem ifenthenclose1:gosub10070:goto6
40
660 fori=1toae-1:input#1,au$(i)
665 gosub7300:al$(i)=h$:ah$(i)=h1$
670 nexti:close1
680 print"§":gosub10030
700 !screen m$(0)
710 gosub10040
720 !imask "§§§§§§§§",s$,t1$,t2$,t3$,t4$,a
u$,v1$,v2$,o$,ja$,se$,b$,an,z1$,z2$
725 gosub10000
730 onpeek(2)goto740,1100,1000,1200,720,
2000,720
740 ifse>1>tsthenprint"stichwortdatei zu
gross":stop

```

Listing 4. »LIT.VERW« ist ein Demoprogramm zur Literaturverwaltung

```

750 gosub 7000: if len(sl$(ei+1)) > 255 then print "stichworttabelle zu gross": stop
760 if ge=1 then h1=i: goto 780
770 h1=ei+1: se=se+1: h3$="": h4$=""
772 for i=ei+1 to se: h$=s$(i): s$(i)=s$: s$=h$
773 h1$=sl$(i): sl$(i)=h3$: h3$=h1$: h2$=sh$(i): sh$(i)=h4$: h4$=h2$
775 next i
780 if ae+1 > t then print "autoren datei zu gross": stop
790 gosub 7100: if len(al$(ei+1)) > 255 then print "autoren tabellen ueberlauf": stop
800 if ge=1 then ei=i-1: goto 820
810 ae=ae+1: h3$="": h4$=""
812 for i=ei+1 to ae: h$=au$(i): au$(i)=au$: au$=h$
813 h1$=al$(i): al$(i)=h3$: h3$=h1$: h2$=ah$(i): ah$(i)=h4$: h4$=h2$: next
820 lr=lr+1: h=lr: gosub 7200
830 sh$(h1)=sh$(h1)+chr$(hb): sl$(h1)=sl$(h1)+chr$(lb)
840 ah$(ei+1)=ah$(ei+1)+chr$(hb): al$(ei+1)=al$(ei+1)+chr$(lb)
845 rem gosub 11000: lr=lr+1: goto 700
850 open 1,8,2,"literatur,1,"+chr$(202)
860 open 2,8,15
870 print #2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
880 h$=t1$+t2$+t3$+t4$+v1$+left$(v2$,10): print #1,h$
890 lr=lr+1: lb=lb+1: if lb=256 then lb=0: hb=hb+1
900 print #2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
910 h3$=right$(v2$,28)+o$
920 h$=str$(ja): gosub 7210: h1$=h$+left$(b1$, (4-len(h$)))
930 h$=str$(an): gosub 7210: h2$=h$+left$(b1$, (2-len(h$)))
940 h$=h3$+h1$+se$+b$+h2$+z1$+z2$+au$(ei+1)+s$(h1): print #1,h$
999 close 2: close 1: goto 700
1000 we=1: gosub 8000: if we=0 then 730
1010 goto 710
1100 open 4,4,7: gosub 10000: gosub 7900: close 4: goto 710
1200 sl=1: open 4,4,7: print #4,chr$(13)chr$(13)chr$(13)
1220 print #4,chr$(14); "STICHWOERTER"; chr$(15); chr$(13); chr$(13)chr$(13)
1230 for i=1 to se-1: for j=1 to 20: h$(j)=" ": next
1240 h$=s$(i): gosub 7210: h$(1)=h$+" ":
1250 z=1: for j=1 to len(sl$(i))
1260 h=asc(mid$(sl$(i),j,1)): h1=asc(mid$(sh$(i),j,1))
1270 h$=str$((h+h1*256)/2): gosub 7210: h$=h$+" ",
1280 if len(h$(z))+len(h$) > 79 then z=z+1
1290 h$(z)=h$(z)+h$: next
1300 h$(z)=left$(h$(z), (len(h$(z))-1))
1310 for j=1 to z: print #4,h$(j)
1315 next j
1320 next i
1330 print #4,chr$(13)chr$(13)chr$(13)
1340 print #4,chr$(14) "AUTOREN" chr$(15) chr$(13)chr$(13)chr$(13)

```

```

1350 for i=1 to ae-1: for j=1 to 20: h$(j)=" ": next
1360 h$=au$(i): gosub 7210: h$(1)=h$+" ":
1370 z=1: for j=1 to len(al$(i))
1380 h=asc(mid$(al$(i),j,1)): h1=asc(mid$(ah$(i),j,1))
1390 h$=str$((h+h1*256)/2): gosub 7210: h$=h$+" ",
1400 if len(h$(z))+len(h$) > 79 then z=z+1
1410 h$(z)=h$(z)+h$: next
1420 h$(z)=left$(h$(z), (len(h$(z))-1))
1430 for j=1 to z: print #4,h$(j)
1440 next j
1450 next i
1460 print #4,chr$(13)chr$(13)chr$(13)
1470 print #4,chr$(14) "LITERATURSTELLEN" chr$(15)chr$(13)chr$(13)chr$(13)
1480 for i=2 to lr-1 step 2
1490 h=i: gosub 7200: gosub 9000: gosub 7400
1510 h$=str$(i/2): gosub 7210: h$=" (" + h$ + " ) "
1520 h$=h$+left$(b1$, (9-len(h$)))+h$(1): print #4,h$
1530 for j=2 to z: print #4,spc(9); h$(j): next
1550 next i
1560 close 4: goto 710
2000 open 1,8,15,"s: stichwort": close 1
2010 open 1,8,2,"stichwort,s,w"
2020 for i=1 to se-1: h$=chr$(34)+s$(i)+chr$(34): print #1,h$
2030 for j=1 to len(sl$(i)): g=asc(mid$(sl$(i),j,1))
2040 print #1,g
2052 g=asc(mid$(sh$(i),j,1)): print #1,g: next
2054 print #1,"++": next i
2060 close 1
2070 open 1,8,15,"s: autor": close 1
2080 open 1,8,2,"autor,s,w"
2090 for i=1 to ae-1: h$=chr$(34)+au$(i)+chr$(34): print #1,h$
2100 for j=1 to len(al$(i)): g=asc(mid$(al$(i),j,1))
2110 print #1,g
2122 g=asc(mid$(ah$(i),j,1)): print #1,g: next
2130 print #1,"++": next i: close 1
2140 open 1,8,2,"literatur,1,"+chr$(202)
2150 open 2,8,15
2160 print #2,"p"+chr$(2)+chr$(1)+chr$(1)+chr$(1)
2170 h$=str$(lr)+chr$(13)+str$(se)+chr$(13)+str$(ae)
2180 print #1,h$
2200 close 2: close 1
2210 print " "; end
7000 li=se: ei=0: i=0: ge=0
7010 if ((li-ei)/2) < 1 then return: rem stichwort nicht vorhanden
7020 i=int((li-ei)/2)+ei
7030 if s$(i)=s$ then ge=1: return
7040 if s$ < s$(i) then li=i: goto 7010
7050 ei=i: goto 7010
7100 li=ae: ei=0: i=0: ge=0
7110 if ((li-ei)/2) < 1 then return: rem autor nicht vorhanden
7120 i=int((li-ei)/2)+ei
7130 if au$(i)=au$ then ge=1: return

```

Listing 4. »LIT.VERW.« (Fortsetzung)

```

7140 if au$(i) then li=i: goto 7110
7150 ei=i: goto 7110
7200 hb=int(h/256): lb=h-hb*256: return
7210 for ii=len(h$) to 1 step -1: if mid$(h$,ii,1)=" " then next
7220 h$=left$(h$,ii): for ii=1 to len(h$): if mid$(h$,ii,1)=" " then next
7230 h$=right$(h$, (len(h$)-ii+1)): return
7300 h1$="": h$="": for j=1 to 256: input #1, g$: if g$="++" then j=256: return
7305 if g$="0" then h$=h$+chr$(0): goto 7315
7310 h$=h$+chr$(val(g$))
7315 input #1, g$
7320 if g$="0" then h1$=h1$+chr$(0): goto 7340
7330 h1$=h1$+chr$(val(g$))
7340 next j: return
7400 for l=1 to 10: h$(l)="": next
7405 if au$=left$(b1$,25) then au$=""
7410 h$=au$: gosub 7210: h$(1)=h$+" "
7420 if t1$=left$(b1$,38) then t1$=""
7430 h$=t1$: gosub 7210: h$(2)=h$+" "
7440 if t2$=left$(b1$,38) then t2$=""
7450 h$=t2$: gosub 7210: h$(2)=h$(2)+h$+" "
7460 if t3$=left$(b1$,38) then t3$=""
7470 z=2: h$=t3$: gosub 7210: h$(3)=h$+" "
7480 if t4$=left$(b1$,38) then t4$=""
7490 h$=t4$: gosub 7210: h$(3)=h$(3)+h$+" "
7500 z=z+1: h$(z)=h$(z)+". "
7510 if b$="n" or b$="N" then b660
7520 z=z+1: if o$=left$(b1$,34) then o$=""
7530 h$=o$: gosub 7210: h$(z)=h$(z)+h$+" "
7540 if v2$=left$(b1$,38) then v2$="": goto 7560
7550 h$=v2$: gosub 7210
7560 if len(h$(z))+len(h$)+2>70 then z=z+1
7570 h$(z)=h$(z)+h$+" "
7580 if ja=0 then h$="": goto 7600
7590 h$=str$(ja): gosub 7210
7600 if len(h$(z))+len(h$)+2>70 then z=z+1
7610 h$(z)=h$(z)+h$+" "
7620 if se$=left$(b1$,10) then se$="": goto 7650
7630 h$=se$: gosub 7210
7640 if len(h$(z))+len(h$)+1>70 then z=z+1
7650 h$(z)=h$(z)+h$+" "
7660 z=z+1: if z1$=left$(b1$,38) then z1$=""
7670 h$=z1$: gosub 7210: h$(z)=h$(z)+h$+" "
7680 if z2$=left$(b1$,38) or h$="?" then h$=""
7690 h$=z2$: gosub 7210
7700 if len(h$(z))+len(h$)+1>70 then z=z+1
7710 h$(z)=h$(z)+h$+" "
7720 if an=0 then h$="": goto 7740
7730 h$=str$(an): gosub 7210
7740 if len(h$(z))+len(h$)+1>70 then z=z+1
7750 h$(z)=h$(z)+h$+" "
7760 if ja=0 then h$="": goto 7780
7770 h$=str$(ja): gosub 7210: h$=" (" +h$+" )"
7780 if len(h$(z))+len(h$)+1>70 then z=z+1
7790 h$(z)=h$(z)+h$+" "
7800 if se$=left$(b1$,10) then se$="": goto 7830
7810 h$=se$: gosub 7210
7820 if len(h$(z))+len(h$)+1>70 then z=z+1
7830 h$(z)=h$(z)+h$+" "
7900 gosub 7400: form=1 to z: print #4, h$(m): next
8000 h=0: h2=0
8005 if s$<>left$(b1$,21) then gosub 7000: if

```

```

ge=1 then h=1: h1=i
8010 if au$<>left$(b1$,25) then gosub 7100: i
fge=1 then h2=1: h3=i
8015 if h=0 and h2=0 then return
8020 if h2=1 or (h2=1 and h=1) then 8080
8030 for i=1 to len(s1$(h1)): lb=asc(mid$(s1$(h1),i,1))
8050 hb=asc(mid$(sh$(h1),i,1)): gosub 9000
8060 gosub 9300: if we=0 then return
8070 next i: return
8080 if h2=1 and h=1 then 8140
8090 for i=1 to len(al$(h3)): lb=asc(mid$(al$(h3),i,1))
8100 hb=asc(mid$(ah$(h3),i,1)): gosub 9000
8110 gosub 9300: if we=0 then return
8120 next i: return
8140 h1$=sh$(h1): h2$=ah$(h3): h4$=s1$(h1): h5$=al$(h3)
8150 if h1$>h2$ then h3$=h2$: h2$=h1$: h1$=h3$: h3$=h5$: h5$=h4$: h4$=h3$
8160 for i=1 to len(h1$): for l=1 to len(h2$)
8170 if mid$(h1$,i,1)<>mid$(h2$,l,1) then next l, i: return
8175 if mid$(h4$,i,1)<>mid$(h5$,l,1) then next l, i: return
8180 lb=asc(mid$(h4$,i,1)): hb=asc(mid$(h1$,i,1))
8190 gosub 9000: gosub 9300
8200 if we=0 then return
8210 next l, i: return
9000 open 1,8,2,"literatur,1,"+chr$(202)
9010 open 2,8,15
9020 print #2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
9030 for j=0 to 3: h$(j)=""
9040 for k=1 to 38: get #1, g$: h$(j)=h$(j)+g$: next: next
9050 t1$=h$(0): t2$=h$(1): t3$=h$(2): t4$=h$(3)
9060 v1$="": for j=0 to 37: get #1, g$: v1$=v1$+g$: next
9061 v2$="": for j=0 to 9: get #1, g$: v2$=v2$+g$: next
9062 lb=lb+1: if lb>255 then lb=0: hb=hb+1
9063 print #2,"p"+chr$(2)+chr$(lb)+chr$(hb)+chr$(1)
9070 for j=1 to 28: get #1, g$: v2$=v2$+g$: next
9080 o$="": for j=1 to 34: get #1, g$: o$=o$+g$: next
9090 h$="": for j=1 to 4: get #1, g$: h$=h$+g$: next: ja=val(h$)
9095 se$="": for j=1 to 10: get #1, g$: se$=se$+g$: next
9100 get #1, b$
9110 h$="": for j=1 to 2: get #1, g$: h$=h$+g$: next: an=val(h$)
9130 z1$="": for j=1 to 38: get #1, g$: z1$=z1$+g$: next
9140 z2$="": for j=1 to 38: get #1, g$: z2$=z2$+g$: next
9150 au$="": for j=1 to 25: get #1, g$: au$=au$+g$: next
9160 s$="": for j=1 to 21: get #1, g$: s$=s$+g$: next
9290 close 2: close 1: return
9300 gosub 10040
9305 !iomask "MATH", s$, t1$, t2$, t3$, t4$, au$, v1$, v2$, o$, ja, se$, b$, an, z1$, z2$
9310 if peek(2)=2 then open 4,4,7: gosub 7900: close 4: goto 9330

```

```

9320 ifpeek(2)<>5thenwe=0
9330 gosub 10000: return
9999 stop
10000 printleft$(cd$,24)bl$left$(bl$,38)
chr$(145)chr$(145);
10010 printleft$(cd$,24)spc(12)"SYSTEM A
RBEITET";
10020 return
10030 printleft$(cd$,24)bl$chr$(20)right

```

```

$(bl$,39)chr$(145)chr$(145);: return
10040 gosub 10030: printleft$(cd$,24)bl$;
: printleft$(cd$,25)b2$chr$(145)chr$(145)
;
10050 return
10060 !dier en,en$,t,s
10070 printen;en$;t;s: stop

```

ready.

Listing 4. »LIT.VERW« (Schluß)

programm : maske 0801 0d85

```

0801 : 0f 08 ca a8 9e 32 30 36 85
0809 : 35 20 46 43 43 00 00 00 7d
0811 : a0 00 b9 69 07 99 00 cd 26
0819 : b9 69 08 99 00 ce b9 69 ec
0821 : 09 99 00 cf c8 d0 eb 4c 4c
0829 : c2 cd 78 a0 ff 84 fb a9 6b
0831 : c6 85 fc a9 36 85 01 8d dd
0839 : 20 d0 c8 a5 2d d0 02 c6 97
0841 : 2e c6 2d a6 2e e0 09 d0 a2
0849 : 04 c9 bf f0 0f b1 2d 91 96
0851 : fb a5 fb d0 02 c6 fc c6 10
0859 : fb 4c d3 cd a2 08 a9 01 3c
0861 : 86 2e 85 2d 84 ff 20 50 6f
0869 : ce c9 f3 d0 27 20 50 ce 85
0871 : aa 86 fa c9 04 b0 04 a9 7f
0879 : f3 d0 03 20 50 ce a0 00 97
0881 : 91 2d c8 c6 fa d0 f9 98 03
0889 : 18 65 2d 85 2d d0 02 e6 7d
0891 : 2e 4c 34 ce a0 00 91 2d 77
0899 : e6 2d f0 f3 a9 0e a2 13 8c
09a1 : e4 2e d0 c2 c5 2d d0 be af
09a9 : a9 37 85 01 a9 fe 8d 20 78
09b1 : d0 58 20 59 a6 4c ae a7 b7
09b9 : a2 ff 86 f7 86 f8 e8 a9 22
09c1 : 01 85 fe a9 7f 85 fd c6 23
09c9 : ff 10 10 e6 fb d0 02 e6 cd
09d1 : fc a9 07 85 ff a0 00 b1 7d
09d9 : fb 85 f9 06 f9 b0 0a 4d 6d
09e1 : fe a5 fd 39 f7 00 99 f7 2e
09e9 : 00 8a 0a a8 a5 f7 38 f9 b5
09f1 : e2 ce a5 f8 f9 e3 ce 90 de
09f9 : 0e e0 0c f0 0a e8 38 66 2e
0901 : fd b0 c4 c6 fe 0b bc 8a 0e
0909 : f0 0f a5 f7 38 f9 e0 ce 5e
0911 : 85 f7 a5 f8 f9 e1 ce 85 0f
0919 : f8 a4 fe f0 07 a5 f8 85 ce
0921 : f7 88 84 f8 a5 fd 4a 90 31
0929 : 07 46 f8 66 f7 4c be ce d9
0931 : bd d2 ce 65 f7 a8 b9 00 63
0939 : cf 60 00 00 02 03 03 07 8b
0941 : 0e 13 20 2e 3a 4f 54 56 c3
0949 : 56 56 00 00 00 80 00 a0 10
0951 : 00 a0 00 c0 00 dc 00 e6 6e
0959 : 00 f3 00 fa 00 fd a0 ff 25
0961 : f0 ff 00 00 00 00 00 00 51
0969 : 21 20 f5 f0 05 85 b5 00 89
0971 : d5 65 5d 95 01 15 45 b0 32
0979 : 80 a5 60 10 a0 c0 14 90 67
0981 : 26 c5 08 12 40 e0 23 25 b0
0989 : 09 0e d0 e5 03 06 07 f3 ef
0991 : 0b 0c d0 13 84 a9 ff 02 e2
0999 : 04 0f 11 30 41 53 5a 0a 9b
09a1 : 17 19 1d 22 2c 31 32 36 52
09a9 : 42 4a 4c 4f 54 55 56 58 07
09b1 : 59 5b 5f 64 6a 9b 9e a3 61
09b9 : a6 bf c1 c7 de e2 f7 76 7a
09c1 : ff ef b3 fe 7e cf b7 6b b8
09c9 : fe bc 30 c3 e7 fa b9 fb df
09d1 : 6d fe 3e 5c be 7e 7e 7f b1
09d9 : 8d bf 0f 97 f7 f9 ff ff 4c
09e1 : 9f fd 5b fa fc bf b3 ff 52
09e9 : 9f f5 fc 3f df fd e1 ff 1f
09f1 : ef f3 f7 ce 99 e6 dd ea d0
09f9 : f2 ed 9f 95 fe b9 dd 6e 8e
0a01 : 98 cc ea 43 23 12 16 21 80
0a09 : 72 a5 8b 99 10 d0 86 25 50
0a11 : 48 68 54 84 ce 3d 0d 1e 7a
0a19 : 0f 07 bd a3 46 8d 1e 67 a8
0a21 : 83 47 99 a3 46 8f 43 e0 d2
0a29 : e0 e0 ef 68 d1 e6 70 77 87
0a31 : b8 3b de 86 8d 1a 3c 9d e5

```

```

0a39 : ed 1a 38 3c cd 1e 0f 27 21
0a41 : 07 06 8f 06 8d 3f 94 f3 fd
0a49 : 6d d3 6a db df 76 ac 63 e1
0a51 : 1b b8 b1 bb 56 31 8c 68 9e
0a59 : e0 e0 f3 be 86 8d 1e a7 1b
0a61 : bd a3 47 7b e8 68 d1 e6 18
0a69 : 70 70 77 bd 0d 1a 67 63 ad
0a71 : d2 5c c4 c4 b9 52 19 19 00
0a79 : 17 2c 58 fe 64 21 89 a1 55
0a81 : 91 73 42 10 86 e3 43 43 7a
0a89 : 62 c4 36 21 52 1b 16 2e b2
0a91 : 42 67 33 85 8a 90 c8 84 5e
0a99 : 2e 6e 34 36 2a 43 89 0d cf
0aa1 : 88 64 62 62 42 10 86 25 4f
0aa9 : 4b 17 2e 42 1e 92 c7 12 0d
0ab1 : 10 e2 42 67 33 b1 91 73 9e
0ab9 : b0 85 88 54 b1 91 0a 90 c9
0ac1 : 84 2a 5c 85 88 42 17 2a 6d
0ac9 : 54 86 25 48 42 a5 cc 8a 4c
0ad1 : 90 87 91 b8 be 73 3c 88 2a
0ad9 : 54 c4 84 2a 42 10 c8 a9 11
0ae1 : 08 42 17 36 21 c4 84 21 24
0ae9 : 08 6e 21 0c 8c 88 43 13 32
0af1 : 22 10 84 31 ce ae 0f 06 0d
0af9 : 8f 53 47 bd ef 7a 9e a7 58
0b01 : a9 a3 d4 d1 ea 79 9c 1a 0c
0b09 : 3c 1e 0f 43 47 06 8d 1f 9a
0b11 : 43 c9 c1 a3 46 8e 0f a1 76
0b19 : de f3 38 34 79 9a 3c ee c1
0b21 : 1f 76 5b 63 d3 5f af 77 a4
0b29 : 74 66 70 a9 52 c5 88 6e 74
0b31 : 21 62 19 17 2a 42 10 85 ad
0b39 : 4a 90 a9 08 42 e5 73 b3 bf
0b41 : b3 57 73 8b 1e c7 65 77 93
0b49 : b8 34 68 d1 e4 d1 a3 47 6a
0b51 : 06 8d 1d ed 1c 1d ee 18
0b59 : 0d 1f 43 d0 d1 e7 78 3b 95
0b61 : da 34 79 de 0d 1e 66 8f 0a
0b69 : 83 46 8f 07 7b c9 a7 f4 63
0b71 : 95 dd dd 5f 7d 75 9e c8 e8
0b79 : c6 cd 98 c6 31 8d 98 e4 d0
0b81 : c6 31 d1 e2 e4 c6 31 c5 85
0b89 : 8d 5b b1 8c 63 1b b1 ab ef
0b91 : 19 9c 21 c4 c4 84 2a 42 77
0b99 : 18 9b 10 84 2a 42 10 b1 6c
0ba1 : 89 08 54 d0 86 e3 12 a4 77
0ba9 : 21 52 17 21 53 12 c5 cc 54
0bb1 : 73 99 c3 42 c5 4b 17 31 9f
0bb9 : 31 21 e9 2c 62 5c 87 12 c6
0bc1 : 15 2e 54 a9 0b 10 86 e2 49
0bc9 : 1e 92 c5 a9 99 10 86 86 4c
0bd1 : e3 43 43 c8 be 75 68 e0 3b
0bd9 : f3 b4 70 70 7c 1c 1a 3e de

```

```

0be1 : 0d 1e 77 a1 a3 bd c1 ef 1e
0be9 : 68 f2 78 3d ee f7 93 ce 2b
0bf1 : d1 ef 77 b4 79 9a 34 7a 60
0bf9 : 1a 3e 0d 1a 34 78 38 38 11
0c01 : 7f 1b 77 4e 98 e5 9c 2a 35
0c09 : 79 1c 48 42 a4 2a 68 71 0b
0c11 : 21 c4 d8 a9 73 b0 ec 21 b2
0c19 : 08 76 16 34 32 21 08 43 3b
0c21 : 12 c4 32 32 26 7a 34 70 50
0c29 : 7e d7 7b 47 83 46 8f 27 52
0c31 : a9 ea 68 d1 de d1 a3 47 3d
0c39 : 07 7b c9 a3 46 8d 1a 34 86
0c41 : 68 d1 de d1 a3 46 8f 43 b5
0c49 : 46 8f 43 46 8d 3f 7e 3a 32
0c51 : db b7 2e e8 cc e7 8d 7c ec
0c59 : 67 8c f1 cb 38 c7 47 b3 42
0c61 : ef d2 b5 f3 4e 9c 6b 19 4f
0c69 : 9d 4b 19 10 84 21 91 52 30
0c71 : 13 38 e8 c6 37 7b 1d ce 15
0c79 : 2d ff 76 5b 57 d5 8f 6d ec
0c81 : dd d9 ee 34 ce 3c 9a 38 37
0c89 : 38 3b df 0f ba bb 57 e3 e7
0c91 : 3d 93 59 fa ca cf 3d b6 1a
0c99 : ec f8 6d f0 95 9f d1 db c0
0ca1 : 3b 78 e5 e3 96 71 a3 46 1e
0ca9 : 8e 0d 1e 4f 27 c1 8c 78 d3
0cb1 : b8 bb 9b 31 8c 63 56 3e 0e
0cb9 : 4e e7 47 46 31 8d d8 c6 06
0cc1 : cd d8 c6 35 71 6c f1 6c 6e
0cc9 : c6 31 c5 c5 ab 77 2f c2 0b
0cd1 : 56 79 ab d3 3e 33 d5 3b 94
0cd9 : 71 ee bf 9a de bd 3a 65 94
0ce1 : 67 9e 79 e6 b5 f8 5e b1 d2
0ce9 : 8c 6a e4 ee 6e f1 63 18 f5
0cf1 : e4 c6 ac 63 1b b1 d9 c5 02
0cf9 : 8c 66 79 10 84 2c 62 5c 05
0d01 : b1 73 62 10 dc 6e 2e 71 e3
0d09 : 21 0d c5 cc 4a 90 84 21 39
0d11 : 08 62 54 c4 85 8b 90 85 fa
0d19 : 88 4c ed 9d c8 42 1c 4b 9c
0d21 : 96 2e 42 1e 44 2e 54 b1 8d
0d29 : 73 22 e4 21 89 0b 1a 1a 98
0d31 : 10 84 38 9a 1b 8d 0e 24 83
0d39 : 2e 71 2a 42 b9 fa 70 f7 18
0d41 : 6b b7 ab 5d 7a fe ad 75 60
0d49 : eb ae ba f5 fa f5 d7 af 17
0d51 : b7 d9 af e7 ee d7 5e bf 84
0d59 : 7e ba f5 f9 eb af e9 e6 a2
0d61 : f5 e1 f5 73 6d af e3 f6 04
0d69 : 7a f5 d7 ec fd 18 7d 3d 82
0d71 : 3a eb d7 b6 ba f5 fb 7f b8
0d79 : fb 87 b7 e8 e1 ed fa 38 2d
0d81 : 7b 7e 8e 00 ff 85 b8 85 f9

```

Listing 5.  
Hilfsprogramm  
»Maske«

```

10 A=6*4096: IFPEEK(A)<>222THENPRINT"ZUER
ST 'MASKE' LADEN !!!":END
20 OPEN#8,2,"ZEITSCHRIFT,S,W"
30 PRINT#2,CHR$(PEEK(A));A=A+1: IFA<=273
61THEN30
40 CLOSE2

```

READY.

Listing 6. Das Programm »SEQ.SAVE« generiert aus  
dem Hilfsprogramm »Maske« das sequentielle File  
»Zeitschrift«.