USERCLUB MC-FREUNDE KÖLN KALKER HAUPTSTRASSE 189 5000 KÖLN 91

EEEEEEEEE RRRRRRRR RRRRRRRR 00000000 RRRRRRRR
EEEEEEEEE RRR RR RRR RR 0000000000 RRR RR
EE RR RR RR RR 00 00 RR RR
EE RR RR RR RR 00 00 RR RR
EEEEEEEEE RRRRRRR RRRRRRR 00 00 RRRRRRR
EEEEEEEEE RRRRRR RRRRRR 00 00 RRRRRR
EE RR RR RR RR 00 00 RR RR
EE RR RR RR RR 00 00 RR RR
EEEEEEEEE RR RR RR RR 0000000000 RR RR
EEEEEEEEE RR RR RR RR 00000000 RR RR

CLUBZEITUNG

NUMMER 3 DEZEMBER 1983

WIR WÜNSCHEN

ALLEN UNSEREN MITGLIEDERN

EIN FROHES WEIHNACHTSFEST

UND EIN KRÄFTIGES

P R O S I T N E U J A H R

ilgo. 1983

ERROR Sprachrohr des

USERCLUB MC-FREUNDE KÖLN

Unabhängiger Verein für TRS-80 + GENIE Anwender

Ausgabe 3 DEZEMBER 83

IMPRESSUM:

Herausgeber	USERCLUB MC- FREUNDE KÖLN Kalker Hauptstraße 189-191 5000 KÖLN 91 ☎ 02204/65254
Redaktion	L. Drossert
Gestaltung	L. Drossert + R. Schröder
Druck	USERCLUB MC-FREUNDE KÖLN

VERANTWORTLICH:

Kurse	L. Drossert
Tauschbörse	R. Schröder
Anzeigen	L. Drossert

W I R Ü B E R U N S

Mit knapper Mühe und Not noch vor Weihnachten geschafft, aber er ist da der ERROR NR 3. Etwas dicker wie die Nummer 2, was er der regen Mitarbeit unseres Mitglieds Jürgen Kuschel zu danken hat, aber auch in etwas neuem Gewande (doppelseitig geheftet), was wir ebenfalls Herrn Kuschel verdanken, welcher uns freundlicherweise seinen DIN a 3 Copierer hierfür zu Verfügung stellte. Soweit die guten Nachrichten, richtig gelesen, das war's da schon. Die schlechten, die jetzt Kommen, sind leider bei weitem umfangreicher.

Leute, wir sind ernsthaft sauer! An einer Eintragung ins Vereinsregister scheint Euch ja wohl nicht viel gelegen zu sein. Bei der hierfür angekündigten Hauptversammlung waren gerade 6 müde Krieger vertreten. 7 Unterschriften von Anwesenden sind aber hierzu nun mal erforderlich. Entweder ist Euch da wirklich nicht dran gelegen, oder Ihr befürchtet alle 'imaginäre' Verpflichtungen einzugehen? Daher hier einige Erläuterungen: Zur Eintragung ins Vereinsregister ist eine Gründungsversammlung erforderlich bei der 7 Anwesende; ihr Interesse an der Gründung eines eingetragenen Vereines mit der zur Disskussion stehenden Satzung bekunden. Diese 7 Personen sollten Mitglied sein oder werden. Sie müssen Ihre Unterschrift notariell beglaubigen lassen, die Kosten trägt der Verein. Die Gründungsmitglieder sind in keiner Weise verpflichtet, irgendwelche Ämter zu bekleiden oder zu übernehmen, ebenso brauchen sie in keiner Form für den Verein, oder dessen wirtschaftliche Grundlage zu haften. Da dürfte es doch wirklich nicht schwer sein, hier endlich zu Rande zu kommen.

Wir starten nun unverdrossen und unverzagt wie wir sind, den dritten Anlauf und setzen die neue "GRÜNDUNGS"
Samstag den 25. Februar 1984 15.oo Uhr an. Natürlich bei uns im Club. Hört sich gut an, Gründungsversammlung bei einem Verein, der fast ein Jahr besteht. Wir bitten diesmal aber um ein zahlreicheres Erscheinen. Die neue Gründungsversammlung ist zugleich Hauptversammlung, welche künftig vier mal jährlich, zu festen Terminen abgehalten wird. Die Termine sind vierteljährlich am jeweils letzten Samstag der Monate Februar, Mai, August und November, werden aber trotzdem vorher bekannt gegeben.

Aber nicht nur bei der Hauptversammlung fehlt es an reger Beteiligung, sondern auch die regelmäßigen Club-Treffen sind sehr schwach besucht. Wahrscheinlich weil einfach zu oft. Wir haben deshalb ab 1984 eine neue Regelung festgelegt. Um die Unkosten zu senken haben wir den Clubraum aufgegeben und die Treffen in einen Raum in meiner Wohnung, bei der selben Adresse, verlegt. CLUB-Treffen sind nicht mehr Mitwoch und Samstag, sondern nur noch 2 mal im Monat. Jeweils am 2. und 4. Samstag ab 13.oo Uhr. Ausnahmen hiervon gibt es nur im Juni und im Dezember 1984, wo die Treffen wegen der Feiertage dann um eine Woche verschoben sind. Die jeweils genauen Daten könnt Ihr unserem Kalender entnehmen, sie werden aber auch regelmäßig hier veröffentlicht.

Leider hat sich ja nun herausgestellt, das der "ERROR" nicht alle 2 sondern alle 3 Monate erscheint. Aber ganz einfach, wenn Ihr nicht schreibt, schreiben wir auch nicht!!! Wie schon beim vorigen Mal erwähnt, sind wir absolut keine 'Profis' und müssen uns unsere Artikel mühesam zusammenklauben. Zumindestens könnte etwas mehr Resonanz auf das kommen, was wir schreiben. Aber wir wären ja schon froh, wenn wir wenigstens erfahren würden, was Ihr lesen wollt. Schreibt doch wenigstens mal welche Themen Ihr lesen wollt, und wo Ihr Probleme habt. Die nötigen Artikel be-

Kommen wir dann schon zusammen. Noch lieber wäre es uns ja, Ihr würdet auch mal diesen oder jenen Artikel schreiben. Nur keine Angst, die anderen wissen auch nicht alles und jeder macht bestimmt diese oder jene Entdeckung und/oder Erfahrung an welcher der Rest der Menschheit nicht ganz uninteressiert ist.

Eigentlich sollte ich ja mit einer Seite auskommen, aber nun ist es doch mehr geworden. Das gibt mir Gelegenheit noch einiges zu den Artikeln zu sagen, erschienene und nicht erschienene. Unsere letzte Knobelaufgabe hat leider im Gegensatz zur ersten keine Resonanz gefunden. Wahrscheinlich ist sie falsch verstanden worden. Destotrotz war sie Anregung zu dem auf Seite 13 beginnenden Bericht über "INPUT", dabei liegt die Würze ja nun nicht gerade in der Kürze, wie bei unserer Knobelaufgabe verlangt, aber dafür hat der es in sich und zeugt von einiger Fachkenntnis. Bei dem Bericht über BASICODE ist mir beim vorigen mal ein Fehler unterlaufen, natürlich ist BASICODE nicht für Level II Basic entworfen, sondern für Basic allgemein. In der neuen Fortsetzung hat sich prompt wieder ein Fehler eingeschlichen. Ich habe vergessen, auf Seite 25 die Adresse von NOS-Hilversum anzugeben. Ein nachträgliches Einfügen war aus Platzmangel nicht möglich. Für Interessenten an dem Buch über BASICODE, hier also die genaue Adresse:

BASICODE
Administratie Algemeen Secretariaat NOS
P.O. Box 10
NL 1200 JB Hilversum

Der nun schon zweimal angekündigte Bericht über Partnerclubs fällt nun wohl doch ins Wasser, da hier leider keiner bereit ist, aktiv mitzuarbeiten. Alles kann ich leider nicht selbst erledigen. So ist die Kommunikation mit einigen interessierten Clubs im Moment eingeschlafen (Sie ruhe sanft). Aber ich hoffe, daß sie sich zu gegebener Zeit wieder aufwecken lässt.

Etwas erfreulicher ist da schon die Nachricht, daß die Programmbibliothek an Umfang zugenommen hat und wir jetzt schon etwas besser wissen, was wir überhaupt alles haben. Ein vorläufiger KATALOG wird Ende Januar fertig sein und Euch dann umgehend zugeschickt werden. Da steht dann alles drin, deshalb ist heute nichts über Bibliothek + Tauschbörse im ERROR. Ihr könnt dann alles Wichtige dem KATALOG entnehmen, der jedem Mitglied zugeschickt wird.

Am Schluss noch die genauen Daten der nächsten CLUB-Treffen.

JANUAR	FEBRUAR	MÄRZ
Samstag den 14.	Samstag den 11.	Samstag den 10.
Samstag den 28.	Samstag den 25.	Samstag den 24.

Jeweils ab 13.oo Uhr Hauptversammlung am 25. Feb. ab 15.oo Uhr.

In der Hoffnung daß Euch der neue ERROR gefällt und Ihr die beiligende Kassette gut geladen bekommt, wünscht Euch allen ein Frohes Fest und Prosit Neujahr,

Euer Leo Brossart

*** FKEY/CMD für GENIE II ***

Die Anregung zu diesem Programm bekamen wir auf den Dürener-Computertagen, wo gefragt wurde, warum beim GENIE II die F-tasten nicht, wie bei Genie III frei Belegbar sind. Unser recht aktives Mitglied (siehe Seite 13 "Gedanken zum Inputbefehl") Jürgen Kuschel hat sich daraufhin hingesetzt und ein erforderliches Maschinensprachprogramm entworfen. Nach Eingabe des Programms kann die F-Tastenbelegung mit SHIFT-CLAER jederzeit neu geändert werden. Aber warum viel schreiben, das Listing erklärt sich beim lesen und eintippen selbst, da es reichlich mit Kommentaren versehen ist.:

```
00010 ;FKEY/ASM  FUNKTIONSTASTEN-BELEGUNG FÜR GENIE II
00020 ;VER2E REV3E EDTASM-VERSION    10-12-83 MCW
00030 ;
00040 ;HIER LABELDEFINITIONEN ENTSPRECHEND DEM BENUTZTEN SYSTEM
00050 ;EINSETZEN FALLS DER BISHER VERWENDETE TASTATUR-TREIBER
00060 ;MIT DEN FUNKTIONS-TASTEN ANDERE ZEICHEN ERGIBT.
00070 ;
00080 F1      DEFL    'Ä'+32  ;ORIGINAL ASCII-CODES DER
00090 SF1     DEFL    'Ä'     ;FUNKTIONSTASTEN AUS BISHERIGEM
00100 F2      DEFL    'Ö'+32  ;TASTATUR-TREIBER
00110 SF2     DEFL    'Ö'
00120 F3      DEFL    'Ü'+32
00130 SF3     DEFL    'Ü'
00140 F4      DEFL    7EH
00150 SF4     DEFL    5EH
00160 ORGLOC DEFL     0FCD7H  ;LADEADRESSE
00170 ;LADEADRESSE IM PRINZIP BELIEBIG, HIER AUF MAXIMALWERT
00180 ;GESETZT. VORSICHT BEI ERWEITERUNG DES PROGRAMMS, EDTASM
00190 ;ERKENNT KEINEN FEHLER, WENN DAS PROGRAMM ÜBER FFFFH
00200 ;HINAUSLÄUFT. LADEADRESSE ENTSPRECHEND TIEFER SETZEN.
00210 ;
00220 KEYDCB EQU      4016H   ;POINTER ZUM TASTATUR-TREIBER
00230 VIDC   EQU      033AH   ;PRINT ASC IN A MIT PUSH U. SCROLL
00240 INLINE EQU      05D9H   ;LINEINPUT-ROUTINE
00250 DEFKEY EQU      31      ;AUSLÖSER FÜR NEUDEFINITION
00260 DEFLEN EQU      65      ;MAXIMALE LÄNGE DER FUNKTIONS-
00270 ;TASTENSTRINGS +1. BEI ÄNDERUNG VON ORGLOC KANN HIERFÜR
00280 ;BEI BEDARF AUCH MEHR PLATZ RESERVIERT WERDEN
00290 ;
00300 SHIFT  EQU      3880H   ;MATRIXADRESSE DER SHIFT-TASTEN
00310 HIBAS  EQU      40B1H   ;TOP-MEM POINTER FÜR BASIC
00320 HIDOS  EQU      4049H   ;TOP-MEM POINTER FÜR DOS
00330 KEY    EQU      0       ;DUMMY TREIBER-ADRESSE
00340 ;
00350 ;BEI CASSETTENSYSTEMEN DIESE 2 ZEILEN MITASSEMBLIEREN
00360 ;       ORG     41E2H   ;SYSTEM-VECTOR AUF AUTOSTART
00370 ;       JP      ORGLOC  ;ADRESSE SETZEN
00380 ;
00390         ORG     ORGLOC
00400 ;
00410 ;INITIALISIERUNGS-ROUTINE WIRD NUR EINMAL DURCHLAUFEN UND
00420 ;KANN ANSCHLIESSEND ÜBERSCHRIEBEN WERDEN.
00430 ;
00440 ;DIESE 2 ZEILEN EBENFALLS NUR FÜR CASSETTENSYSTEME
00450 ;WICHTIG: HIERFÜR ORGLOC 5 BYTE TIEFER SETZEN!
```

```
00460 ;       LD      A,0C9H          ;RETURN-OPCODE ZURÜCK AUF
00470 ;       LD      (41E2H),A       ;SYSTEM-VECTOR
00480 ;
00490         LD      HL,(KEYDCB)     ;ORIGINAL TASTATUR-TREIBER
00500         LD      (GETKEY+1),HL   ;INS PROGRAMM EINKLINKEN
00510         LD      (WDEF+1),HL
00520         LD      HL,FKEY         ;NEUE TREIBER-ADRESSE
00530         LD      (KEYDCB),HL     ;IN DCB EINTRAGEN
00540         DEC     HL              ;SICHERUNGSADRESSE
00550         DEC     HL
00560         EX      DE,HL           ;IN DE
00570         LD      HL,(HIBAS)      ;TESTEN OB HIBAS<ORGLOC
00580         OR      A               ;CARRY LÖSCHEN
00590         SBC     HL,DE
00600         JR      C,MTST          ;HIBAS KLEINER, OK
00610         LD      (HIBAS),DE      ;SONST NEU SETZEN
00620 MTST    LD      HL,(HIDOS)      ;DITO MIT HIMEM FÜR DOS
00630         OR      A
00640         SBC     HL,DE
00650         JP      C,VIDTBL
00660         LD      (HIDOS),DE
00670         JP      VIDTBL          ;LISTENAUSGABE
00680 ;
00690 ;START DER EIGENTLICHEN FUNKTIONSTASTEN-ROUTINE
00700 ;
00710 FKEY    PUSH    HL
00720         PUSH    DE
00730         PUSH    BC
00740         LD      A,(SFLAG)       ;F-TASTE ACTIV?
00750         OR      A
00760         JR      NZ,SUBST        ;DANN STRING-AUSGABE
00770         LD      A,(4022H)       ;CURSOR-FLAG
00780         LD      (CFLAG),A       ;AUFBEWAHREN
00790 GETKEY  CALL    KEY             ;SONST NEUE TASTE HOLEN
00800         JR      Z,RETKEY        ;KEINE GEDRÜCKT
00810         CP      DEFKEY          ;CLEAR-TASTE?
00820         JR      Z,DEFINE        ;NEUDEFINITION ERWÜNSCHT
00830         CALL    TSTKEY          ;RETURN NUR WENN F-TASTE
00840         PUSH    AF
00850         LD      A,(MFLAG)       ;TEST MODUS
00860         OR      A
00870         POP     BC
00880         LD      A,B             ;POP A OHNE F
00890         JR      NZ,RETKEY       ;UMLAUT MODUS
00900 SUBST   LD      HL,(SPOS)       ;POINTER ZUR LFD. POSITION
00910         LD      A,(HL)          ;IM FUNKTIONS-STRING
00920         INC     HL              ;NEUE POSITION
00930         LD      (SPOS),HL       ;SICHERN
00940         LD      (SFLAG),A       ;0 WENN STRING ZU ENDE
00950 RETKEY  OR      A               ;NZ FÜR DIESES ZEICHEN
00960         PUSH    AF
00970         LD      A,(CFLAG)       ;CURSOR-FLAG
00980         LD      (4022H),A       ;WIE GEHABT
00990         POP     AF
01000         POP     BC
01010         POP     DE
01020         POP     HL
01030         RET                     ;OK, DAS WAR'S
01040 ;
01050 ;FÜR NEUDEFINITION MUSS NACH SHIFT-CLEAR DIE NEU ZU
01060 ;BELEGENDE FUNKTIONSTASTE GEDRÜCKT WERDEN, DARAUF ER-
```

```
01070 ;SCHEINT DER BISHER FÜR DIESE TASTE DEFINIERTE STRING AUF
01080 ;DEM BILDSCHIRM. ANSCHLIESSEND KANN DER NEUE STRING EIN-
01090 ;GEGEBEN WERDEN. EINGABE-ENDE MIT NEW-LINE SETZT CARRIAGE
01100 ;RETURN ANS ENDE DES FUNKTIONS-STRING, EINGABEENDE MIT
01110 ;BREAK LÄSST DEN FUNKTIONS-STRING OHNE CR OFFEN FÜR
01120 ;WEITERE MANUELLE EINGABEN.
01130 ;
01140 ;MIT SHIFT-CLEAR UND ? ALS ANTWORT AUF DAS PROMPT 'F?'
01150 ;KANN JEDERZEIT EINE LISTE DER AKTUELLEN DEFINITIONEN AB-
01160 ;GEFRAGT WERDEN, WOBEI DER CURSORBLOCK ALS ZEICHEN FÜR
01170 ;CARRIAGE-RETURN AM ENDE EINES STRING AUSGEGEBEN WIRD.
01180 ;DAMIT TROTZ FUNKTIONSBELEGUNG DIE UMLAUTE AUF DEN
01190 ;FUNKTIONSTASTEN ERHALTEN BLEIBEN, KANN MIT CLEAR OHNE
01200 ;SHIFT ZWISCHEN UMLAUTEN UND FUNKTIONS-STRINGS UMGE-
01210 ;SCHALTET WERDEN. NORMALE FUNKTION DER CLEAR-TASTE WIRD
01220 ;DURCH 2-MALIGE BETÄTIGUNG (MIT SHIFT) ERREICHT.
01230 ;
01240 ;NEUDEFINITIONEN DER FUNKTIONS-STRINGS WERDEN HIER NICHT
01250 ;DAUERHAFT GESPEICHERT SONDERN BLEIBEN NUR FÜR DIE DAUER
01260 ;DES PROGRAMMLAUFS AKTIV. DA DIE MAXIMALE STRINGLÄNGE
01270 ;RESERVIERT IST UND DIE ADRESSEN DER STRINGS DAHER UNVER-
01280 ;ÄNDERT BLEIBEN, KÖNNEN DAUERHAFTE NEUDEFINITIONEN AUCH
01290 ;OHNE NEU-ASSEMBLIEREN MIT SUPERZAP O.Ä. INS FERTIGE
01300 ;PROGRAMM-FILE EINGETRAGEN WERDEN. DABEI ABER DAS 0-BYTE
01310 ;ALS TERMINATOR HINTER DEM STRING NICHT VERGESSEN!
01320 ;DIESE PROZEDUR KANN DURCH LADEN DESS PROGRAMMS MIT
01330 ;ANSCHLIESSENDEM DUMP AUS DEM SPEICHER NOCH VEREINFACHT
01340 ;WERDEN. BEI DEM SPEICHERDUMP STEHEN DIE STRINGS IM
01350 ;RICHTIGEN ABSTAND IM PROGRAMM-FILE, WÄHREND EDTASM BEI
01360 ;DEN DEFS-STATEMENTS JEWEILS NEUE BLOCK-HEADER MIT DEN
01370 ;ENTSPRECHENDEN LADEADRESSEN INS FILE EINTRÄGT.
01380 ;
01390 DEFINE LD      A,(SHIFT)       ;SHIFT-CLEAR?
01400        OR      A
01410        JR      NZ,DODEF        ;MIT, ALSO NEU-DEFINITION
01420        LD      A,(MFLAG)
01430        XOR     1               ;SONST MODUS UMSCHALTEN
01440        LD      (MFLAG),A
01450        XOR     A               ;'KEINE TASTE' SIMULIEREN
01460        JR      RETKEY
01470 ;
01480 DODEF  LD      HL,PROMPT       ;EINGABE ANFORDERN
01490        LD      B,9             ;TAB
01500        CALL    VIDMSG
01510 WDEF   CALL    KEY             ;WARTEN AUF F-TASTE FÜR
01520        JR      Z,WDEF          ;NEUDEFINITION
01530        CP      '?'
01540        CALL    Z,VIDTBL        ;LISTE ANGEFORDERT
01550        CALL    TSTKEY          ;RET HIER WENN F-TASTE
01560        CALL    VIDMSG          ;AUSGABE DES BISHER
01570                                ;DEFINIERTEN STRINGS
01580        LD      HL,(WDEF+1)     ;ALTEN KEY-DRIVER FÜR
01590        LD      (KEYDCB),HL     ;LINEINPUT ZURÜCKSETZEN
01600        LD      HL,(SPOS)       ;JETZT BUFFER FÜR INPUT
01610        LD      B,DEFLEN-1      ;MAXIMALE EINGABELÄNGE
01620        CALL    INLINE          ;LINEINPUT
01630        PUSH    AF              ;CARRY WENN BREAK
01640        LD      C,B             ;ANZAHL ZEICHEN
01650        LD      B,0
01660        ADD     HL,BC           ;POINTER AUF EINGABE-ENDE
01670        POP     AF
```

```
01680          JR       C,SET0            ;BREAK GEDRÜCKT, ALSO
01690                                     ;STRING OHNE CR GEWÜNSCHT
01700          LD       (HL),13           ;SONST CR HINTER STRING
01710          INC      HL                ;EINTRAGEN
01720 SET0     LD       (HL),0            ;TRAILERBYTE SETZEN
01730          LD       HL,FKEY           ;NEUE TREIBER-ADRESSE
01740          LD       (KEYDCB),HL       ;WIEDER IN DCB EINTRAGEN
01750          LD       A,24              ;SHIFT-BACKSPACE SIMUL.
01760 ;DIE DEFINITIONS-ROUTINE BENUTZT DIE INPUT-ROUTINE DES
01770 ;BASIC-INTERPRETERS, ÜBERSCHREIBT ALSO DEN INHALT DES
01780 ;KEYBOARD WORKSPACE FALLS IM LAUFENDEN BASIC-PROGRAMM
01790 ;MITTEN IN EINEM INPUT ODER LINEINPUT AUFGERUFEN.
01800          JR       RETKEY
01810 ;
01820 TSTKEY   LD       HL,FTBL           ;TABELLE F-TASTEN
01830          LD       B,(HL)            ;ANZAHL EINTRÄGE
01840          INC      HL                ;POINTER AUF 1. EINTRAG
01850          LD       DE,DEFLEN+2       ;OFFSET ZUM NÄCHSTEN
01860 FCMP     CP       (HL)              ;VERGLEICH MIT TASTE
01870          JR       Z,FOUND
01880          ADD      HL,DE             ;POINTER NÄCHSTEN EINTRAG
01890          DJNZ     FCMP              ;BIS TABELLEN-ENDE
01900          POP      HL                ;RETURN-ADRESSE VERGESSEN
01910          JR       RETKEY            ;KEINE FUNKTIONSTASTE GE-
01920 ;FUNDEN, ALSO ASCII-WERT UNVERÄNDERT ÜBERGEBEN.
01930 ;
01940 FOUND    INC      HL                ;POINTER ZUR STRING-ADR.
01950          LD       (SPOS),HL         ;SICHERN
01960          RET
01970 ;
01980 VIDMSG   LD       A,(HL)            ;BYTE FÜR VIDEO-AUSGABE
01990          OR       A                 ;ENDE WENN 0
02000          JR       Z,VCT
02010          CP       13
02020          JR       NZ,VCC
02030          LD       A,95              ;ZEICHEN FÜR CR
02040 VCC      CALL     VIDC              ;SONST AUSGABE
02050          INC      HL
02060          JR       VIDMSG
02070 VCT      LD       A,32              ;SPACE HINTERHER
02080          JP       VIDC              ;RETURN VON DORT
02090 ;
02100 VIDTBL   LD       HL,FTBL           ;STRING-TABELLE
02110          LD       B,(HL)            ;ANZAHL EINTRÄGE
02120          INC      HL
02130          LD       DE,DEFLEN+1       ;OFFSET ZUM NÄCHSTEN
02140 VTBL     LD       A,(HL)            ;ORIGINAL-ASCII
02150          INC      HL                ;POINT STRING
02160          CALL     VIDC
02170          LD       A,'='
02180          CALL     VIDC
02190          PUSH     HL                ;STRING-ADRESSE
02200          CALL     VIDMSG            ;STRING AUSGABE
02210          LD       A,10              ;CR
02220          CALL     VIDC
02230          POP      HL                ;LETZTE STRING-ADRESSE
02240          ADD      HL,DE             ;POINTER ZUM NÄCHSTEN
02250          DJNZ     VTBL
02260          LD       A,24              ;SHIFT-BACKSPACE
02270          RET
02280 ;
```

```
02290 SFLAG  DEFB    0        ;FLAG FÜR F-TASTE ACTIV
02300 MFLAG  DEFB    0        ;FLAG FÜR SUBST.MODUS
02310 CFLAG  DEFB    0        ;CURSOR ON/OFF-FLAG
02320 SPOS   DEFW    0        ;POSITION IM F-STRING
02330 PROMPT DEFB    10               ;CR
02340        DEFM    'F?'
02350        DEFW    14               ;CURSOR AN
02360 ;
02370 ;BEI ERWEITERUNG DIESER TABELLE UND EINTRAG DER ZUGE-
02380 ;HÖRIGEN STRINGS KANN PRINZIPIELL JEDE BELIEBIGE TASTE
02390 ;MIT JEDEM BELIEBIGEN STRING BELEGT WERDEN, ZUM BEISPIEL
02400 ;KLEINBUCHSTABEN MIT BASIC-STATEMENTS. AUCH EINZELNE
02410 ;ZEICHEN KÖMNEN ALS FUNKTIONS-STRINGS VORGEGEBEN WERDEN
02420 ;Z.B. UM AUF DER TASTATUR Z UND Y ZU VERTAUSCHEN.
02430 ;AUF JEDEN FALL MUSS DIE ANZAHL DER TABELLENEINTRÄGE
02440 ;ALS ERSTES BYTE IN DER TABELLE VORGEGEBEN WERDEN,
02450 ;HINTER JEDEM STRING EIN 0-BYTE ALS TERMINATOR STEHEN
02460 ;UND MIT DEFS DEFLEN MINUS LÄNGE DES FUNKTIONS-STRINGS
02470 ;PLATZ BIS ZUM NÄCHSTEN TABELLENEINTRAG RESERVIERT
02480 ;WERDEN.
02490 ;
02500 ;HIER STARTVERSION FÜR FUNKTIONS-STRINGS NACH BEDARF
02510 ;EINSETZEN, LÄNGENBYTE IN DEFS-STATEMENT NICHT VERGESSEN!
02520 ;
02530 FTBL   DEFB    8                ;8 EINTRÄGE
02540        DEFB    F1
02550        DEFM    'BASIC'
02560        DEFW    13
02570        DEFS    DEFLEN-6
02580        DEFB    F2
02590        DEFM    'RUN'
02600        DEFW    13
02610        DEFS    DEFLEN-4
02620        DEFB    F3
02630        DEFM    'EDIT'
02640        DEFB    0
02650        DEFS    DEFLEN-4
02660        DEFB    F4
02670        DEFM    'LIST'
02680        DEFB    0
02690        DEFS    DEFLEN-4
02700        DEFB    SF1
02710        DEFM    'DIR :0'
02720        DEFW    13
02730        DEFS    DEFLEN-7
02740        DEFB    SF2
02750        DEFM    'DIR :1'
02760        DEFW    13
02770        DEFS    DEFLEN-7
02780        DEFB    SF3
02790        DEFM    'MENU'
02800        DEFW    13
02810        DEFS    DEFLEN-5
02820        DEFB    SF4
02830        DEFM    'CMD"S"'
02840        DEFW    13
02850        DEFS    DEFLEN-7
02860 ;
02870 LAST   DEFB    0                ;DUMMY, LETZTES BYTE
02880 ;
02890        END     ORGLOC           ;STARTADRESSE NUR FÜR DOS
```

Zu Unserem letzten Tip von Herrn Langguth hier eine Ergänzung von unserem Mitglied Richard Rensch aus Laufen am Neckar.

32-Charakter-Mode:

Diesen kann man (typografisch) auch S P E R R - S C H R I F T nennen. Das Gegenteil heißt "kompreß". Um von "gesperrt" wieder in "kompreß" zu kommen, ist der in ERROR Nr.2 angegebene POKE in Adr. 403D H natürlich schon eine große Erleichterung. Immerhin muß man aber 11 Tastenanschläge eingeben, man muß darüberhinaus die Adresse im Kopf behalten oder nachschlagen und (SHIFT → funktioniert ja auch im DOS-Modus) man muß ins BASIC. Noch einfacher geht es mit einem kleinen Maschienenprogramm:

```
LD        HL,   403D H     ;POKE-Adresse in HL
LD        (HL),   00       ;00 in 403D H schreiben
RET                        ;zurück.
```

Dieses Prográmmlein kann man beim Genie III in einer freien Zeile des DOS ablegen (über EDTASM z.B. mit ORG 4490 H) oder bei Genie I & II an irgendeiner zugänglichen Stelle z.B. bei FFF0 H, und nennt es K/CMD (natürlich geht auch Eingeben, z.B.

```
  4490   21  38   40  36   00  C9
```

und Abspeichern mit DUMP). Dann braucht man zum Wechseln aus der Sperrschrift nach Kompreß nur "K" im DOS oder "CMD"K"" aus BASIC einzugeben.

AUS 1 MACH 2.

In der Regel lassen sich Disketten bei den meisten Geräten nur von einer Seite bespielen. Eine Sache, die man ganz einfach ändern kann. Disketten sind nicht wie Kassettenbänder nur einseitig mit einer Eisen-Ferror Beschichtung versehen, sondern von beiden Seiten. Daraus ergibt sich, daß auch die Rückseite formatiert werden kann.

Hierzu ist ein kleiner Eingriff erforderlich. Es muß genau gegenüber von dem vorhandenen ein zweites Indexloch (auf beiden Seiten) und eine zweite Schutzkerbe auf der anderen Seite angebracht werden. Schreibschutzkerbe und Indexloch die normalerweise rechts an der Diskette sind müssen also auch auf der linken Seite angebracht werden.

Als Hilfswerkzeug enmpfiehlt sich eine Hülle von einer defekten Diskette, aus der die Scheibe entfernt worden ist, Ein "EINLOCHLOCHER" aus dem Bürofachhandel, ein großer flacher Radiergummi und ein Stückchen selbstklebenden Filz.

Sie bekleben die Unterseite des Einlochlochers mit dem Filz (damit die Diskette nicht beschädigt wird, legen die leere Diskettenhülle auf die zu präparierende Diskette (Spiegelverkehrt) und zeichnen Indexloch und Schreibschutzkerbe mit weichem Filzstift genau an, dann schieben sie vorsichtig den Einlochlocher zwischen Diskette und Hülle und unterlegen das ganze auf der Disketten-unterseite mit dem Radiergummi. Jetzt können Sie vorsichtig Lochen und dann die Prozedur auf der Rückseite wiederholen. Die Schreibschutzkerbe lässt sich mit dem gleichen Locher anbringen. Gut die Hälfte der Lochbreite reicht normal als Kerbe.

Natürlich übernimmt der Hersteller auf die Rückseite keine Garantie, aber wir garantieren Euch, daß wir noch keinen Ausfall hatten. Lutz Grossert

INFOS

COMPUTERKURS für BASTLER! im FERNSEHEN (NDR/RB/SFB)

Ab Januar 1984 startet im dritten Fernsehprogramm der Nordkette (NDR/RB/SFB) ein Microcomputer-Kurs. Er besteht aus 26 Folgen a 15 Minuten; Im Lauf des Kurses soll ein Kleincomputer gebaut und programmiert werden. Dazu wird es ergänzend ein Angebot an Bausätzen geben. Der Kurs wird im ersten Teil den Aufbau eines Single-Board-Computers auf Z80A-Basis und seiner Anwendungsmöglichkeiten behandeln. Im zweiten Teil den Bau eines Systems auf 68008-Basis und eine Einführung in PASCAL. Die Sendung wird ab 2. Januar wöchentlich zwischen 19 Uhr und 19 Uhr 15 ausgestrahlt und ab 14. Januar zwischen 17 Uhr 30 (17 Uhr 45) und 17 Uhr 45 (18 Uhr) wiederholt. Der Kurs soll durch einen telefonischen Beratungsdienst und Begleitmaterial unterstützt werden.

Soweit Kommentarlos aus " C.P." Nr. 25 entnommen. Wir meinen das es sich hier um eine ausgesprochen interessante Sache handelt und würden uns wünschen, wenn sie nicht nur auf die Sender der "Nordkette" beschränkt wäre. Wir werden uns deshalb bemühen für alle die nicht aus diesem Sendebereich sind, eine VIDEO-Aufzeichnung anzufertigen und Sie dann allen Mitgliedern zugänglich zu machen. Wir suchen dringend noch jemanden im Sendegebiet, der bereit ist die Sendungen für uns aufzunehmen. Hier in Köln ist ein Empfang leider nicht mehr möglich.

Also; Bitte umgehend bei uns melden!

INFOS

Der sehr rührige holländische Computerclub "H C C" veranstaltet mit dem HOBBY COMPUTER CLUB afd. VENLO am Samstag den 14. Januar 1984 einen Computertag. Nach den eigenen Worten eine "Internationale Computershow für Hobby und täglichen Gebrauch für Jedermann. Wer Spaß hat mal wieder ein Wochenende in den Niederlanden zu verbringen dem können wir dieses Treffen von vorwiegend TRS-80 und GENIE Fans wärmstens empfehlen. Disketten nicht vergessen! Die Computertage des "H C C" sind immer auch eine recht rege Tauschbörse. Hier die Adresse:

Sa. 14. Jannuar 1984 11.00 - 19.00 UHR
COLLEGIUM MARIANUM, CASINOWEG 1, VENLO

INFOS

Wir haben eine Einladung von einer "USER CLUB ZENTRALE" erhalten, sie hält am 28.1.84 ab 11.oo Uhr in Frankfurt im Hotel CP PLAZA CITY ein USERCLUB-TREFFEN mit Infos über Sinn und Zweck der User-Club-Zentrale ab. Dieses Treffen soll bis 29.1.84 gegen Mittag gehen.

Leider war das, was uns darüber geschickt wurde sehr mager und wir hätten gerne gewusst was eine USERCLUB-ZENTRALE denn nun sein soll. Ob das wirklich eine Club-Sache ist, oder ob da Kommerz hintersteckt. Wenn einer von Euch der aus der Gegend ist da mal reinriechen könte um uns zu Berichten, wären wir nicht Böse. Vieleicht kann es ja auch ganz Interessant sein?

##################################

FÜR EUCH GELESEN

##################################

Heute wieder ein Buch von Rodny Zacks, "Programmieren des Z80". Ein Buch das Ideal ist für den, der sich intensiver mit der Programmierung in Z80-ASSEMBLER beschäftigen möchte. Es führt den Leser schrittweise in die Geheimnisse der Maschienensprache ein. Man lernt dabei langsam die Z80 zu programmieren und erlangt darüberhinaus ein besseres Verständnis für die Arbeitsweise eines Computers. Die einzelnen Kapitel sind Lehrgangsmäßig aufgebaut und immer nur so Umfangreich, daß man sie mühelos verstehen kann. Auch wenn man nicht in der Lage ist unheimlich viel Zeit mit Büffeln zu verbringen. Man kann das Buch ruhig einmal aus der Hand legen und das Thema ein paar Tage vergessen, ohne gleich den Faden zu verlieren.

Darüberhinaus ist es wie alle Bücher von Rodny Zacks in einer überaus gut verständlichen Sprache geschrieben, die bem Leser kein "Fachchinesisch" vorraussetzt, sondern ihn langsam darin einführt. Wie bei Fachbüchern nicht anders zu erwarten, ist es mit 49.-- DM absolut nicht billig, liefert aber auf gut 600 Seiten eine recht runde Fülle Information und Wissen.

Erschienen ist es beim SYBEX-Verlag Düsseldorf unter der Best.Nr. C 280 D. Wir meinen, ernsthafte Interessenten an der Z80 Maschinensprache werden auf Dauer um diese Lektüre nicht herum kommen.

HA HA HA

DA LACHT DIE CPU

Andropov fordert Reagan auf: "Laß doch mal euren Computer feststellen, wie ihr Amerikaner in 40 Jahren leben werdet!" Sekunden später antwortet der Computer: "SOZIALISTISCH". Andropov schüttelt sich vor lachen. Da verlangt Reagan: "Jetzt prüfe du mal, was dann mit euch Russen sein wird!" Andropov studiert die Antwortkarte und meint irritiert: "Das kann ich nicht lesen. Das ist chinesisch."

"Freulein Meier, laut Computer schreiben sie nur ganze 1,2759 Briefe pro Tag, machen pro Zeile 0,1176 Fehler und verbringen 3,478% Ihrer Arbeitszeit auf der Toilette! Was sagen sie dazu?"
"Setzen sie sich doch mal ihren Computer auf den Schoß Chef"

Und dann ist da noch der Supercomputer mit dem Kontrollämpchen oben rechts. Wenn es blinkt zeigt es an, daß die Firma pleite ist.

Streiten sich die zwei letzten menschlichen Beschäftigten in einem großen Konzern:
"Weg da! bei der letzten Störung vor 4 Wochen hast du gesagt, daß ich den nächsten Handgriff tun darf!"

PS: Etwaige Druckfehler in dieser Zeitung sind darauf zurückzuführen, daß der Satz mit Hilfe eines Computers hergestellt worden ist.

****
FUTTER für KNOBLER
****

Wie schon Eingangs erwähnt, ist zu unserer letzten Aufgabe das Kürzt mögliche Textverarbeitungsprogramm zu schreiben keine Lösung gekommen. Daher hier die Lösung die Marko Halber, von dem die Aufgabe stammte, uns gezeigt hat.

```
10 PRINTCHR$(14);INKEY$;:GOTO10
```

Kürzer geht es wohl kaum noch. Leider lässt sich der Spass aber nur im DOS-Handling mit JKL ausdrucken. Dementsprechend ist die neue Knobelaufgabe ganz darauf abgestellt. Hier ist sie:

↓↓

Wer schreibt zu obigem "TEXTVERARBEITUNGS-STRING" die Kürzeste Routine, die es ermöglicht das Geschriebene auch auszudrucken. Und wie gesagt, in der Kürze liegt die Würze! Ein Tip für Seitenfüller, es geht alles noch in Zeile 10. - Ran an den Speck.

↑↑

Wenn auch bei der Knobelaufgabe nicht viel rausgekommen ist, so hat sie doch den Erfolg gehabt, daß Jürgen Kuschel sich einige Gedanken über "INPUT" gemacht hat und sich darüber auf den nächsten "9" Seiten breit und Gründlich austobt. Hier ist Kost für Feinschmecker geboten, weshalb wir den Artikel auch in voller Länge aufgenommen haben. Solche Artikel könnten wir ein par mehr gebrauchen. Sie müssen ja nicht unbedingt ganz so lang und brauchen auch nicht auf dem gleichen Wissensstand zu sein, nur geschrieben werden müssen sie halt. Ein kleiner Anstoß von uns, "Macht mal Pause beim Programmieren und probiert ob Eure Textverarbeitungsprogramme und der Drucker noch funktionieren."
Auf jeden Fall aber hoffen wir, daß Ihr Euch jetzt ein paar nützliche Anregungen holt, beim lesen von:

G E D A N K E N z u m I N P U T - B E F E H L

über die leidigen Input-Probleme im Basic ist schon viel geschrieben worden - was nicht bedeutet, daß es nach dem 1001. Vorschlag hierzu nicht doch noch etwas zu verbessern oder zu erweitern gäbe. Falls die Problematik nicht allen bewußt ist - hier vorsichtshalber nochmal, worum es überhaupt geht :

Zahleneingaben per "INPUT X" zu holen, sollte man ohnehin vermeiden, besonders wenn auch andere Leute das Programm benutzen sollen. Die Input-Anweisung mit einer numerischen Variablen als Argument gibt ahnungslosen oder hinterlistigen Zeitgenossen die Möglichkeit, mühselig erstellte Bildschirm-Masken zu zerstören weil dem Basic-Interpreter nichts Besseres einfällt, als sich

mit "?REDO" oder ähnlichem Unsinn über die Eingabe von Buchstaben oder Sonderzeichen zu beschweren. Sicherer ist hier schon " INPUT X$:X=VAL(X$) ". Allerdings besteht auch hier noch die einfache Möglichkeit, einen Overflow-Error zu erzeugen. Wenn X keine Integervariable ist, dann eben durch Eingabe eines zu hohen Exponenten.

Bei Eingabe von Text bzw. Strings resultiert ein eingegebenes Komma in der dümmlichen Feststellung: "?Extra ignored" oder "?Zuviele Daten". Dies kann zwar durch Anführungszeichen vermieden werden, aber das wiederum ist halt lästig und leicht zu vergessen. Im übrigen ist häufig genug das Fragezeichen dieser Input-Anweisung unerwünscht - die betreffende Eingabe muß nicht unbedingt die Antwort zu einer Frage sein. Der einzige Vorteil der Input-Anweisung besteht in der Möglichkeit, mehrere Zahlen und/oder Strings in einer Eingabe zu tippen. Aber auch das ist nicht unbedingt das Gelbe vom Ei solange nicht sichergestellt werden kann, daß der Programm-Benutzer auch wirklich die richtige Reihenfolge und Anzahl trifft. Fazit: Die Input-Anweisung sollte man, zumindest für Programme, die man nicht ausschließlich selbst benutzt - getrost vergessen.

Lineinput hat gewisse Vorteile, zumindest bei Texteingaben - kein störendes Fragezeichen, Kommas in der Eingabe erlaubt usw. Lineinput hat aber auch schwerwiegende Nachteile : es erlaubt die Eingabe von Funktionszeichen, also die Möglichkeit, mit den Cursortasten kreuz und quer über den Bildschirm zu schreiben. Die maximal erlaubte Länge der Eingabe kann nicht vorgegeben werden - oft genug ist es nötig, eine ganz bestimmte Anzahl von Eingabezeichen von der Tastatur anzufordern. Und die Eingabe steht erst nach Betätigen der NEW LINE-Taste dem weiteren Programm zu Verfügung - manchmal möchte man Single-Key-Kommandos erkennen und sofort verarbeiten.

Hierzu werden oft Inkey$-Routinen verwendet, etwa in der Form: "I$=INKEY$:IF I$=..." - schön und gut, damit kann man zwar die Single-Key-Kommandos erkennen und jedes einzelne Zeichen auf Zulässigkeit überprüfen, aber spätestens wenn bei längeren Eingaben die einzelnen Inkey$s zusammengefaßt werden - etwa mit "J$ = J$ + I$" - gibt's ein neues Problem: Wegen der häufigen Änderung von J$ kann es dem Basic-Interpreter mitten in der Eingabe einfallen, wegen Platzmangel seine Strings zu sortieren. Der Effekt ist bekannt: sekunden- oder gar minutenlang tut sich gar nichts mehr und wer dann weitertippt, ohne auf den Bildschirm zu achten, darf sich anschließend wundern, wo seine Eingabe geblieben ist.

Aus diesen Erkenntnissen ergeben sich also etliche konkrete Forderungen an eine brauchbare Eingabe-Routine :

a) idiotensicher muß sie sein; alle unerwünschten Eingabezeichen sollten schlicht ignoriert werden. Ein gutes Programm darf auf keinen Fall durch unacht- oder gewaltsam unsinnige Eingaben ins Schleudern zu bringen sein
b) die geforderte oder maximal zulässige Eingabelänge muß vorgegeben werden können - den Chaoten keine Chance zum Kaputtschreiben der Bildschirm-Maske
c) das jeweils erste Zeichen einer Eingabe sollte als Single-Key-Kommando separat behandelt werden können, z.B. um per Cursortasten in eine andere Zeile des Bildschirms zu kommen
d) bei Zahleneingaben muß Overflow-Error verhindert werden
e) für Eingabe-Korrekturen müssen Backspace und Shift-Backspace entsprechend verarbeitet werden

f) die Eingabe sollte gezielt an jede beliebige Stelle des Bildschirms gesetzt werden können, möglichst ohne die Position vorher selbst ausknobeln zu müssen
g) der Garbage-Collector darf nicht während der Eingabe zum Zuge kommen, also keine Benutzung sortieranfälliger String-manipulationen
h) Eingaben müssen selbstverständlich auch vor Erreichen der maximalen Eingabelänge mit NEW-LINE beendet werden können
i) ein blinkender Cursor wäre auch nicht zu verachten
j) alle programminternen Manipulationen müssen schnell genug erledigt werden, um bei 10-Finger-Weltrekord-Tippern keine Eingabezeichen zu verschlucken

Auch wenn die Erfüllung dieser Anforderungen einen halben Texteditor zu benötigen scheint - die Lösung ist mit ein paar Tricks kurz genug, um irgendwo noch ins Programm zu passen. Es bedarf lediglich noch einiger Vorbereitung. Irgendwo am Anfang des Programms, vor dem ersten Aufruf der eigentlichen Eingabe-Routine, ist folgendes unterzubringen :

```
10 CLEARsowenigwiemöglich:DEFINTA-Z:B=B:C=C:R=R:M=M:BE=20
11 ZL=PEEK(14321):J$=STRING$(ZL," "):I$=I$:UG=32:OG=127
12 Z$=CHR$(30):N$=CHR$(14):O$=CHR$(15):L$=CHR$(24):BS$=L$+"."+L$
   :T$=CHR$(7):E$=CHR$(13):S$=CHR$(32):B$=CHR$(8)
13 O=0:E=1:Z=10:H=100:T=1000:E1=2:E2=4:E3=8:E4=16:E5=32:E6=64:
   E7=128:E8=256:JL=JL:JA=JA:MI=45
14 FORN=ETOZ:READI:K$=K$+CHR$(I):NEXT:DATA13,8,9,10,11,24,25,26
   ,27,31
15 DEFFNAD(X)=PEEK(X)+PEEK(X+E)*E8+65536*(PEEK(X+E)=>E7)
```

Die Zeilen 10 - 15 können selbstverständlich so weit wie möglich zusammengefaßt werden und sind hier nur für die noch folgenden Erklärungen getrennt aufgeführt. Die eigentliche Eingabe-Routine ist als Unterprogramm ausgeführt. Es wird beispielsweise aufgerufen mit `R=12:C=32:L=20:UG=48:OG=93:GOSUB900`
Hierbei ist R die Zeile in der die Eingabe beginnen soll, C die Spalte, im Beispiel also Eingabebeginn in der Mitte der 13. Zeile L ist die maximal zulässige Eingabelänge, UG der untere Grenzwert der zulässigen ASCII-Zeichen, OG der obere Grenzwert, im Beispiel also ASCII "0" bis ASCII "ü". Sollen nur Ziffern erlaubt sein, ist OG auf 57 (ASCII "9") zu setzen, UG=65 und OG=127 erlaubt nur Groß- und Kleinbuchstaben incl. Umlaute, ß und die zwischen den Buchstaben in der ASCII-Tabelle liegenden Accents, UG=65:OG=90 erlaubt nur Großbuchstaben ohne Umlaute usw.

Das Unterprogramm übergibt die Eingabe als J$ und die tatsächlich eingegebene Länge als L, oder die Nummer des eingegebenen Single-Key-Kommandos als K, was mit ON K GOTO 1000,2000... leicht weiterverarbeitet werden kann. Wenn eine konkrete Eingabe vorliegt, bleibt K=0 - das ON K GOTO Statement fällt durch und J$ kann unmittelbar dahinter behandelt werden.

Die Parameter R,C,L,UG und OG werden von der Eingabe-Routine nicht verändert. Sofern das übrige Programm hier keine Änderungen vornimmt, brauchen diese Parameter nicht, oder zumindest nicht alle vor jedem Aufruf der Eingaberoutine neu gesetzt werden. So ist

```
1200 C=20:L=4:UG=48:OG=57:FORR=1TO8:GOSUB900:IFK,Z(R)=0:NEXT
     ELSEZ(R)=VAL(J$):NEXT
```

ohne weiteres möglich, wenn z.B. mehrere Zahleneingaben unmittelbar nach- und untereinander benötigt werden. Man beachte, daß hier VAL(J$) direkt in das Integer-Array Z(R) übernommen werden kann, da bei 4 Zeichen Eingabelänge, also maximal 9999 kein

Overflow-Error auftreten kann. Bei größeren Eingabelängen sollte VAL(J$) tunlichst zuerst in eine Single- oder Double-Precision Variable übernommen und erst nach Abchecken der Integergrenzen in einen Integer übertragen werden. Exponenteneingabe kann durch Beschränken der Eingabe auf Ziffern nur ganz unterdrückt werden, aber wann bitteschön, braucht man unbedingt die Exponenten ? Notfalls kann man immer noch die entsprechende Anzahl Ziffern tippen. Vielleicht eine etwas umständliche Art besonders bei extrem hohen Zahlen, aber 38-stellige sind wohl selten genug.

```
900 N=H:K=0:B=0:JL=VARPTR(J$):POKEJL,0:JA=FNAD(JL+E)-E:PRINT@
    R*ZL+C,0$STRING$(L,".")Z$STRING$(L,L$)N$T$;:FORN=ETOL
910 FORM=0TO0:I$=INKEY$:M=I$="":B=(B<BE)*NOT(B):PRINTCHR$(14-
    (B>BE/E1)):NEXT:I=ASC(I$):ON-(I<E5)GOTO920:ON(I<UGORI>OG)
    *(I<>E5ANDI<>MI)GOTO910:PRINTI$;:POKEJA+N,I:NEXT:PRINT0$T$;
    :POKEJL,L:RETURN
920 ON-(N=E)GOTO930:IFI$=B$,N=N-E1:PRINTBS$;:NEXTELSEIFI$=L$,
    900ELSEIFI$=E$,PRINT0$STRING$(-NOT(L-N),S$);:L=N-E:N=H:
    NEXT:POKEJL,L:RETURNELSEN=NOT(-N):NEXT
930 K=INSTR(K$,I$):IFK,N=H:L=0:PRINT0$;:RETURNELSEN=NOT(-N):
    NEXT
```

Das war's auch schon. Wer sich mit dem Funktionieren als gegebener Tatsache zufrieden gibt und kein Interresse an wieso und warum hat, der möge eintippen und weiterblättern. Ansonsten gibt es vielleicht noch einige Klarheiten zu beseitigen :

zu Zeile 10
Die Clear-Anweisung ist mit Vorsicht zu genießen. So wenig wie möglich bedeutet, das häufig String-Space sortiert werden muß - aber gerade deswegen gibt es eben nicht viel zu sortieren und die Zwangspausen sind meist so kurz, daß sie garnicht mehr auffallen. Umgekehrt wird bei reichlich String-Space seltener, vielleicht auch garnicht sortiert - aber wenn, dann möglicherweise minutenlang. Im übrigen ist die Dauer des Sortier-Vorgangs außer von der Häufigkeit von String-Manipulationen auch stark von der durchschnittlichen String-Länge abhängig - wenige lange Strings sind schneller sortiert als viele kurze. DEFINTA-Z sollte ohnehin verwendet werden. Integerzahlen werden nun mal am Häufigsten gebraucht bzw. nicht alle Zahlen müssen unbedingt Single- oder Double-Precision sein.

Ansonsten sollten die am Häufigsten verwendeten Variablen tunlichst als Erste definiert werden, auch wenn denen am Anfang noch gar kein konkreter Wert zugewiesen werden kann. B=B erfüllt denselben Zweck wie B=0. Beim Aufruf einer Variablen wird die Variablen-Tabelle sequentiell durchsucht - logischerweise werden die zuerst definierten Variablen somit auch am Schnellsten gefunden.
BE ist der Blinkzähler-Endwert, verantwortlich für die Blinkfrequenz des Cursors. 20 gilt für den 4 MHz Takt des Genie III oder Genie I/II mit 3.5 MHz Speedup-Mod. Für Genie I/II mit 1.78 MHz empfiehlt sich etwa 10-12.

zu Zeile 11
ZL ist die momentan aktuelle Breite der Bildschirmzeilen, nötig für die Positions-Berechnung aus Zeile und Spalte. Als Peek aus der Parameter-Tabelle allerdings nur bei den verschiedenen Bildschirm-Formaten des Genie III nötig. Bei Genie I und II kann hier schlicht ZL=64 stehen. Ich weiß - es gibt auch noch das 32-Zeichen Format. Aber wer sich ein Gerät kauft, das 64 Zeichen bietet um dann mit 32 Zeichen zu arbeiten, der sollte

sich besser gleich einen 250-Mark-32-Zeichen-Micky-Maus-Computer zulegen, eine stärkere Brille verpassen lassen oder meinetwegen ZL=64-4*(PEEK(16445)AND8) einsetzen - siehe Tip von Hans-Otto Langguth aus ERROR Nr.2 Seite 9. (Ich fordere Verfechter des 32-Zeichen-Formats zu geharnischten Leserbriefen auf...)

J$ ist der eigentliche Eingabe-String, der hier als Dummy mit der Bildschirm-Zeilenlänge als maximal zulässiger Eingabelänge definiert wird.UG und OG erlauben erst einmal den gesamten Bereich ausdruckbarer ASCII-Zeichen, brauchen also nur dann gesetzt werden, wenn dieser Bereich eingeschränkt werden muß.

zu Zeile 12
Hier sind Bildschirm-Funktionen vorgegeben: Löschen bis zum Ende der Zeile, Cursor an, Cursor aus, Cursor links, Back-space spezial, Tastatur-Piep (nur für Genie III), New-Line Space und Backspace. Warum ? Nun - PRINTCHR$(14) belegt 6 Byte im Programmtext, PRINTN$ nur 3 und ist außerdem schneller.

zu Zeile 13
Die 10er und 2er Potenzen werden hier nur teilweise benutzt, ansonsten aber doch oft genug um frei auf Verdacht mitge-schleppt zu werden. Schließlich wird ein Variablen-Aufruf immer schneller ausgeführt als die Umwandlung einer ASCII-Ziffern-Folge in die entsprechende Binärzahl oder gar Neuberechnung von Potenzen. Aus diesem Grund sollte man ohnehin bei mehr als einmaliger Benutzung besser Variablen statt Klartext-Zahlen verwenden, auch wenn es sich eigentlich um Konstanten handelt - niemand zwingt einen, Variable zu variieren...

zu Zeile 14
Hier werden die vorgesehenen Single-Key Kommandos (New - Line, Pfeil links, rechts, runter, rauf, dito mit Shift, Clear) zu K$ zusammengefaßt. Die 10-malige Neuzuweisung von K$ stopft zwar allerhand Müll in den String-Space und ist daher sehr sortier-gefährlich, aber als einmaliger Vorgang bei der Programm-Initialisierung sei's erlaubt. Bei der Gelegenheit: Level II Benutzer, denen die INSTR-Funktion nicht zu Verfügung steht, sollten hier besser gleich "FORN=ETOZ:READK(N):NEXT:DATA..." einsetzen. DIM ist bekanntlich bei 10 noch nicht nötig.

zu Zeile 15
Mit der Funktion AD(X) können auch anderweitig bequem Adressen unter dem Pointer X aus den Parameter- und Vectortabellen geholt werden. Mit "+65536*(PEEK(X+E)=>E7)" wird dafür gesorgt, daß Adressen über 7FFFh in negative Integer umgewandelt werden. Level II Benutzern bleibt allerdings nichts Anderes übrig, als diese Berechnung statt FNAD in Zeile 900 unterzubringen.

zu Zeile 900
N ist der Eingabelängen-Zähler und wird hier über den maximal zulässigen Wert gesetzt. Damit wird bei eventuellem Neustart der Eingabe-Routine die Eingabeschleife gewaltsam abgebrochen weil - mmh, naja - die Eingabe-Routine zumindest ein bißchen rekursiv ist. Jedenfalls wird bei Eingabe von Shift-Backspace aus der laufenden Schleife herausgesprungen und die gesamte Routine neu gestartet. Was allerdings Compilieren mit ACCEL o.ä. unmöglich macht, aber da gelten ohnehin noch ganz andere Kriterien.

K ist die Nummer des letzten Single-Key-Kommandos und wird vor-sichtshalber zurückgesetzt falls von der letzten Eingabe noch

eine Nummer übriggeblieben ist. B ist der Delay-Zähler für's Cursor-Blinken. JL ist der Pointer zum Längenbyte von J$, das vor Beginn der Eingabe auf Leerstring gepoked wird. Dies erspart das Löschen von J$. JA ist der Pointer zum eigentlichen String-Inhalt. Wie gesagt, Level II Benutzer müssen die Berechnung von FNAD direkt hier einsetzen. Wichtig ist, das JL nach wie vor auf das Längenbyte und nicht auf das untere Adressbyte von J$ zeigt, also PEEK(X) durch PEEK(JL+E) und PEEK(X+E) durch PEEK(JL+E1) ersetzen. Ferner zeigt JA vor Beginn der Eingabe um 1 Byte vor das erste Zeichen von J$, also -E nicht vergessen!

In der PRINT@-Anweisung wird die Bildschirm-Position aus Zeile und Spalte berechnet. BASIC-80 Benutzer können sich die Berechnung mit ZL schenken und direkt PRINT@(R,C), vorgeben. (Wobei mir auch dabei schleierhaft bleibt, warum hinter einem PRINT@-Statement ein Komma stehen muß, wo ein Semikolon wie im Mc-Farland-Basic doch wohl logischer wäre - kenn' sich einer in den Hirnwindungen der Microsoft-Spezis aus...)

O$ schaltet den Cursor erst einmal ab, damit der nicht störend über den Bildschirm flimmert, wenn mit STRING$(L,".") die angeforderte Eingabelänge in Form einer Punktlinie ausgegeben und mit STRING$(L,L$) an den Beginn der Punktlinie zurückpositioniert wird. PRINT STRING$ benutzt wohlgemerkt nur den vom System bereitgestellten Literal String Pool für Zwischenergebnisse bei String-Manipulationen. Es erfolgt kein Eintrag des STRING$ in den reservierten String-Speicherbereich, somit auch keine Gefahr, daß der sortiert werden muß.

Mit Z$ wird der Rest der Bildschirmzeile hinter der Punktlinie gelöscht, falls von einer vorhergehenden, längeren Eingabe an der gleichen Stelle hier noch etwas übriggebleiben ist. Soll hinter der Eingabe noch irgend etwas auf dem Bildschirm in der gleichen Zeile stehen bleiben, ist Z$ wegzulassen. Mit N$ wird der Cursor schließlich wieder eingeschaltet und mit T$ durch einen kurzen Piepton Aufmerksamkeit erzeugt. Bei Genie I und II könnte T$ durch einen entsprechenden USR-Call ersetzt werden. Für Leute, die's interressiert, habe ich ein paar Byte Object-Code in der Schublade.
FOR N=1 TO L ist der Beginn der eigentlichen Eingabeschleife.

zu Zeile 910
For M gleich null to null?? So blödsinnig, wie's aussieht, ist die Sache garnicht. Bekanntlich werden FOR-TO-NEXT-Schleifen mindestens einmal durchlaufen, auch wenn der Schleifenzähler-Endwert bereits vor Beginn der Schleife erreicht oder überschritten ist. Hauptaufgabe dieser Schleife ist die laufende Wiederholung der Tastaturabfrage mit I$=INKEY$. Bei der vielleicht ebenfalls merkwürdig erscheinenden Anweisung M=I$="" sind die beiden Gleich-Zeichen keineswegs gleich. Das zweite Gleich-Zeichen wird als Vergleichs-Operator interpretiert, das erste als Zuweisungs-Operator, mit dem dem Schleifenzähler M das Ergebnis des Vergleichs I$="" zugewiesen wird. Klammern sind hier nicht nötig, weil der Vergleich Priorität gegenüber der Zuweisung hat. Solange bei INKEY$ keine Taste gedrückt wurde, ist das Ergebnis des Vergleichs I$="" logisch wahr, also -1. Damit wird M vor dem NEXT unter den Schleifenzähler-Endwert gesetzt und die Schleife folglich wiederholt. Wenn eine Taste gedrückt wurde, ist der Vergleich falsch, M wird null - der Schleifen-Endwert ist erreicht. Praktisch wird hiermit eine - sonst im Genie-Basic nicht vorhandene -

DO UNTIL -Schleife simuliert, nämlich DO I$=INKEY$ UNTIL I$>""
Auf ähnliche Weise kann auch WHILE WEND simuliert werden, wäre vielleicht eine neue Knobelaufgabe, denkt mal drüber nach!
Noch was zum Thema INKEY$: Selbstverständlich wird dieser INKEY$ nach gedrückter Taste vom Interpreter neu in den String-Space eingetragen. Ohne Tastendruck wird aber lediglich das Längenbyte im Variablenpointer von I$ gelöscht, die Adresse des Inhalts von I$ bleibt unverändert erhalten und wird auch weiter benutzt solange zwischen den Tastaturabfragen keine anderen String-Manipulationen stattfinden. Das bedeutet, das neu von der Tastatur gelesene Zeichen wird vom Interpreter ausnahmsweise an der alten Stelle in den String-Space eingetragen, es wird also nicht mehr Platz belegt als vorher und der gierig lauernde Garbage-Collector hat keine Chance sein nervtötendes Werk zu tun.

BASIC-80 Benutzer sollten es sich verkneifen, hier einfach die Möglichkeit I$=INPUT$(1) zu nutzen. Die Zeit zwischen den Tastendrücken kann schließlich noch genutzt werden, um für den blinkenden Cursor zu sorgen. Das geschieht so ganz nebenbei mit den übrigen beiden Anweisungen innerhalb der "DO UNTIL"-Schleife. NOT(B) ist nichts anderes als -(B+1). Der Vergleich B<BE liefert -1 solange der Blinkzähler B kleiner als der Blinkzähler-Endwert BE ist. -1*-(B+1)=B+1 Warum also nicht gleich B=B+1 ? Die Pointe kommt beim Erreichen des Blinkzähler-Endwertes. Dann ist der Vergleich B<BE falsch, der ganze Ausdruck wird null und B somit laufend von 0 - BE-1 hochgezählt und wieder auf 0 gesetzt. In verschiedenen Basic-Versionen ist der Modulo-Operator verfügbar. In diesem Fall kann die Angelegenheit natürlich mit B=-NOT(B)MODBE ebenfalls durchgeführt werden, wobei zwar nicht mehr von 0 bis BE-1 sondern von 1 bis BE gezählt wird, aber das ändert nur minimal das Tastverhältnis und nicht die Blinkfrequenz des Cursors.

Im folgenden PRINT-Statement liefert der Vergleich B>BE/2 null, solange der Blinkzähler kleiner als die Hälfte des Blinkzähler-Endwertes ist. PRINT CHR$(14-0) => Cursor an. Ist der Vergleich wahr, folgt CHR$(14-(-1)) = CHR$(14+1) = CHR$(15) => Cursor aus. Der Cursor wird also mit nur zwei Anweisungen laufend ein- und ausgeschaltet und blinkt friedlich vor sich hin.
Nebenbei bemerkt: Der vorgeschlagene BE-Wert von 10-12 für einen etwa Sekunden-Blinkrhythmus beim Genie I/II zeigt schon, daß die Tastatur-Abfrage nur etwa 10-12 mal pro Sekunde durchlaufen wird. Zwar wären mehr als 720 Anschläge pro Minute absoluter Weltrekord im Schnellfeuer-Tippen, (hab' mal was läuten hören, der liegt tatsächlich so um 500 - melde sich, wer's besser weiß - oder kann...) aber es gibt schließlich noch mehr zu tun, als nur die Tastatur-Abfrage. Deshalb wird auch im Folgenden mit ON GOTO statt IF THEN eine beschleunigte Gangart eingelegt. ON-(I<E5)GOTO entspricht IF irgeneine Kontroll-Taste gedrückt THEN Zeile. Der Vergleich I<E5 liefert -1 wenn das gelesene ASCII-Zeichen kleiner als 32, also eine Kontroll-Taste war. ON-(-1) also ON 1 GOTO funktioniert, alles Andere liefert ON 0 GOTO, was durchläuft und hinter dem GOTO mit der Interpretation weitermacht. Bei IF THEN wäre für den gleichen Zweck ein ELSE notwendig, was verhältnismäßig viel Zeit kostet, da dieses ELSE erst einmal vom Interpreter in der Zeile gesucht und gefunden werden muß.

Der folgende Ausdruck sorgt dafür, daß alle ASCII-Zeichen außerhalb des per UG/OG zugelassenen Bereichs ignoriert werden. Schlauberger sollten hier nicht voreilig auf die Idee kommen, das OR durch ein + zu ersetzen. Nicht nur, weil Addition der beiden Vergleichsergebnisse auch -2 ergeben könnte, wenn beide Vergleiche war sind - solange UG und OG korrekt vorgegeben sind, kann I nicht gleichzeitig kleiner als UG und größer als OG sein - vielmehr erfordert die Addition wegen Priorität gegenüber den Vergleichsoperatoren, beide Vergleiche in eigene Klammern zu setzen. Wobei die expression-evaluation-routine des Interpreters zusätzlich Zeit zum separaten Auflösen der Klammern benötigt. Der Vergleich (I<>E5ANDI<>MI) sorgt dafür, daß unabhängig von UG und OG Leerzeichen und Minuszeichen immer zugelassen werden indem im Fall eines Leer- oder Minuszeichens der ganze Ausdruck mit Null multipliziert wird, oder für eine positive Eins für das ON GOTO-Statement wenn I außerhalb der Grenzwerte liegt und weder Leerzeichen noch Minuszeichen ist.

Sobald dieses ON GOTO durchläuft, steht fest, das I$ ein zulässiges, ausdruckbares ASCII-Zeichen ist, das dann mit POKE JA+N,I in den reservierten Dummy-String J$ eingetragen wird. NEXT bezieht sich auf den Eingabe-Zähler N und holt auf die gleiche Weise das nächste Zeichen. Hinter dieses NEXT gerät der Interpreter nur, wenn die volle vorgegebene Länge eingetippt wurde. Dann wird mit O$ der Cursor wieder abgeschaltet, mit T$ zart darauf hingewiesen daß es jetzt reicht, mit POKE JL,L das Längenbyte im Variablenpointer von J$ gesetzt und der Fall ist erledigt.

Noch ein Wort zum Poken in den Dummy-String: Prinzipiell besteht natürlich auch die Möglichkeit, den eingegebenen String einfach nur auf dem Bildschirm zu printen und dann die Stringadresse auf eben diese Bildschirmadresse umzupoken. Auch keinerlei Stringmanipulation und sogar noch schneller. Leider hat dieses Verfahren den Nachteil, daß die Eingabe schnell zum Teufel ist, wenn sich anschließend noch mehr auf dem Bildschirm tut. Besonders beim Editieren des Programms hat man leicht einen ganz anderen String an der entsprechenden Adresse stehen. Für andere Tricks wie z.B. einen schnellen Screen-Editor in Basic oder Basic-Programm schreibende Basic-Programme sind solche Strings im Bildschirm allerdings sehr nützlich.

Für Genie III Benutzer gibts noch einen anderen Haken: Alles was im 80-Zeichen Modus ab Zeile 13, Spalte 64 auf dem Bildschirm steht, liegt im zweiten Video-RAM. Und das wird nach der Bildschirmausgabe vom Sytem wieder abgeschaltet und ist vom Basic aus nicht mehr zugänglich. Vom Basic aus gesehen würde J$ dann irgendwo mitten in den Parameter- und Vectortabellen ab Adresse 4000h liegen - PRINT J$ den Bildschirm zur Geisterbahn machen und eine Änderung von J$ u.U. das Betriebssystem in die ewigen Jagdgründe schicken...

zu Zeile 920
Hier werden Backspace, Shift-Backspace und New-Line behandelt. Zuerst jedoch mit ON-(N=E)GOTO sprich IF N=1 THEN zur entsprechenden Single-Key-Testroutine verzweigt, falls dies das erste Zeichen der Eingabe war, denn nur das jeweils an erster Stelle getippte Zeichen soll ja als Single-

Key-Kommando verwendet werden. Außerdem gäb's beim ersten Zeichen auch noch nichts zu backspacen und der entsprechende Test wäre überflüssig.
Hier sei ausnahmsweise der zeitverschwenderische Luxus einer IF-Abfrage genehmigt. Eingabekorrekturen mit Bildschirmbeobachtung werden sowieso langsamer getippt und ein ON GOTO -Verteiler auf mehrere Zeilen würde wiederum in der Quell-Programmlänge in die Vollen gehen. Bei Backspace muß hier der Längenzähler N um 2 zurückgesetzt werden weil das Backspace-Zeichen selbst ja auch schon mitgezählt ist. BS$ ist eine Backspace-Spezialausgabe, nämlich Cursor links, Punkt, Cursor links. Die Punktlinie für die Längenvorgabe muß schließlich auch wieder in die Reihe kommen. Falls wer sich an dem Komma stört und sich das immer noch nicht herumgesprochen hat: beim Genie I/II/III und TRS-80 I Basic (nicht bei allen Microsoft-Basic-Dialekten, siehe BASIC-80) kann das THEN jederzeit durch ein schlichtes Komma ersetzt werden auch in Verbindung mit ELSE. Im Prinzip also eine Abkürzung wie das ? für PRINT (das ' für REM steht wieder auf einem anderen Blatt, weil's nur scheinbar eine Abkürzung ist...)

Einen Haken hat dieses THEN-Komma aber doch: einige der gebräuchlichsten Renumber-Utilities erkennen bei Kombinationen wie "IF Bedingung Komma Zeilennummer" eben diese Zeilennummer nicht als Sprungziel sondern als Konstante und renumbern sie nicht mit. Und genau das ist hier bei IF I$=L$,900 zu berücksichtigen.

I$=E$ bedeutet New-Line getippt, also muß die Angelegenheit korrekt beendet werden indem mit 0$ der Cursor abgeschaltet und der überflüssige Rest der Punktlinie gelöscht wird. -NOT(L-N) heißt vorgegebene Maximallänge (=Länge der Punktlinie) minus tatsächlich eingegebene Länge plus 1 (für die bereits als Eingabe mitgezählte New-Line-Taste) BASIC-80 könnte hier ohne weiteres SPACE$(-NOT(L-N)) verwenden oder besser SPC(...) was eben keine Stringfunktion ist.

Anschließend wird die Länge L auf Eingabezähler -1 gesetzt und die Eingabeschleife mit N=100:NEXT vorzeitig abgebrochen. Newdos 80 oder Gdos Benutzer könnten hier eventuell mit CMD"F=POPN" noch ein paar Sekundenbruchteile rausquetschen, aber erstens lohnt es sich nicht weil die Eingabe eh' zu Ende ist und zweitens besteht dabei akute Schleudergefahr - das heißt entweder mach' ich grundsätzlich was falsch oder der Erfinder dieser CMD"F=POPx"-Statements hat irgendwo noch einen Wurm dringelassen. Jedenfalls findet sich das Programm auf dem Rückweg gelegentlich im Urwald wieder. Zwar ist für solche Fälle die Paniktaste serienmäßig im Gerät eingebaut, aber ich glaube kaum daß besonderer Bedarf an speziellen "Reset-Tasten-Testprogrammen" herrscht - ich halte mich in diesem Fall lieber an bewährte Methoden. (Möglicherweise hat jemand hierzu einen Tip auf Lager ?)
Falls der Interpreter hinter dem letzten ELSE landet, ist ein ungültiges Kontrollzeichen getippt worden. Das wird ignoriert, indem der Schleifenzähler wieder um 1 zurückgesetzt und mit NEXT dieselbe Eingabe wiederholt wird.

zu Zeile 930
Hier findet sich nur noch der Test auf gültige Single-Key-Kommandos. Falls dieses "IF K ," ungewohnt ist - das Argument einer IF-Abfrage muß nicht notwendigerweise ein Vergleich sein, IF reagiert auf alles, was ungleich null ist.

"IFK," ist also nichts Anderes als "IF K<>0 THEN". Level II Benutzer brauchen angesichts der INSTR-Funktion nicht die Flinte ins Korn zu werfen. Zeile 14 mit K%(1 bis 10) und hier 930 FORS=ETOZ:K=-S*(I=K(S)):S=S+K*Z:NEXT:IFK,... erfüllt den gleichen Zweck, nur nicht ganz so schnell. K=-S*(I$=MID$(K$,S,E)) wäre prinzipiell auch möglich um die INSTR-Funktion zu simulieren - man kann sich in diesem Fall ja die Zeit bis zu Ermittlung von K gemütlich mit ERROR-Lesen vertreiben...

Der langen Rede kurzer Sinn: Es lohnt sich, die kleinen grauen Gehirnzellen ins Rotieren zu bringen und bis ins Detail auszutüfteln, was Programm werden soll. Zwar geht nach wie vor Probieren über studieren, aber auch bei solch kleinen Sachen dürfte etwas Denken und die Beherrschung von ein paar Tricks bessere Ergebnisse versprechen als das übliche Drauflos-Tippen mit anschließendem Editier-Flickwerk. Gerade bei Daten-Eingabe-Routinen wird viel Mist verzapft - siehe Bemerkungen zum INPUT-Statement. Und gerade sicheres Funktionieren sollte oberstes Gebot für ein Programm sein. Wobei ich wahrhaftig keinen Anspruch erhebe, mit diesen paar Programmzeilen die einzig wahre Eingabe-Routine gebracht zu haben. Im Gegenteil - ein Wort in den Gehörgang aller Tüftler und Besserwisser : Ran an den Speck! Es gibt viel zu verbessern, Resonanz ist erwünscht!

Wie wäre es zum Beispiel mit Autorepeat? Oder Zahleneingabe mit Exponenten? Kommandowort-Erkennung? Verbesserungsmöglichkeiten gibts genug und gerade eine solide Eingabe-Routine wird immer wieder gebraucht.

So, jetzt reicht's. Nachdem aus ein paar Programmzeilen ein abendfüllender Roman geworden ist, bin ich sicher, die Durchblicker unter euch ausreichend mit überflüssigen Erklärungen gelangweilt zu haben (soweit die überhaupt bis hierhin mitgelesen haben). Nichtsdestotrotz sollte dieser Artikel auch ein Anreiz für alle sein, sich hinzusetzen und aus dem Nähkästchen zu plaudern, soll heißen : etwas aus der Trickkiste zu schreiben - muß ja nicht gleich ein Gedicht sein... Der ERROR könnte durchaus mehr Inhalt gebrauchen und es gibt bestimmt genug Leute, die sich dafür interressieren. Also nur keine Hemmungen!

Alle, die durch meine Ausführungen auf den Geschmack gekommen sind und nach mehr lechzen, möchte ich auf andere Quellen vieler Weisheiten hinweisen. Lewis Rosenfelders "BASIC FASTER AND BETTER" zum Beispiel. TRS-80 Information Series Vol.IV, ISBN 0 936200 03 0 Auch wenn die hierin aufgeführten Anwendungs-Beispiele und vor Allem die Variablen-Benennung teilweise etwas hahnebüchen sind und der Preis (au weia!) dem Inhalt an üppigkeit in nichts nachsteht - dieses Buch ist sozusagen eine Art Meisterkurs in BASIC und durchaus sein Geld wert.
Wem allerdings kein 3-Sterne-Talent in Englisch gegeben ist und trotzdem der Sinn nach mehr Durchblick steht, dem seien die Basic-Kurse des Clubs wärmstens ans Herz gelegt. Hier werden solche und ähnliche programmtechnichen Klimmzüge ausführlich in Deutsch breitgetreten. Also auch hier - nur keine Hemmungen, man lernt nie aus!

Jürgen Kuschel

** DAS PROGRAMM DES QUARTALS **

Heute wie versprochen die etwas schnellere Sortierroutine QICK-SORT, die gegenüber BUBBEL-SORT erheblich Vorteile bringt, aber dafür etwas länger ist. Beim nächsten Mal stellen wir Euch dann noch 2 andere Sortierroutinen vor, HAURUCKSORT und EINFÜGESORT womit wir dieses Thema dann abschließen.

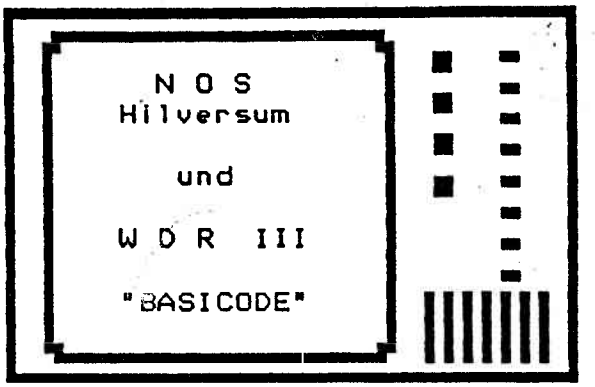
```
100 REM *** QUICKSORT, EINE ETWAS SCHNELLERE SORTIERROUTINE ***
110 CLS: CLEAR 100
120 INPUT "WIEVIELE EINGABEN"; N
130 DIM NA$(N): DIM S(N,2)
140 FOR I=1 TO N
150 PRINT "EINGABE NR.";I;
160 INPUT NA$(I)
170 NEXT I
180 L=1 : R=N
190 GOSUB 250
200 PRINT:PRINT "SORTIERTE REIHENFOLGE:"
210 FOR I=1 TO N
220 PRINT NA$(I),
230 NEXT I
240 END
250 SZ=1: S(1,1)=L: S(1,2)=R
260 IF SZ <1 THEN RETURN
270 LA=S(SZ,1): RA=S(SZ,2)
280 GOSUB 330
290 SZ=SZ-1
300 IF LA <TZ-1 THEN SZ=SZ+1: S(SZ,1)=LA: S(SZ,2)=TZ-1
310 IF (TZ+1<RA) THEN SZ=SZ+1: S(SZ,1)=TZ+1: S(SZ,2)=RA
320 GOTO 260
330 LZ=LA: RZ=RA: TE$=NA$(LZ)
340 IF LZ=RZ THEN 400
350 IF (NA$(RZ)>=TE$) AND (LZ<RZ) THEN RZ=RZ-1: GOTO 350
360 NA$(LZ)=NA$(RZ)
370 IF (NA$(LZ)<=TE$) AND (LZ<RZ) THEN LZ=LZ+1: GOTO 370
380 NA$(RZ)=NA$(LZ)
390 GOTO 340
400 TZ=LZ: NA$(TZ)=TE$
410 RETURN
```

****
IN LETZTER MINUTE
****

Noch ein paar Anmerkungen in eigener Sache:

Natürlich gilt unser Angebot die GENIE-DATA über uns für 20.-- statt für 30.-- DM zu abbonieren immer noch. Also fleißig bestellen, denn erst bei 20 ABBOS setzen wir nicht mehr zu.

Die EACA-GENIE-NACHRICHTEN, welche dem ERROR ja eigentlich beiliegen sollen, sind leider noch nicht ausgeliefert. Entweder hat es sich mit der Nr. 1 schon gehabt, oder Trommeschläger ist mit der Produktion genau so am schleudern, wie wir mit unserem ERROR. Wir hoffen aber das sie beim nächsten mal dabei sind.



Wie wir im ersten Teil über BASICODE bereits berichtet haben, ergeben sich durch dessen Gebrauch die Möglichkeit alle Basic-Programme, unter verschiedenen, normalerweise nicht kompatiblen Geräten austauschbar zu machen. Womit sich vor allen für CLUBS und Hobby-Anwender die Möglichkeit zum unbegrenzten Tausch von BASIC-Programmen ergibt. Dafür müssen dann aber einige Bedingungen erfüllt sein, auf welche schon beim Erstellen von Programmen zu achten ist. Diese Bedingungen sind in einem STANDARD festgelegt, der von " HOBBYSCOPE-NOS-HILVERSUM " neu festgelegt worden ist, und bei der Ausstrahlung von Programmen in BASICODE verwendet wird.

1.) BASICODE-Programme dürfen keine Maschinensprach-Subroutine enthalten.
2.) Da bei verschiedenen Fabrikaten diverse Basic-Befehle eine unterschiedliche Benennung haben, darf nur ein vorgegebener Satz von STANDARD-BEFEHLEN und OPERATOREN verwendet werden. Für Befehle die nicht diesem STANDARD entsprechen, werden vorgegebene Subroutinen verwendet, welche in BASICODE durch "GOSUB" angesprungen werden. Die Zeilennummern für diese Subroutinen sind vorgegeben und liegen im Bereich von 0 bis 999. Sie werden dem jeweiligen BASICODE-Programm vorgeladen. Sie müssen beim Anwender in diverser Form mit dem BASICODE-übersetzungsprogramm vorliegen und sind von Gerät zu Gerät verschieden.
3.) VARIABLEN sind in Ihrer Form und Benennung teilweise vorgegeben und teilweise ausgeschlossen.
4.) BASICODE funktioniert nur im KASSETTEN-Betrieb. Alle DOS-funktionen sind also ausgeschlossen und dürfen nicht verwendet werden.
5.) Das Bildschirmformat enthält 24 Zeilen a 40 Zeichen Länge. (TRS-80 und VIDEO-GENIE Anwender sind hier insofern benachteiligt, daß eingeladene Fremdprogramme in diesem Standard erst auf 16 Zeilen abgeändert werden müssen. Hier wäre bestimmt eine NORM von 16 x 40 Zeichen angebrachter gewesen.)

Durch diese Konventionen, auf die bei Erstellung eines Programms für BASICODE zu achten ist, ist es nicht mehr erforderlich, ein in BASICODE ausgestrahltes Programm, vor Gebrauch dem eigenen Rechner anzupassen, bevor es ohne Fehler arbeitet. Wie sieht das nun in der Praxis aus? Hier ein Beispiel:

> Der Befehl "CLEAR-SCREEN (CLS)" ist bei jedem Computer anders. Darum soll in BASICODE jedes ausgesendete Programm, das zu einem bestimmten Zeitpunkt den Bildschirm löschen soll, an dieser Stelle den Befehl > GOSUB 100 < enthalten. Wenn dann in der Zeile 100 eine SUBROUTINE steht, die den Bildschirm löscht, dann ist das Problem des "CLEAR-SCREEN" von selbst gelöst. Bevor man das ausgestrahlte BASICODE-Programm lädt, muß man natürlich die benötigten Subroutinen laden. In diesem Fall "100 CLS:RETURN".

Es gibt diverse Subroutinen, die jeweils eine bestimmte notwendige Funktion ausführen sollen, welche in den verschiedenen Computern unterschiedliche Kommandos benötigen.

z.B. - Cursor auf eine bestimmte Stelle setzen.
- Feststellen, ob eine Taste gedrückt ist oder nicht.
- Warten bis eine Taste gedrückt ist.
- Darstellen von Zahlen in Formatierung (PRINT-USING).
- Erzeugen von Zufallszahlen mit RND. Usw. usw..

Diese Subroutinen sind sehr sorgfältig neu ausgewählt worden, natürlich sollten auch alle früher ausgestrahlten Programme mit diesen neuen Subroutinen, ohne große Probleme, sofort auf Ihrem Computer laufen. Was dann übrig bleibt, ist leider immer noch eine ganze Menge, wie beispielsweise: Zeit abfragen, blinkende Texte, Musik, Farbe, Highresolution und einiges andere mehr.

Warum hierfür im neuen BASICODE-Standard noch keine Subroutinen vorgesehen sind, liegt auf der Hand. Es gibt einfach noch zuwenig Fabrikate, welche diese Möglichkeiten alle verarbeiten können. Mit fortschreitender Entwicklung werden diese Möglichkeiten aber sicher auch berücksichtigt werden. Genügend Platz ist in dem vorgegebenen Zeilenbereich, der im Moment von Zeile 400 bis 1000 noch frei ist, auf jeden Fall vorhanden.

Die bisher festgelegten Subroutinen erreichen auf jeden Fall einen Stand, mit welchem es möglich ist, entsprechend der derzeitigen Situation auf dem Mikro-Computer-Markt eine erfreulich große Anzahl von BASIC-Programmen für die meisten Geräte kompatibel zu machen. Natürlich wollen sie genau wissen welche Subroutine an welcher Stelle stehen muß. Dafür reicht diesmal leider der Platz und die Zeit nicht, so daß wir diese dann im nächsten ERROR veröffentlichen werden. Heute haben wir erst mal eine Liste mit den Konventionen für das Erstellen von Basicode-Programmen zusammengestellt. (Siehe nächste Seite).

Für alle Computer für die BASICODE-übersetzungsprogramme bestehen, wurden diese dem neuen Standard gemäß ausgestrahlt. Sie liefern genaue und klare Definationen der neuen Subroutinen. Es ist also keinesfalls erforderlich diese und ihren Platz selbst herausfinden zu müssen, was natürlich auch möglich ist. Es ist aber erforderlich, daß beim Erstellen von Programmen, diese an den neuen Standard angepasst werden. Ohne diesen Standard wird kein Programm auf einem anderen Computerfabrikat laufen können.

Gegen übersendung von 25.-- HFL (Gulden) per EuroscheckK oder internationaler Postanweisung, kann man bei NOS-Hilversum ein Buch über BASICODE, mit dem zugehörigen übersetzungsprogramm auf Kasette bestellen. Es ist unter Mitarbeit des niederländische Computerclubs "H C C" und der Redaktion von "Hobbyscope" erstellt worden und enthält alles wissenswerte. Leider ist es, wie nicht anders zu erwarten, in englisch geschrieben.

Die Kassette mit dem übersetzungsprogramm und den Subroutinen gibt es für folgende Geräte:

ACORN ATOM	APPLE II	COSMICOS	EXIDY-SORCERER
DAI	PET/CBM	PHILLIPS P-200	OSI-CHALLENGER
MZ80 B/K	TI 99/4	SINCLAIR ZX81	TRS-80/VIDEO-GENIE
BBC Mod.A+B	VIC 20	COMMODORE 8000	S.W.T.P. COMPUTER

In dem Buch wird ein Listing des übersetzungsprogrammes gegeben, alle FREKS von anderen Fabrikaten sind aufgerufen, dieses für ihre Zwecke umzustricken. Es wird ausdrücklich darauf hingewiesen, daß auf Buch sowie Programmen kein Copyrecht besteht, und BASICODE für "non-profit use" konzipiert ist. Wir meinen, ein recht netter Zug, den wir uns auch bei manch anderem wünschen würden.

Wird fortgesetzt Lars Brossart

* BASICODE-STANDARD-KONVENTIONEN *

1.) Die Konventionen:

A. Man benutzt nur BASIC-Statements, die alle Computer kennen.
B. Die Zeilennummern bis 1000 sind für Subroutinen reserviert. Sie enthalten die Subroutinen im Rahmen des übersetzungs-Programms für den Computer des Anwenders.
C. Der Bildschirm enthält 24 (16) Zeilen von 40 Zeichen Länge.
D. Eine Programmzeile darf einschließlich Zeilennummer und der Leerstellen maxim. 60 Zeichen lang sein.
E. Zeilennummern über 32767 dürfen nicht verwendet werden.

2.) Der BASICODE-Programmaufbau:

```
   0 -   999 : die BASICODE-Standardroutinen.
        1000 : A=(WERT) : GOTO 20 : REM  (Name des Programmes).
               (WERT) = die Anzahl Zeichen, die für Strings re-
               serviert werden müssen. Computer die das benöti-
               gen dimensionieren in Zeile 20 den Speicherraum
               Danach geht das Programm in Zeile 1010 weiter.
        1010 : erste frei zu belegende Zeile des Programmes.
1010 - 32776 : das eigentliche Programm. Die Einteilung ist im
               Prinzip frei, es ist aber ratsam, eine bestimmte
               Systematik zu verwenden. Es ist dann für den An-
               wender besser zu verstehen und leichter Anzupas-
               sen. Wir stellen uns die Einteilung etwa so vor:
 1000 - 19999 : Das Hauptprogramm.
20000 - 24999 : Notwendige Subroutinen, die in BASICOD verboten
                sind. Blinkende Texte, Farbe, Musik, DOS-Anwei-
                sungen u.s.w. Sie können vom Anwender dann lei-
                chter gefunden und geändert werden.
25000 - 29999 : Zeilen mit DATA-Statements.
30000 - 32767 : Zeilen mit REM-Statements. Hier sollte eine Be-
                schreibung des Programms, Anweisungen und als
                letztes Ihr Namen und Adresse stehen.
```

3.) Zugelassene BASIC-Statements und OPERATOREN:

ABS	AND	ASC	ATN	CHR$	COS
DATA	DIM	END	FOR	GOSUB	GOTO
IF	INPUT	INT	LEFT$	LEN	LET
LOG	MID$	NEXT	NOT	ON	OR
PRINT	READ	REM	RESTORE	RETURN	RIGHT$
RUN	SGN	SIN	SQR	STEP	STOP
TAB	TAN	THEN	TO	VAL	

+	-	*	/	↑	=	<	>	<>	<=	>=	

4.) Vereinbarung zur Verwendung von VARIABLEN:

A. Numerische Variablen sind "single precision" und rechnen in der Praxis auf nicht mehr als 6 Ziffern.
B. Namen von Variablen sind maximal 2 Zeichen lang, wovon das erste ein Buchstabe sein muß, aber nicht "O (OTTO)" sein darf. Für Stringvariablen wird der Name durch das $ Zeichen ergänzt. Kleinbuchstaben sind nicht zugelassen.
C. Stringvariablen dürfen maximal 255 Zeichen lang sein.
D. Ausgeschlossen sind Variablen die mit O beginnen und:
   AS AT DS DS$ FN GR IF PI ST TI TI$ TO
E. Zur Kommunikation mit Standardsubroutinen werden folgende Variablen gebraucht: HO VE FR SR CN CT RV IN$ SR$

K U R S E K U R S E K U R S E

BASIC 1 Level 2 + Genie

Einführung in die Programmiersprache BASIC und die Arbeitsweise eines Mikrocomputers.
Vermittlung von Grundkenntnissen des BASIC. Einsteiger, auch ohne Vorkenntnisse haben die Möglichkeit, anhand der erworbenen Kenntnisse eigene Programme zu erstellen.

8 x 2 Std. nach Vereinbarung

Kursgebühren DM 120.--
Höchstteilnehmerzahl 6

BASIC 2 Level 2 + Genie

Der Kurs wendet sich an alle Anwender welche über Grundkenntnisse der BASIC verfügen und diese erweitern wollen.
Behandelt werden Fragen, die den Anwender befähigen sollen, seine Kenntnisse umfassender und effektiver einzusetzen.
Möglichkeiten der BASIC bei Diskettenbetrieb - Tricks und Kniffe zum beschläunigen von Programmen - Einsatz externer Geräte mit BASIC-Befehlen usw.

8 x 2 Std. nach Vereinbarung
Kursgebühren DM 120.--

Höchstteilnehmerzahl 6

TRS-DOS + NEWDOS/80 HANDLING

Der Kursus wendet sich an alle Anwender die sich mit den Nutzungsmöglichkeiten der beiden Betriebssysteme vertraut machen wollen. Neben dem Handling von TRS-DOS u. NEWDOS wird auch das Arbeiten mit den folgenden Zusatzprogrammen geübt.:

SUPERZAP
Für Änderung im Betriebssystem und Programmen in Z80 Assembler

DISASSEM
Zur Rückübersetzung von Z 80 Maschienencode und Erstellung von Crossreferenzen.

LMOFFSET
Zur Änderung des Ladebereiches von u. auf Diskette u. Kassette

EDIT-ASSEMBLER
Zum Schreiben und Ändern von Assemblerprogrammen.
Die weiteren Nebenprogramme von NEWDOS/80 werden miterfaßt.

8 x 2 Std. nach Vereinbarung
Kursgebühren DM 120.--

Höchstteilnehmerzahl 6

Z-80 ASSEMBLER - EINFÜHRUNG IN DIE MASCHIENENSPRACHE

Preis und Termine sind offen, bei genügendem Interesse kann ein Kurs abgehalen werden.

Teilnehmer der Kurse, BASIC 2 und DOS-Handling, die über kein eigenes Gerät verfügen, können sich einmal jede Woche, zum üben bei dem MC-Freundeskreis Köln treffen. (Nach Terminabsprache)

T E R M I N E & P R E I S E

Termine sind offen, vorbehaltlich einer ausreichenden Belegung. Sie werden mit den Teilnehmern abgesprochen und deren Verbindlichkeit frühzeitig mitgeteilt. Die Preise sind bindend. Nach bekanntgabe der Termine wird eine Anzahlung von 50 % der Kursgebühren erhoben. Bei Nichtteilnahme an vereinbarten Kursen kann die Anzahlung einbehalten werden. Alle Preise gelten für Mitglieder. Nichtmitglieder zahlen einen Zuschlag von 50 %.

KLEINANZEIGEN!

EDIT-ASSEMBLER-HANDBUCH
in deutscher übersetzung
für TANDY TRS-80 Edit-
Assembler Anwender.
DM 15.--. Leo. Drossert
Kalker Hauptstraße 189
5000 K ö l n 91

64K Interface Disketten-
laufwerke (mit Kontroler)
und gebr. Matrixdrucker
für GENIE I sof. gesucht
R. Schröder 500 Köln-41
Tel. 0221 - 7901693

Farbmonit. f. COLOR-GENIE
gebraucht gesucht. Gerät
mit eingeb. VHF-UHF Teil
bevorzugt.
Dagmar Adler
5000 Köln 30
Steinkauzweg 12

Änderungen in Programmen
Basic / Pascal / Assembl.
nach individuellem Wunsch
Klaus Schmidt-Trenk
Traubenstrasse 55
7000 Stuttgart 1

Wegen Systemaufgabe ca.
200 Programme (UTILITIS
GESCHÄFTSPROGRAMME SPIELE
und ADWENTURES) auf ca.
30 Disketten G-II SD/SS
Preis Verhandlungssache.
Eduard Mulders
Dreslinger Straße 11
5226 Reichshof-Denklingen

Profiprogramm für Werbe-
Schreiben, mit Adress-
datenpflege, Etiketten-
druck und Standardbrief-
speicherung. usw. usw.
Auch für Textverarbeitung
geeignet. DM 380.--
Willi Johnen, 02421/51376
Hansemannstr.1, 560 Düren

TAUSCHE 64 K GENIE-I evtl.
mit MONITOR gegen 32-K
COLOR-GENIE
Jürgen Kuschel. Eichstr. 9
5000 Köln 60 - 0221/779181

Programme für den Keramik-
bereich: Glasurerstellung,
Brennofenkalkulationen,
Proportionsberechnungen
für figürliches Arbeiten
u.s.w. ILSE BERNDT-JOCHUM
Stachelsgut 24
5060 Bergisch Gladbach 1
Keramische Lehrkurse:
Aufbau, Drehen, figürliches
Arbeiten. Information auf
Anfrage. 02204 / 65254

GENIE I, Zusatzeprom, 48 K
Grünmonitor, Doppelfloppy
mit 1 Laufwerk und EPSON-
Drucker mit Papier sowie
diverse Software außer G-1
alles neu, wegen Geschäfts-
aufgabe zu verkaufen.
Jörg + Waltraut Hildebrandt
Rote Erde 11 - 02426/5328
5164 Nörvenich-Eschweiler

Wir suchen für unsere Mit-
glieder mehrere gebrauchte
Floppylaufwerke (SD + DD)
Einzel- und Doppelstation
mit und ohne Gehäuse.
USERCLUB MC-FREUNDE KÖLN
Kalker Hauptstraße 189

80 MICRO Fachzeitschrift
von 1980 und 81 (einzeln
oder Gesamtausgabe) sucht:
Hans Otto Langguth
KÖLN 0221 - 556643

Auf dieser Seite können Sie alles Anbieten oder Suchen, z.B. Hardware, Programmierarbeiten oder auch Privat- und Geschäfts-anzeigen. Bitte fassen Sie sich Kurz!
Für gewerbliche Anzeigen überlassen wir es unseren Mitgliedern, uns den Wert Ihrer Anzeige nach eigenem Ermessen zu vergüten.