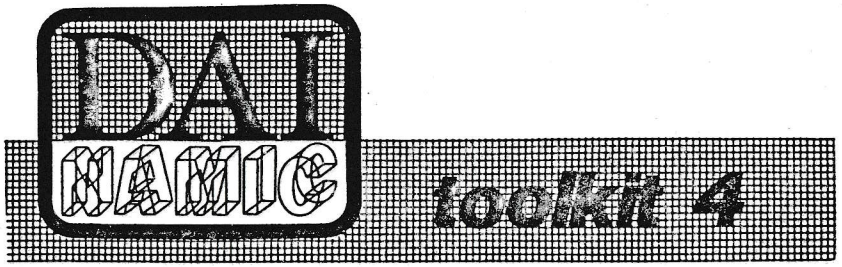
DAI NAMIC toolkit 4

toolkit 4

including tk.1A

```
1-ARR - TOOLKIT 4
2-BAS - BOOTSTRAP SYSTEM-PROGRAM-UNIT
3-OBJ -SYSTEM-PROGRAM-UNIT mlp
4-BAS - SYSTEM-PROGRAM-UNIT BASIC start & go
5-OBJ -2F0 400 SCRN COPY EPSON 5 GREY SCALE 180-82-1
6-OBJ -2F0 400 SCRN COPY EPSON 5 GREY SCALE MX82/100
7-ASS -SCREENCOPY MODE 1-8 5 GRIJSTINTEN
8-OBJ -300 40F SRCN COPY 9 GREY SCALE MX80-82-100
9-OBJ -300 41F SCRN COPY 9 GREY SCALE MX82/100
10-ASS -SOURCE SCRN COPY 9  GREY SCALE MX82/100
11-ASS - ARRAYDATA
12-OBJ - ARRAYDATA 298-41F
13-BAS - ARRAY-DATA DEMO 1
14-BAS - ARRAY-DATA DEMO 2
15-BAS - ARRAY-DATA DEMO 3
16-ASS - EDITDATA
17-OBJ - EDITDATA 298-410
18-BAS - EDIT-DATA DEMO 1
19-BAS - EDIT-DATA DEMO 2
20-BAS - SCRN COPY SEIKOSHA 5/6 RS-232
21-BAS - SCRN COPY SEIKOSHA 7/8 RS-232
22-ARR -
```



toolkit 4

including tk.1A

```
 1-ARR - TOOLKIT 4
 2-BAS - BOOTSTRAP SYSTEM-PROGRAM-UNIT
 3-OBJ -SYSTEM-PROGRAM-UNIT alp
 4-BAS - SYSTEM-PROGRAM-UNIT BASIC start & go
 5-OBJ -2F0 400 SCRN COPY EPSON 5 GREY SCALE (80-82-1
 6-OBJ -2F0 400 SCRN COPY EPSON 5 GREY SCALE MX82/100
 7-ASS -SCREENCOPY MODE 1-8 5 GRIJSTINTEN
 8-OBJ -300 40F SRCN COPY 9 GREY SCALE MX80-82-100
 9-OBJ -300 41F SCRN COPY 9 GREY SCALE MX82/100
10-ASS -SOURCE SCRN COPY 9  GREY SCALE MX82/100
11-ASS - ARRAYDATA
12-OBJ - ARRAYDATA 29B-41F
13-BAS - ARRAY-DATA DEMO 1
14-BAS - ARRAY-DATA DEMO 2
15-BAS - ARRAY-DATA DEMO 3
16-ASS - EDITDATA
17-OBJ - EDITDATA 29B-410
18-BAS - EDIT-DATA DEMO 1
19-BAS - EDIT-DATA DEMO 2
20-BAS - SCRN COPY SEIKOSHA 5/6 RS-232
21-BAS - SCRN COPY SEIKOSHA 7/8 RS-232
22-ARR -
```

TOOLKIT 4

HOW TO USE THE PROGRAMS

1/ SYSTEM PROGRAM UNIT

a) for NEW programs

Position tape before file 2 ,"BOOTSTRAP SYSTEM-PROGRAM-UNIT", and type LOAD:RUN. SYSTEM-PROGRAM-UNIT is loaded, heap-pointers are adapted and an introduction program is started. Now you can start programming, and call the routines from SPU with CALLM2000:.... , followed by operation command codes.

b) for OLD programs

Adapt Heappointers :

```
POKE #29C,#11
CLEAR 1000
```

Position tape before file 3 ,"SYSTEM-PROGRAM-UNIT mlp",
type UT <return>
Z3 <return>
R <return>
start tape and Read SPU mlp-file
type B

Load your BASIC program and call SPU with CALLM2000:....,
followed by operation commands.

heide 4 3171 westmeerbeek belgium
copyright dainamic — tel 016/69.86.23

system-program-unit

SYSTEM - PROGRAM - UNIT contains in one machine language part following routines :

* renumber	- step - mode - add - mode - sub - mode
* rename variables	- global - with line-restrictions
* variablen-atlas	- V list - Q list (jump-table)
* check	- number of lines - BASIC : size - number of symbols - size of symbol table

The different routines are called with :

CALLM2000: ..... : ..... : ......

followed by normal BASIC-commands.

If a BASIC-command after CALLM2000 is not recognised by SPU, action is transferred to normal BASIC-execution.

SPU - commands :

LET OLD = NEW	rename variable
LET OLD = NEW : OUT 100,500	rename only in lines 100-500
RESTORE	CLEAR SYMBOL-TABLE
LIST	Q-list of total program
LIST 100	Q-list of line 100
LIST 100-500	Q-list from 100-500
MODE 0	variablen-atlas
MODE 1	variablen-atlas with line-numbers
WAIT start,end,new	renumber in STEP-mode
WAITMEM start,end,add	renumber in ADD-mode
DOT start,end,sub	renumber in SUB-mode
CHECK	programm-check-list

1

```
10    REM ********************************
11    REM *  SYSTEM - PROGRAM - UNIT  *
12    REM *                           *
13    REM *                      DEMO  *
14    REM ********************************
15    MODE 6A:COLORG 0 5 10 15
20    W=RND(5.0)
21    ON W GOSUB 100,101,102,103
23    PRINT A,B,C," SUM : ";A+B+C
25    DRAW 50,0 50,A 21
26    DRAW 100,0 100,B 22
27    DRAW 150,0 150,C 23
30    IF A>200 OR B>200 OR C>200 THEN 500
36    WAIT TIME 20:GOTO 20
100   A=A+0.5:RETURN
101   B=B+0.5:RETURN
102   C=C+0.5:RETURN
103   D=D+0.5:RETURN
500   FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:C=0.0:GOTO 20
```

2

```
CALLM2000:WAIT 10,500,10
```

```
10    REM ********************************
20    REM *  SYSTEM - PROGRAM - UNIT  *
30    REM *                           *
40    REM *                      DEMO  *
50    REM ********************************
60    MODE 6A:COLORG 0 5 10 15
70    W=RND(5.0)
80    ON W GOSUB 150,160,170,180
90    PRINT A,B,C," SUM : ";A+B+C
100   DRAW 50,0 50,A 21
110   DRAW 100,0 100,B 22
120   DRAW 150,0 150,C 23
130   IF A>200 OR B>200 OR C>200 THEN 190
140   WAIT TIME 20:GOTO 70
150   A=A+0.5:RETURN
160   B=B+0.5:RETURN
170   C=C+0.5:RETURN
180   D=D+0.5:RETURN
190   FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:C=0.0:GOTO 70
```

3

```
*CALLM2000:CHECK
```

```
Datum :          Programm :

Lines   : 19
BASIC   : 502   Bytes
Symbols : 6
Table   : 43   Bytes
```

system-program-unit

1 - the original dummy program
2 - renumber in step mode
3 - CHECK : programm information

4

```
CALLM2000:WAITMEM 150,190,1000
```

```
10    REM ********************************
20    REM *  SYSTEM - PROGRAM - UNIT  *
30    REM *                           *
40    REM *                      DEMO  *
50    REM ********************************
60    MODE 6A:COLORG 0 5 10 15
70    W=RND(5.0)
80    ON W GOSUB 1150,1160,1170,1180
90    PRINT A,B,C," SUM : ";A+B+C
100   DRAW 50,0 50,A 21
110   DRAW 100,0 100,B 22
120   DRAW 150,0 150,C 23
130   IF A>200 OR B>200 OR C>200 THEN 1190
140   WAIT TIME 20:GOTO 70
1150  A=A+0.5:RETURN
1160  B=B+0.5:RETURN
1170  C=C+0.5:RETURN
1180  D=D+0.5:RETURN
1190  FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:C=0.0:GOTO 70
```

5

```
*CALLM2000:MODE 0
```

```
Datum :          Programm :

A !
B !
C !
D !
W !
X %
```

6

```
*CALLM2000:MODE 1
```

```
Datum :          Programm :

A !  90,100,130,1150,1190,
B !  90,110,130,1160,1190,
C !  90,120,130,1170,1190,
D !  1180,
W !  70,80,
X %
```

4 - renumber in ADD-mode
5 - V-list : variables-atlas
6 - C-LIST : cross-reference

7 *CALLM2000:LIST

Datum : Programm :

```
  70:  140,1190,
1150:  80,
1160:  80,
1170:  80,
1180:  80,
1190:  130,
```

8 *CALLM2000:LIST1190

Datum : Programm :

```
1190:  130,
```

7 - JUMP-TABLE of entire programm
8 - where is a jump/call to 1190?
9 - rename 'W' to 'RANDOMVALUE'
10 - rename 'C' to 'THIRDBAR'

9 *CALLM2000:LET W = RANDOMVALUE

```
*LIST
10    REM ********************************
20    REM *  SYSTEM - PROGRAM - UNIT  *
30    REM *                           *
40    REM *                      DEMO  *
50    REM ********************************
60    MODE 6A:COLORG 0 5 10 15
70    RANDOMVALUE=RND(5.0)
80    ON RANDOMVALUE GOSUB 1150,1160,1170,1180
90    PRINT A,B,C," SUM : ";A+B+C
100   DRAW 50,0 50,A 21
110   DRAW 100,0 100,B 22
120   DRAW 150,0 150,C 23
130   IF A>200 OR B>200 OR C>200 THEN 1190
140   WAIT TIME 20:GOTO 70
1150  A=A+0.5:RETURN
1160  B=B+0.5:RETURN
1170  C=C+0.5:RETURN
1180  D=D+0.5:RETURN
1190  FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:C=0.0:GOTO 70
```

*CALLM2000:LET C=THIRDBAR

10 *LIST

```
10    REM ********************************
20    REM *  SYSTEM - PROGRAM - UNIT  *
30    REM *                           *
40    REM *                      DEMO  *
50    REM ********************************
60    MODE 6A:COLORG 0 5 10 15
70    RANDOMVALUE=RND(5.0)
80    ON RANDOMVALUE GOSUB 1150,1160,1170,1180
90    PRINT A,B,THIRDBAR," SUM : ";A+B+THIRDBAR
100   DRAW 50,0 50,A 21
110   DRAW 100,0 100,B 22
120   DRAW 150,0 150,THIRDBAR 23
130   IF A>200 OR B>200 OR THIRDBAR>200 THEN 1190
140   WAIT TIME 20:GOTO 70
1150  A=A+0.5:RETURN
1160  B=B+0.5:RETURN
1170  THIRDBAR=THIRDBAR+0.5:RETURN
1180  D=D+0.5:RETURN
1190  FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:THIRDBAR=0.0:GOTO 70
```


11 *CALLM2000:MODE 1

Datum : Programm :

```
A !
        90,100,130,1150,1190,
B !
        90,110,130,1160,1190,
C !
D !
        1180,
RANDOMVALUE !
        70,80,
THIRDBAR !
        90,120,130,1170,1190,
W !
X %
*
*
```

12 *CALLM2000:RESTORE

Datum : Programm :

```
A !
        90,100,130,1150,1190,
B !
        90,110,130,1160,1190,
D !
        1180,
RANDOMVALUE !
        70,80,
THIRDBAR !
        90,120,130,1170,1190,
*
```

13 *CALLM2000:WAIT10,1190,2000

```
RENUMBER RANGE  : 10-1190
NEW LINE NUMBER : 2000-2180
*
*LIST
2000  REM ********************************
2010  REM *  SYSTEM - PROGRAM - UNIT  *
2020  REM *                           *
2030  REM *                      DEMO  *
2040  REM ********************************
2050  MODE 6A:COLORG 0 5 10 15
2060  RANDOMVALUE=RND(5.0)
2070  ON RANDOMVALUE GOSUB 2140,2150,2160,2170
2080  PRINT A,B,THIRDBAR," SUM : ";A+B+THIRDBAR
2090  DRAW 50,0 50,A 21
2100  DRAW 100,0 100,B 22
2110  DRAW 150,0 150,THIRDBAR 23
2120  IF A>200 OR B>200 OR THIRDBAR>200 THEN 2180
2130  WAIT TIME 20:GOTO 2060
2140  A=A+0.5:RETURN
2150  B=B+0.5:RETURN
2160  THIRDBAR=THIRDBAR+0.5:RETURN
2170  D=D+0.5:RETURN
2180  FILL 0,0 XMAX,YMAX 20:A=0.0:B=0.0:THIRDBAR=0.0:GOTO 2060
```

11 - 'W' & 'RANDOMVALUE' are in the symbol table
'C' & 'THIRDBAR' are in the symbol table
12 - RESTORE : only actual symbols are preserved in the symbol table
13 - again RENUMBER in step-mode

HOW TO USE THE PROGRAMS

2/ EPSON SCREEN COPIES (5 & 9 grey scales)

general information

For this Jumbo size screen copies, the video ram is scanned from right to left. This technique is used because the horizontal resolution of MODE 5/6, combined with the expression of brightness to represent colours, exceeds the maximum number of dots/line in bit image printing.

MX-80 : normal density : 480 dots/line
dual density : 960 dots/line

MX-82 : normal density : 576 dots/line
dual density : 1152 dots/line

MX-100: there is no need for restriction on MX-100, but the same technique of MX-82 has been used.

restrictions for MX-80

- normal density bit graphics (5 grey scale)

5 grey scale copy takes 2*2 dots for each pixel,so the number of dots/line should be 256*2=512. This is outside the range of MX-80, to solve this problem, 16 lines of the screen are not printed.(Y 0-15)

- dual density bit graphics (9 grey scale)

9 grey scale copy takes 4*2 dots for each pixel, so the number of dots/line should be 256*4=1024. This is outside the range of MX-80, again 16 lines of the screen are not printed.(Y 241-256)

COLOUR/BRIGHTNESS TABLE

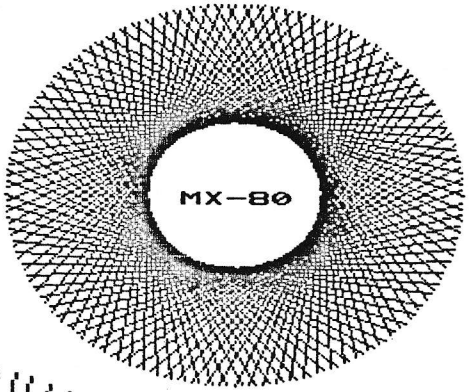
COLOUR Nr	5 grey scale	9 grey scale
0	0	0
1	1	1
2	1	1
3	1	2
4	2	2
5	2	3
6	2	3
7	2	4
8	3	4
9	3	5
10	3	5
11	3	6
12	4	6
13	4	7
14	4	7
15	4	8

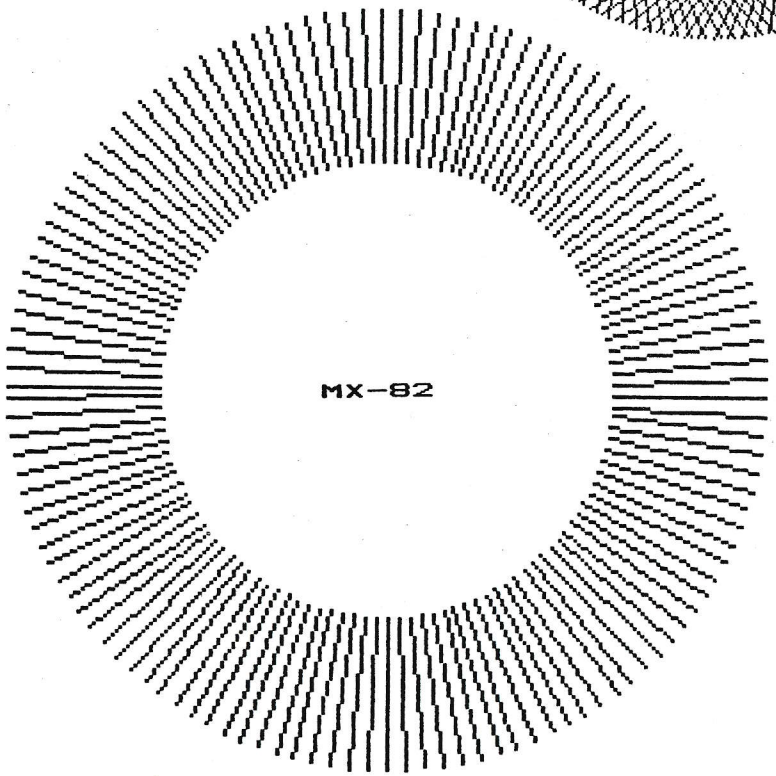
general information (cont.)

proportion

From the EPSON MX-series, only MX-82 gives 1/1 proportion,thus circles on screen are round on paper.
MX-80 and MX-100 give 1/1.2 proportion.

see following examples :





timing

If timing on your interface is critical,you will have to increase the delay value as indicated in the source-listings.

5 grey MX-80 : standard value in address #3D5 : #40
to increase : POKE #3D5,#45 .... #99
(With a higher value in address #3D5, screen copy will take more time)

5 grey MX-82 : idem

9 grey MX-80 : standard value in address #3ED : #40
to increase : POKE #3ED,#45 .... #99

9 grey MX-82 : standard value in address #3F1 : #40
to increase : POKE #3F1,#45 ... #99

LOAD & RUN

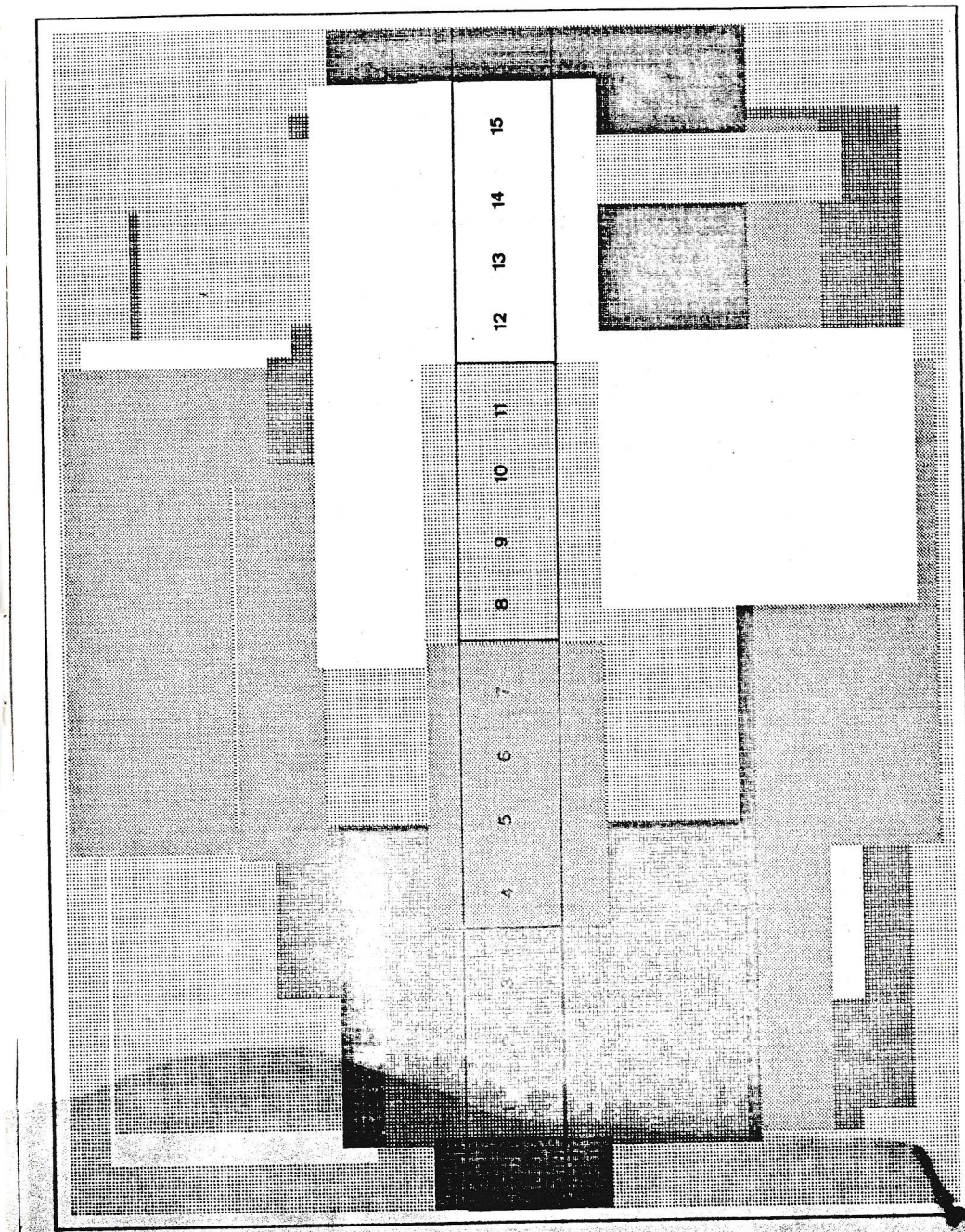
Adapt Heappointers above screen copy routines:

```
POKE #29B,#10:POKE #29C,4
CLEAR 1000

type UT <return>
     Z3 <return>
position tape before mlp routine
type R  <return>
     B

Call screen copy routine with :

CALLM#300
```

SCREEN COPY 5 GREY SCALE (EPSON)

5 grey scale

```
                                                                    5
02F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0300 F5 C5 D5 E5 21 00 00 01 00 00 EF 27 78 FE FF DA
0310 14 03 3E FF 32 F9 02 EB CD 97 03 3A F9 02 4F E5
0320 21 00 00 E3 06 04 C5 0D EF 27 0C C1 FE 00 C2 37
0330 03 11 03 03 C3 5B 03 FE 04 D2 42 03 11 01 03 C3
0340 5B 03 FE 08 D2 4D 03 11 01 02 C3 5B 03 FE 0C D2
0350 5B 03 11 00 01 C3 5B 03 11 00 00 E3 29 29 19 E3
0360 2B 05 C2 26 03 E3 7C CD D3 03 7D CD D3 03 E1 11
0370 04 00 19 0D C2 1F 03 11 FC FF 19 DA 18 03 3E 0A
0380 CD D3 03 3E 0D CD D3 03 3E 1B CD D3 03 3E 4E CD
0390 D3 03 E1 D1 C1 F1 C9 3E 1B CD D3 03 3E 4F CD D3
03A0 03 3E 0A CD D3 03 3E 0D CD D3 03 3E 1B CD D3 03
03B0 3E 41 CD D3 03 3E 08 CD D3 03 3E 1B CD D3 03 3E
03C0 4B CD D3 03 E5 2A F9 02 29 7D CD D3 03 7C CD D3
03D0 03 E1 C9 F5 3E (46) 3D C2 D6 03 F3 3A 00 FD E6 08
03E0 CA DB 03 3A 03 FF E6 10 CA E3 03 F1 32 06 FF FB
03F0 C9 29 19 19 19 11 B9 40 19 EB 21 F0 02 7B 96 23
0400 FF
```

MX80

increase this value if RS-232 causes problems

```
                                                                    5
02F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0300 F5 C5 D5 E5 21 00 00 01 00 00 EF 27 78 FE FF DA
0310 14 03 3E FF 32 F9 02 EB CD 97 03 3A F9 02 4F E5
0320 21 00 00 E3 06 04 C5 0D EF 27 0C C1 FE 00 C2 37
0330 03 11 03 03 C3 5B 03 FE 04 D2 42 03 11 01 03 C3
0340 5B 03 FE 08 D2 4D 03 11 01 02 C3 5B 03 FE 0C D2
0350 5B 03 11 00 01 C3 5B 03 11 00 00 E3 29 29 19 E3
0360 2B 05 C2 26 03 E3 7C CD D3 03 7D CD D3 03 E1 11
0370 04 00 19 0D C2 1F 03 11 FC FF 19 DA 18 03 3E 0A
0380 CD D3 03 3E 0D CD D3 03 3E 1B CD D3 03 3E 4E CD
0390 D3 03 E1 D1 C1 F1 C9 3E 1B CD D3 03 3E 4F CD D3
03A0 03 3E 0A CD D3 03 3E 0D CD D3 03 3E 1B CD D3 03
03B0 3E 41 CD D3 03 3E 08 CD D3 03 3E 1B CD D3 03 3E
03C0 4B CD D3 03 E5 2A F9 02 29 7D CD D3 03 7C CD D3
03D0 03 E1 C9 F5 3E (46) 3D C2 D6 03 F3 3A 00 FD E6 08
03E0 CA DB 03 3A 03 FF E6 10 CA E3 03 F1 32 06 FF FB
03F0 C9 29 19 19 19 00 00 00 00 EB 21 F0 02 7B 96 23
0400 FF
```

MX82-100

increase this value if RS-232 causes problems


```
002                     *       ******************************************
003                     *       IN 5 GRIJSTINTEN YMAX=240 XMAX= ~
004                     *       A PROGRAM BY N.P.LOOIJE
005                     *       PALUDANUSHOF 22,3151 CM HOEK VAN HOLLAND
006                     *       ******************************************
007                     *       VOOR DE EPSON MX 80-II VIA RS 232 C
008                             ORG   :2F9
009 02F9 0000           YMAX    DBL   0
010                             ORG   :300
011 0300 F5                     PUSH  PSW
012 0301 C5                     PUSH  B
013 0302 D5                     PUSH  D
014 0303 E5                     PUSH  H
015 0304 210000                 LXI   H,0
016 0307 010000                 LXI   B,0
017 030A EF                     RST   5
018 030B 27                     DATA  :27       XMAX+YMAX
019 030C 78                     MOV   A,B
020 030D FEF0                   CPI   :F0       PRINTMAX =480(/2) PUNT
021 030F DA1403                 JC    LBL0
022 0312 3EF0                   MVI   A,:F0
023 0314 32F902         LBL0    STA   YMAX
024 0317 EB                     XCHG
025 0318 CD9703         XLOOP   CALL  CTRL$
026 031B 3AF902                 LDA   YMAX
027 031E 4F                     MOV   C,A
028 031F E5             YLOOP   PUSH  H
029 0320 210000                 LXI   H,0       CLEAR PRINTACCU
030 0323 E3                     XTHL
031 0324 0604                   MVI   B,4
032 0326 C5             NIBBLE  PUSH  B
033 0327 0D                     DCR   C
034 0328 EF                     RST   5
035 0329 27                     DATA  :27       SCRN(HL,C)
036 032A 0C                     INR   C
037 032B C1                     POP   B
038 032C FE00                   CPI   0
039 032E C23703                 JNZ   COL1      **
040 0331 110303                 LXI   D,:303    ** dot
041 0334 C35B03                 JMP   LBL1
042 0337 FE04           COL1    CPI   4
043 0339 D24203                 JNC   COL2      *-
044 033C 110103                 LXI   D,:301    ** dot
045 033F C35B03                 JMP   LBL1
046 0342 FE08           COL2    CPI   8
047 0344 D24D03                 JNC   COL3      *-
048 0347 110102                 LXI   D,:201    -* dot
049 034A C35B03                 JMP   LBL1
050 034D FE0C           COL3    CPI   :C
051 034F D25803                 JNC   COL4      --
052 0352 110001                 LXI   D,:100    -* dot
053 0355 C35B03                 JMP   LBL1      --
054 0358 110000         COL4    LXI   D,0       -- dot
```

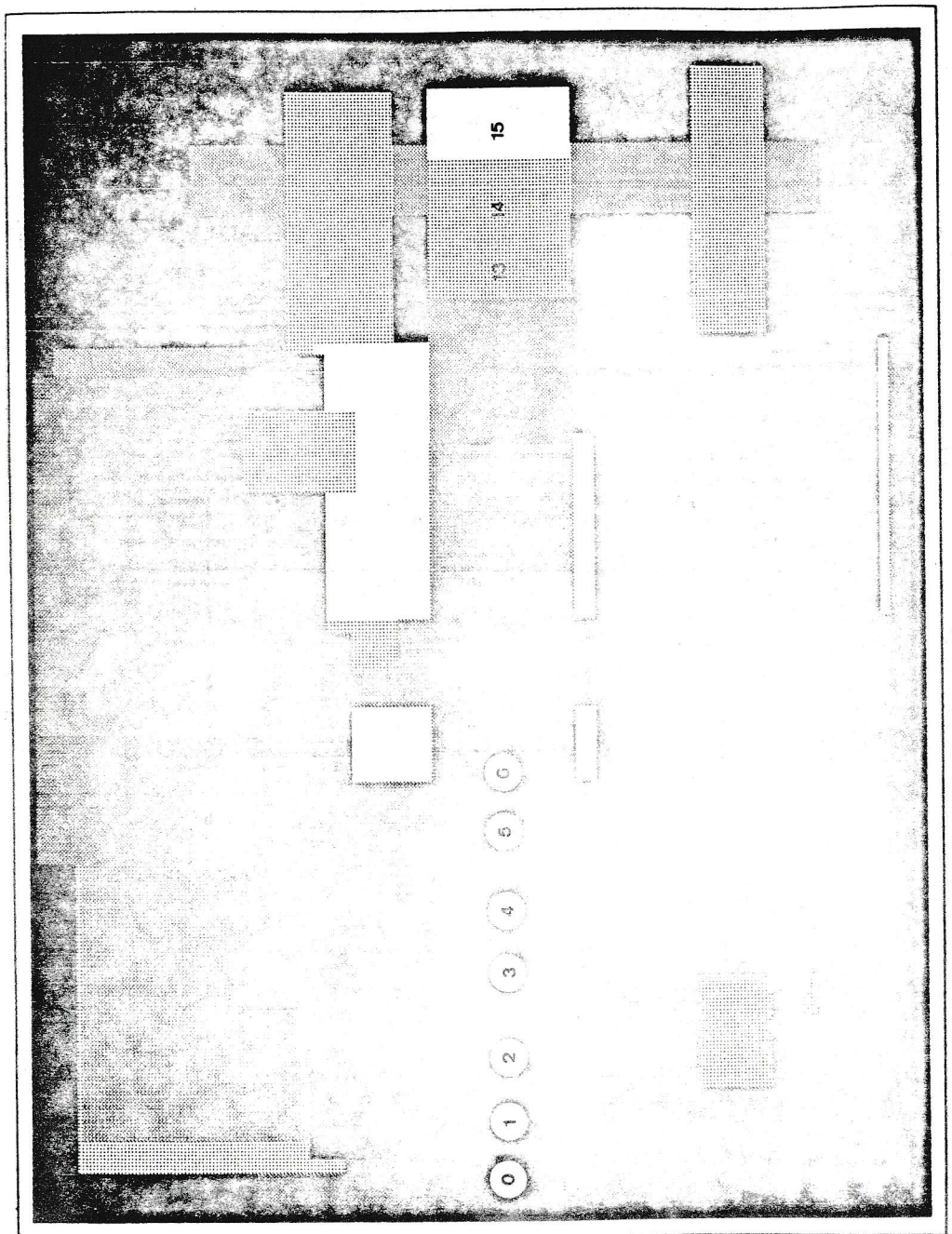
```
055 035B E3        LBL1   XTHL
056 035C 29               DAD   H
057 035D 29               DAD   H
058 035E 19               DAD   D
059 035F E3               XTHL
060 0360 2B               DCX   H
061 0361 05               DCR   B
062 0362 C22603           JNZ   NIBBLE
063 0365 E3               XTHL
064 0366 7C               MOV   A,H
065 0367 CDD303           CALL  PRINT
066 036A 7D               MOV   A,L
067 036B CDD303           CALL  PRINT
068 036E E1               POP   H
069 036F 110400           LXI   D,4
070 0372 19               DAD   D
071 0373 0D               DCR   C
072 0374 C21F03           JNZ   YLOOP
073 0377 11FCFF           LXI   D,:FFFC   = -4
074 037A 19               DAD   D
075 037B DA1803           JC    XLOOP
076 037E 3E0A             MVI   A,:A      LF
077 0380 CDD303           CALL  PRINT
078 0383 3E0D             MVI   A,:D      CR
079 0385 CDD303           CALL  PRINT
080 0388 3E1B             MVI   A,:1B     ESC set skip over
081 038A CDD303           CALL  PRINT
082 038D 3E4E             MVI   A,'N'
083 038F CDD303           CALL  PRINT
084 0392 E1        OUT    POP   H
085 0393 D1               POP   D
086 0394 C1               POP   B
087 0395 F1               POP   PSW
088 0396 C9               RET
089 0397 3E1B      CTRL$  MVI   A,:1B     release skipover
090 0399 CDD303           CALL  PRINT
091 039C 3E4F             MVI   A,'O'
092 039E CDD303           CALL  PRINT
093 03A1 3E0A             MVI   A,:A      LF
094 03A3 CDD303           CALL  PRINT
095 03A6 3E0D             MVI   A,:D      CR
096 03A8 CDD303           CALL  PRINT
097 03AB 3E1B             MVI   A,:1B     ESC linespacing 8/72
098 03AD CDD303           CALL  PRINT
099 03B0 3E41             MVI   A,'A'     A
100 03B2 CDD303           CALL  PRINT
101 03B5 3E08             MVI   A,8
102 03B7 CDD303           CALL  PRINT
103 03BA 3E1B             MVI   A,:1B     ESC single density graphics
104 03BC CDD303           CALL  PRINT
105 03BF 3E4B             MVI   A,'K'     K
106 03C1 CDD303           CALL  PRINT
107 03C4 E5               PUSH  H
108 03C5 2AF902           LHLD  YMAX
109 03C8 29               DAD   H
110 03C9 7D               MOV   A,L       Y MOD 256
111 03CA CDD303           CALL  PRINT
112 03CD 7C               MOV   A,H       Y/256
113 03CE CDD303           CALL  PRINT
114 03D1 E1               POP   H
115 03D2 C9               RET
116 03D3 F5        PRINT  PUSH  PSW       OUTPUT RS232C
117 03D4 3E30             MVI   A,:30
```



```
118 03D6 3D        DELAY  DCR   A
119 03D7 C2D603           JNZ   DELAY
120 03DA F3               DI
121 03DB 3A00FD    SERRDY LDA   :FD00
122 03DE E608             ANI   8
123 03E0 CADB03           JZ    SERRDY
124 03E3 3A03FF    BUFEMP LDA   :FF03
125 03E6 E610             ANI   :10
126 03E8 CAE303           JZ    BUFEMP
127 03EB F1        SEROUT POP   PSW
128 03EC 3206FF           STA   :FF06
129 03EF FB               EI
130 03F0 C9               RET
131 03F1                  END

**************************
* S Y M B O L   T A B L E *
**************************

BUFEMP 03E3   COL1   0337   COL2   0342   COL3   034D
COL4   0358   CTRL$  0397   DELAY  03D6   LBL0   0314
LBL1   035B   NIBBLE 0326   OUT    0392   PRINT  03D3
SEROUT 03EB   SERRDY 03DB   XLOOP  0318   YLOOP  031F
YMAX   02F9
```



CREEN COPY 9 GREY SCALE (EPSON)

9 grey scale

9

```
0300 F5 C5 D5 E5 21 00 00 01 00 00 EF 27 78 FE F0 DA
0310 14 03 3E F0 32 0D 04 EB CD AE 03 3A 0D 04 4F 06
0320 04 E5 21 00 00 22 09 04 22 0B 04 E1 C5 E5 0D EF
0330 27 11 00 00 01 00 00 3C E6 1E 21 43 03 85 6F 7C
0340 89 67 E9 0C 0C 1C 00 04 04 14 00 1C 1C 0C 00 14
0350 14 04 00 2A 09 04 29 29 09 22 09 04 2A 0B 04 29
0360 29 19 22 0B 04 E1 C1 2B 05 C2 2C 03 D5 E5 21 09
0370 04 7E CD EB 03 23 7E CD EB 03 23 7E CD EB 03 23
0380 7E CD EB 03 E1 11 04 00 19 D1 0D C2 1F 03 11 FC
0390 FF 19 DA 18 03 3E 0A CD EB 03 3E 0D CD EB 03 3E
03A0 1B CD EB 03 3E 4E CD EB 03 E1 D1 C1 F1 C9 3E 1B
03B0 CD EB 03 3E 4F CD EB 03 3E 0A CD EB 03 3E 0D CD
03C0 EB 03 3E 1B CD EB 03 3E 41 CD EB 03 3E 08 CD EB
03D0 03 3E 1B CD EB 03 3E 4C CD EB 03 E5 2A 0D 04 29
03E0 29 7D CD EB 03 7C CD EB 03 E1 C9 F5 3E 40 3D C2
03F0 EE 03 F3 3A 00 FD E6 08 CA F3 03 3A 03 FF E6 10
0400 CA FB 03 F1 32 06 FF FB C9 00 00 00 00 00 00 35
```

MX80

increase this value if RS-232 causes problems

9

```
0300 F5 C5 D5 E5 21 00 00 01 00 00 EF 27 78 FE FF C2
0310 18 03 3E 01 32 12 04 AF 32 11 04 EB CD B2 03 3A
0320 11 04 4F 06 04 E5 21 00 00 22 0D 04 22 0F 04 E1
0330 C5 E5 0D EF 27 11 00 00 01 00 00 3C E6 1E 21 47
0340 03 85 6F 7C 89 67 E9 0C 0C 1C 00 04 04 14 00 1C
0350 1C 0C 00 14 14 04 00 2A 0D 04 29 29 09 22 0D 04
0360 2A 0F 04 29 29 19 22 0F 04 E1 C1 2B 05 C2 30 03
0370 D5 E5 21 0D 04 7E CD EF 03 23 7E CD EF 03 23 7E
0380 CD EF 03 23 7E CD EF 03 E1 11 04 00 19 D1 0D C2
0390 23 03 11 FC FF 19 DA 1C 03 3E 0A CD EF 03 3E 0D
03A0 CD EF 03 3E 1B CD EF 03 3E 4E CD EF 03 E1 D1 C1
03B0 F1 C9 3E 1B CD EF 03 3E 4F CD EF 03 3E 0A CD EF
03C0 03 3E 0D CD EF 03 3E 1B CD EF 03 3E 41 CD EF 03
03D0 3E 08 CD EF 03 3E 1B CD EF 03 3E 4C CD EF 03 E5
03E0 2A 11 04 29 29 7D CD EF 03 7C CD EF 03 E1 C9 F5
03F0 3E 40 3D C2 F2 03 F3 3A 00 FD E6 08 CA F7 03 3A
0400 03 FF E6 10 CA FF 03 F1 32 06 FF FB C9 00 00 00
```

MX 82-100

increase this value if RS-232 causes problems

```
002                     *       ******************************************
003                     *       IN 9 GRIJSTINTEN YMAX=240(ZIE REM) XMAX= ~
004                     *       A PROGRAM BY N.P.LOOIJE
005                     *       PALUDANUSHOF 22,3151 CM HOEK VAN HOLLAND
006                     *       ******************************************
007                     *       VOOR DE EPSON MX 80-II (MX82,MX 100)VIA RS 2
008                             ORG   :300
009 0300 F5                     PUSH  PSW
010 0301 C5                     PUSH  B
011 0302 D5                     PUSH  D
012 0303 E5                     PUSH  H
013 0304 210000                 LXI   H,0
014 0307 010000                 LXI   B,0
015 030A EF                     RST   5
016 030B 27                     DATA  :27       XMAX+YMAX
017 030C 78                     MOV   A,B       MX 82/100 I.P.V. 18,19,20
018 030D FEF0                   CPI   :F0       CPI    :FF  /  JNZ  LBL0
019 030F DA1403                 JC    LBL0      MVI    A,1  /  STA  YMAX1
020 0312 3EF0                   MVI   A,:F0     XRA    A
021 0314 320D04         LBL0    STA   YMAX
022 0317 EB                     XCHG
023 0318 CDAE03         XLOOP   CALL  CTRL$     VOLGENDE NIBBLE
024 031B 3A0D04                 LDA   YMAX
025 031E 4F                     MOV   C,A
026 031F 0604           YLOOP   MVI   B,4       NIBBLECOUNT VOLGENDE Y
027 0321 E5                     PUSH  H         SAVE X
028 0322 210000                 LXI   H,0       CLEAR PRINTACCU
029 0325 220904                 SHLD  PRACH
030 0328 220B04                 SHLD  PRACL
031 032B E1                     POP   H         RETRIEVE X
032 032C C5             NIBBLE  PUSH  B         SAVE NIBBLECOUNT+Y
033 032D E5                     PUSH  H         SAVE X
034 032E 0D                     DCR   C         Y=Y-1
035 032F EF                     RST   5
036 0330 27                     DATA  :27       SCRN(HL,C)
037 0331 110000                 LXI   D,0       CLEAR PRINTDATALOW
038 0334 010000                 LXI   B,0       CLEAR PRINTDATAHIGH
039 0337 3C                     INR   A         INCR KLEUR
040 0338 E61E                   ANI   :1E       CLEAR LSB KLEUR
041 033A 214303                 LXI   H,COL00   )
042 033D 85                     ADD   L         )
043 033E 6F                     MOV   L,A       )
044 033F 7C                     MOV   A,H       )
045 0340 89                     ADC   C         )
046 0341 67                     MOV   H,A       )
047 0342 E9                     PCHL            )ON (KLEUR+1)/2 GOTO
048 0343 0C             COL00   INR   C
049 0344 0C                     INR   C
050 0345 1C             COL12   INR   E
051 0346 00                     NOP
052 0347 04             COL34   INR   B
053 0348 04                     INR   B
054 0349 14             COL56   INR   D
```

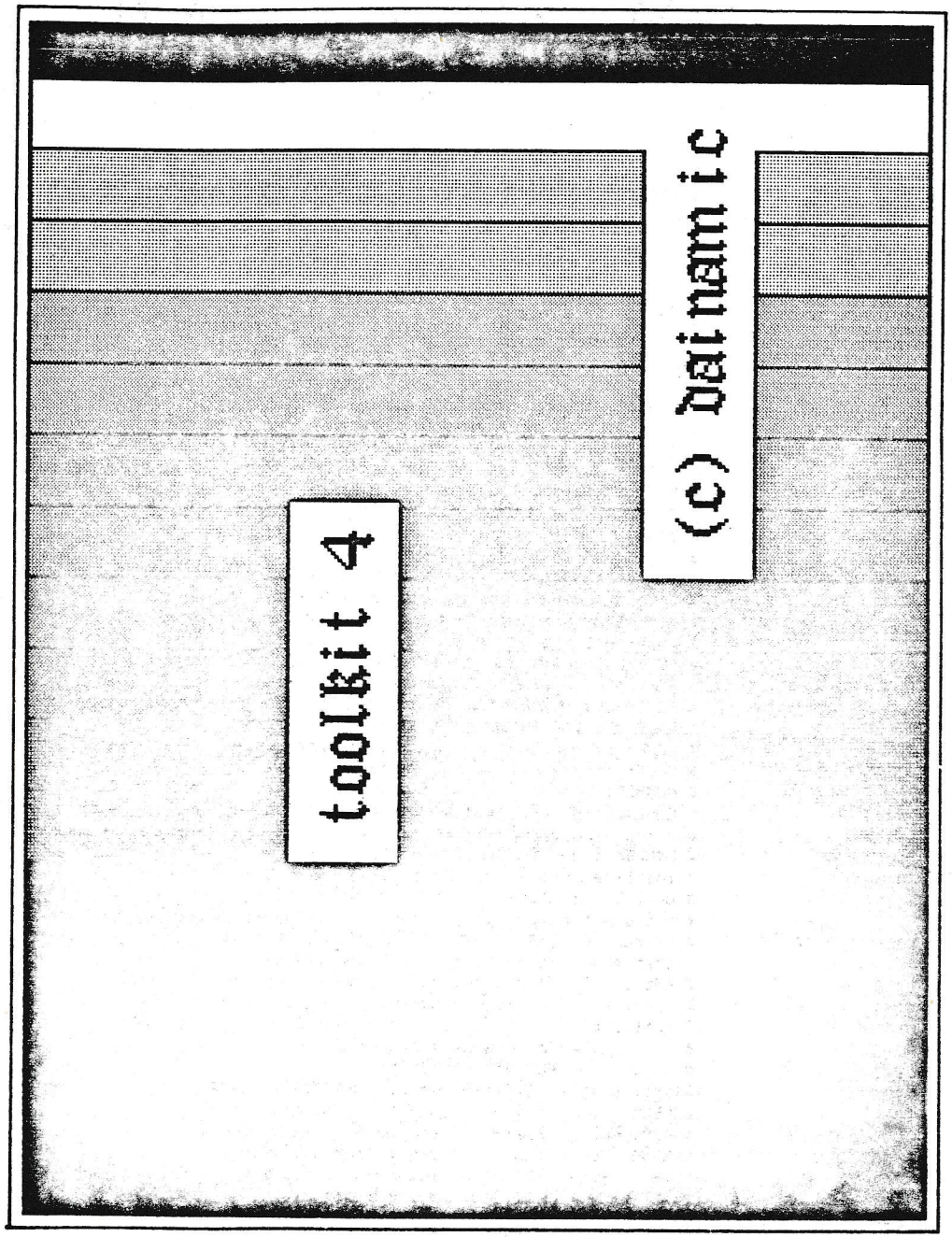


```
055 034A 00                     NOP
056 034B 1C             COL78   INR   E
057 034C 1C                     INR   E
058 034D 0C             COL9A   INR   C
059 034E 00                     NOP
060 034F 14             COLBC   INR   D
061 0350 14                     INR   D
062 0351 04             COLDE   INR   B
063 0352 00                     NOP
064 0353 2A0904         COLF    LHLD  PRACH     PRINTACCUHIGH+PRINTDATAHIGH
065 0356 29                     DAD   H
066 0357 29                     DAD   H
067 0358 09                     DAD   B
068 0359 220904                 SHLD  PRACH
069 035C 2A0B04                 LHLD  PRACL     PRINTACCULOW+PRINTDATALOW
070 035F 29                     DAD   H
071 0360 29                     DAD   H
072 0361 19                     DAD   D
073 0362 220B04                 SHLD  PRACL
074 0365 E1                     POP   H         RETRIEVE X
075 0366 C1                     POP   B         RETRIEVE Y+COUNT
076 0367 2B                     DCX   H
077 0368 05                     DCR   B
078 0369 C22C03                 JNZ   NIBBLE    NIBBLE<>0 REPEAT
079 036C D5                     PUSH  D         SAVE Y+COUNT
080 036D E5                     PUSH  H         SAVE X
081 036E 210904                 LXI   H,PRACH   DRUK PRINTACCU
082 0371 7E                     MOV   A,M       PRACHL
083 0372 CDEB03                 CALL  PRINT
084 0375 23                     INX   H
085 0376 7E                     MOV   A,M       PRACHH
086 0377 CDEB03                 CALL  PRINT
087 037A 23                     INX   H
088 037B 7E                     MOV   A,M       PRACLL
089 037C CDEB03                 CALL  PRINT
090 037F 23                     INX   H
091 0380 7E                     MOV   A,M       PRACLH
092 0381 CDEB03                 CALL  PRINT
093 0384 E1                     POP   H         RETRIEVE X
094 0385 110400                 LXI   D,4       X=X+4
095 0388 19                     DAD   D
096 0389 D1                     POP   D         RETRIEVE Y+COUNT
097 038A 0D                     DCR   C         Y=Y-1
098 038B C21F03                 JNZ   YLOOP     Y<>0 REPEAT
099 038E 11FCFF                 LXI   D,-4      X=X-4
100 0391 19                     DAD   D
101 0392 DA1803                 JC    XLOOP     X<>0 REPEAT
102 0395 3E0A                   MVI   A,:A      LF
103 0397 CDEB03                 CALL  PRINT
104 039A 3E0D                   MVI   A,:D      CR
105 039C CDEB03                 CALL  PRINT
106 039F 3E1B                   MVI   A,:1B     ESC set skip over
107 03A1 CDEB03                 CALL  PRINT
108 03A4 3E4E                   MVI   A,'N'
109 03A6 CDEB03                 CALL  PRINT
110 03A9 E1             OUT     POP   H
111 03AA D1                     POP   D
112 03AB C1                     POP   B
113 03AC F1                     POP   PSW
114 03AD C9                     RET
115 03AE 3E1B           CTRL$   MVI   A,:1B     release skipover
116 03B0 CDEB03                 CALL  PRINT
117 03B3 3E4F                   MVI   A,'O'
```

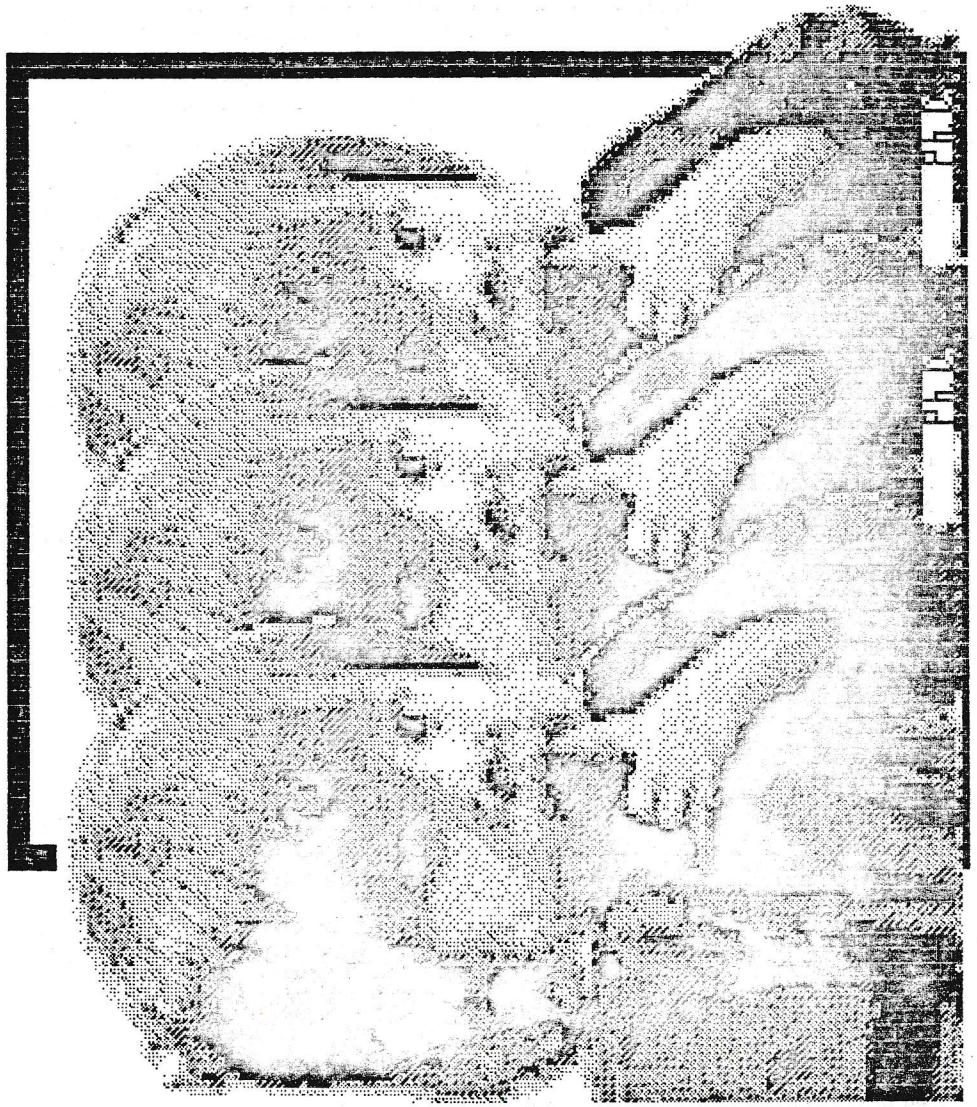
```
118 03B5 CDEB03              CALL  PRINT
119 03B8 3E0A                MVI   A,:A      LF
120 03BA CDEB03              CALL  PRINT
121 03BD 3E0D                MVI   A,:D      CR
122 03BF CDEB03              CALL  PRINT
123 03C2 3E1B                MVI   A,:1B     ESC linespacing 8/72
124 03C4 CDEB03              CALL  PRINT
125 03C7 3E41                MVI   A,'A'     A
126 03C9 CDEB03              CALL  PRINT
127 03CC 3E08                MVI   A,8
128 03CE CDEB03              CALL  PRINT
129 03D1 3E1B                MVI   A,:1B     ESC double density graphics
130 03D3 CDEB03              CALL  PRINT
131 03D6 3E4C                MVI   A,'L'     L
132 03D8 CDEB03              CALL  PRINT
133 03DB E5                  PUSH  H
134 03DC 2A0D04              LHLD  YMAX
135 03DF 29                  DAD   H
136 03E0 29                  DAD   H
137 03E1 7D                  MOV   A,L       Y MOD 256
138 03E2 CDEB03              CALL  PRINT
139 03E5 7C                  MOV   A,H       Y/256
140 03E6 CDEB03              CALL  PRINT
141 03E9 E1                  POP   H
142 03EA C9                  RET
143 03EB F5          PRINT   PUSH  PSW       OUTPUT RS232C
144 03EC 3E40                MVI   A,:40
145 03EE 3D          DELAY   DCR   A
146 03EF C2EE03              JNZ   DELAY
147 03F2 F3                  DI
148 03F3 3A00FD      SERRDY  LDA   :FD00
149 03F6 E608                ANI   8
150 03F8 CAF303              JZ    SERRDY
151 03FB 3A03FF      BUFEMP  LDA   :FF03
152 03FE E610                ANI   :10
153 0400 CAFB03              JZ    BUFEMP
154 0403 F1          SEROUT  POP   PSW
155 0404 3206FF              STA   :FF06
156 0407 FB                  EI
157 0408 C9                  RET
158 0409 0000        PRACH   DBL   :0
159 040B 0000        PRACL   DBL   :0
160 040D 00          YMAX    DATA  :0
161 040E 00          YMAX1   DATA  :0
162 040F                     END
```

```
**************************
* S Y M B O L   T A B L E *
**************************

BUFEMP 03FB   COL00  0343   COL12  0345   COL34  0347
COL56  0349   COL78  034B   COL9A  034D   COLBC  034F
COLDE  0351   COLF   0353   CTRL$  03AE   DELAY  03EE
LBL0   0314   NIBBLE 032C   OUT    03A9   PRACH  0409
PRACL  040B   PRINT  03EB   SEROUT 0403   SERRDY 03F3
XLOOP  0318   YLOOP  031F   YMAX   040D   YMAX1  040E
```

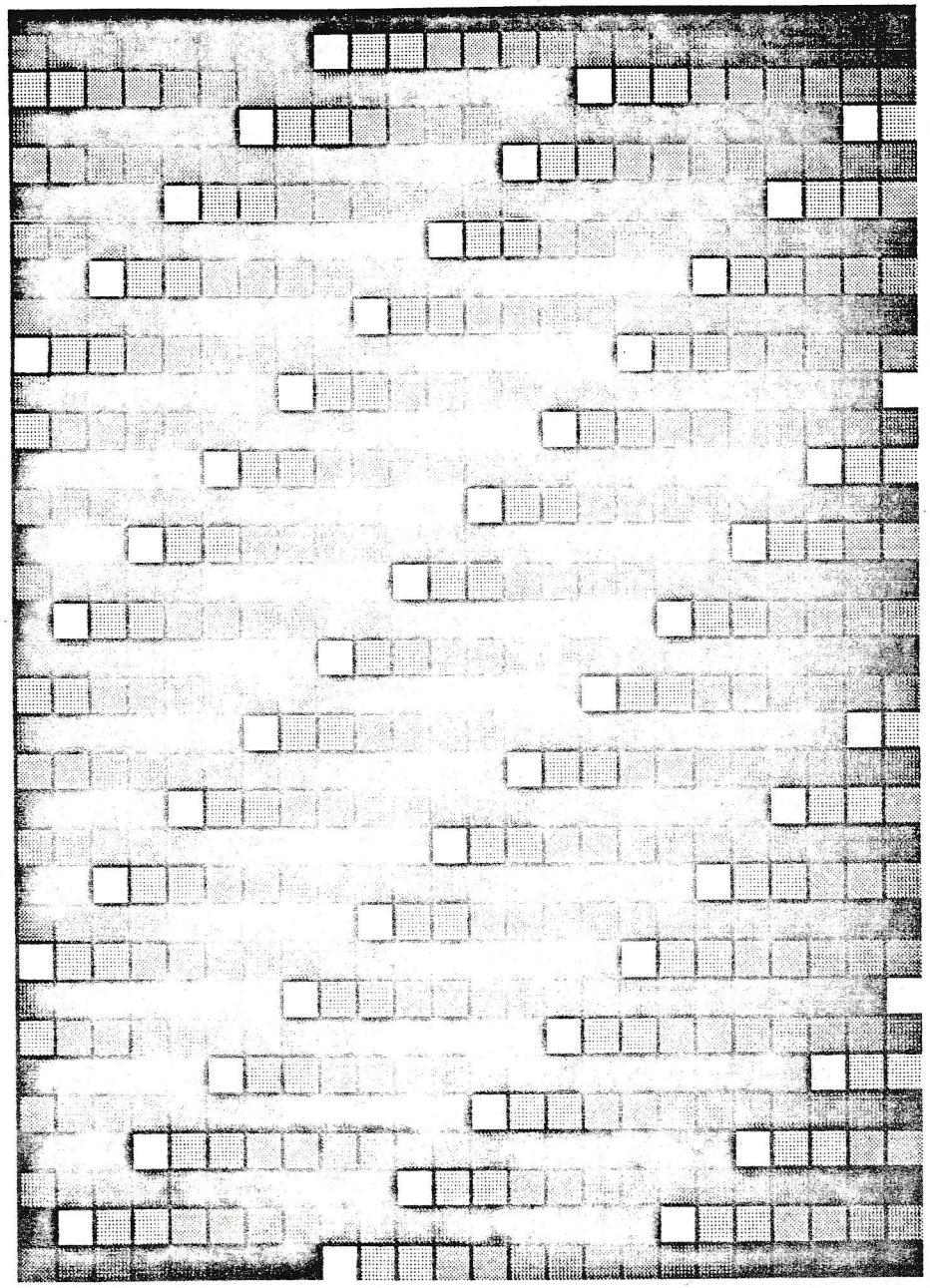
```
001                ***************************************************
002                *                                                 *
003                *  AAAA    RRRR    RRRR    AAAA   Y     Y    >    *
004                * A    A  R    R  R    R  A    A  Y     Y     >    *
005                * A    A  R    R  R    R  A    A   Y   Y       >   *
006                * AAAAAA  RRRRR   RRRRR   AAAAAA    Y Y   >>>>>>>  *
007                * A    A  R  R    R  R    A    A     Y          >   *
008                * A    A  R   R   R   R   A    A     Y         >    *
009                * A    A  R    R  R    R  A    A     Y        >     *
010                *                                                 *
011                * DDDDD    AAAA   TTTTTTT  AAAA                   *
012                * D    D  A    A     T    A    A                  *
013                * D    D  A    A     T    A    A                  *
014                * D    D  AAAAAA     T    AAAAAA                  *
015                * D    D  A    A     T    A    A                  *
016                * D    D  A    A     T    A    A       Markus Sigg *
017                * DDDDD   A    A     T    A    A            8/82 *
018                *                                                 *
019                ***************************************************
020                * This subroutine can be used to transfer data    *
021                * from $-arrays to data lines. The call of the    *
022                * routine looks as following:                     *
023                * POKE #300,LINE MOD 256:POKE #301,LINE/256:      *
024                * POKE #302,STEP MOD 256:POKE #303,STEP/256:      *
025                * POKE #304,NUMBER:POKE #305,QMFLAG               *
026                * CALL #310,A$(FIRST)                             *
027                * LINE: line-number of first data-line            *
028                * STEP: step for line-numbering                   *
029                * NUMBER: number of $-s to transfer               *
030                * QMFLAG: if QMFLAG=34, the data is included      *
031                *         by quotation-marks                      *
032                * FIRST: index of first $                         *
033                * The line from which the subroutine is called    *
034                * must be smaller than the first data-line,       *
035                * because the following lines are moved up.       *
036                * Of course LINE+(NUMBER*STEP) should be less     *
037                * than the next following line and less than      *
038                * 65536.  The maximum-length of every $ of the    *
039                * array is 121.  Supernumerary characters are     *
040                * cutted of.      The subroutine should not be     *
041                *      called by FOR var= loops, but by           *
042                *            FOR var(x)= loops !                  *
043                ***************************************************
044                BSCBGN  EQU   :29F      start of basic-memory
045                VARBGN  EQU   :2A1      start of variable-table
046                VAREND  EQU   :2A3      end of variable-table
047                MOVE    EQU   :DE4F     memory-move-routine
048                MAXLEN  EQU   122       maximum-length+1
049                *
050                        ORG   :300
051                *
052 0300 E8FD      LINE    DBL   65000     first data-line
053 0302 0A00      STEP    DBL   10        line-step
```



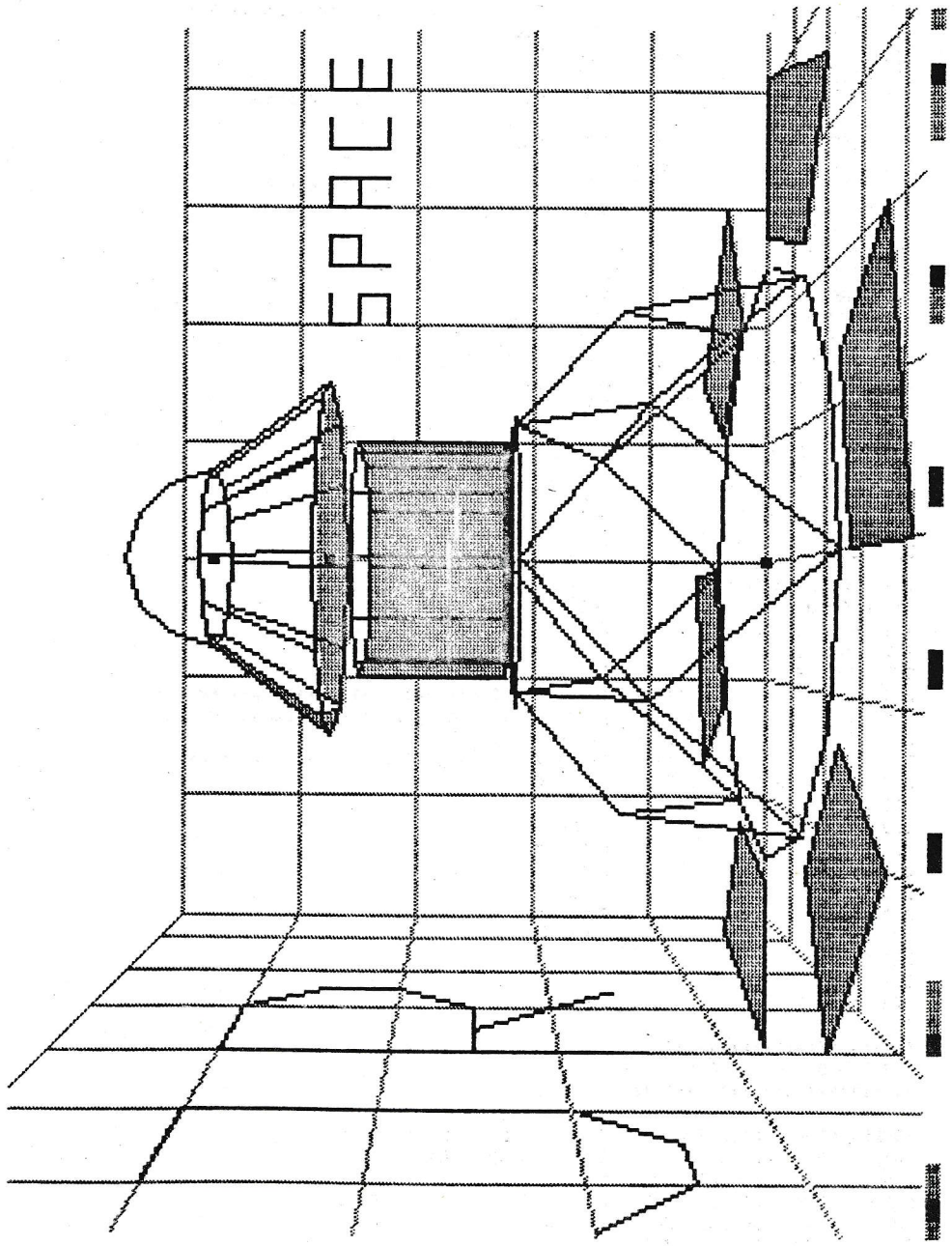
Ph.D.

k4/12

```
054 0304 00          NUMBER  DATA  0          number of $-s to transfer
055 0305 00          QMFLAG  DATA  0          quotation-mark-flag,34=yes
056 0306 0000        VARPTR  DBL   0          adress of pointer to $
057 0308 0000        HLSAVE  DBL   0
058                  *
059                          ORG   :310       startadress
060                  ************************************************
061                  * The first part of the program calculates the *
062                  *       number of bytes to be inserted:        *
063                  ************************************************
064 0310 C5                  PUSH  B
065 0311 110000              LXI   D,0
066 0314 D5                  PUSH  D          number of bytes to insert
067 0315 220603              SHLD  VARPTR     save VARPTR-value
068 0318 3A0403              LDA   NUMBER
069 031B 4F                  MOV   C,A        number of $-s
070 031C 5E          GETLNG  MOV   E,M
071 031D 23                  INX   H
072 031E 56                  MOV   D,M        pointer to $ in DE
073 031F 23                  INX   H
074 0320 7A                  MOV   A,D
075 0321 B3                  ORA   E          null-$ ?
076 0322 CA4003              JZ    NULL$
077 0325 1A                  LDAX  D          length of $ in A
078 0326 FE7A                CPI   MAXLEN     $ too long ?
079 0328 DA2D03              JLS   OK1
080 032B 3E79                MVI   A,MAXLEN-1
081 032D 1600        OK1     MVI   D,0
082 032F 5F                  MOV   E,A
083 0330 E3                  XTHL
084 0331 19                  DAD   D          add length to insert-number
085 0332 110500              LXI   D,5        4 bytes for rest of line
086 0335 19                  DAD   D
087 0336 3A0503              LDA   QMFLAG
088 0339 B7                  ORA   A          quotation-marks ?
089 033A CA3F03              JZ    NOQT1
090 033D 23                  INX   H
091 033E 23                  INX   H
092 033F E3          NOQT1   XTHL
093 0340 0D          NULL$   DCR   C          last element ?
094 0341 C21C03              JNZ   GETLNG
095                  ************************************************
096                  * The number of bytes to insert is calculated now *
097                  * and saved on stack.  The following instructions *
098                  * search the location where the data-lines are to *
099                  * be set in, and move the following lines and the *
100                  *              variable-table up:                *
101                  ************************************************
102 0344 2A0003                LHLD  LINE
103 0347 EB                  XCHG             number of first data-line
104 0348 2A9F02              LHLD  BSCBGN     adress of first program-line
105 034B 0600                MVI   B,0
106 034D 7E          NXTLIN  MOV   A,M        length of whole line
107 034E B7                  ORA   A          end of program ?
108 034F CA6D03              JZ    GRT2
109 0352 4F                  MOV   C,A
110 0353 23                  INX   H
111 0354 7E                  MOV   A,M        left byte of line-number
112 0355 BA                  CMP   D          greater/equal than left byte
113 0356 D25D03              JGE   MSBGRT     .of first data line ?
114 0359 09                  DAD   B          add length
115 035A C34D03              JMP   NXTLIN     next line
116 035D C26C03      MSBGRT  JNZ   GRT1       greater than left byte
```



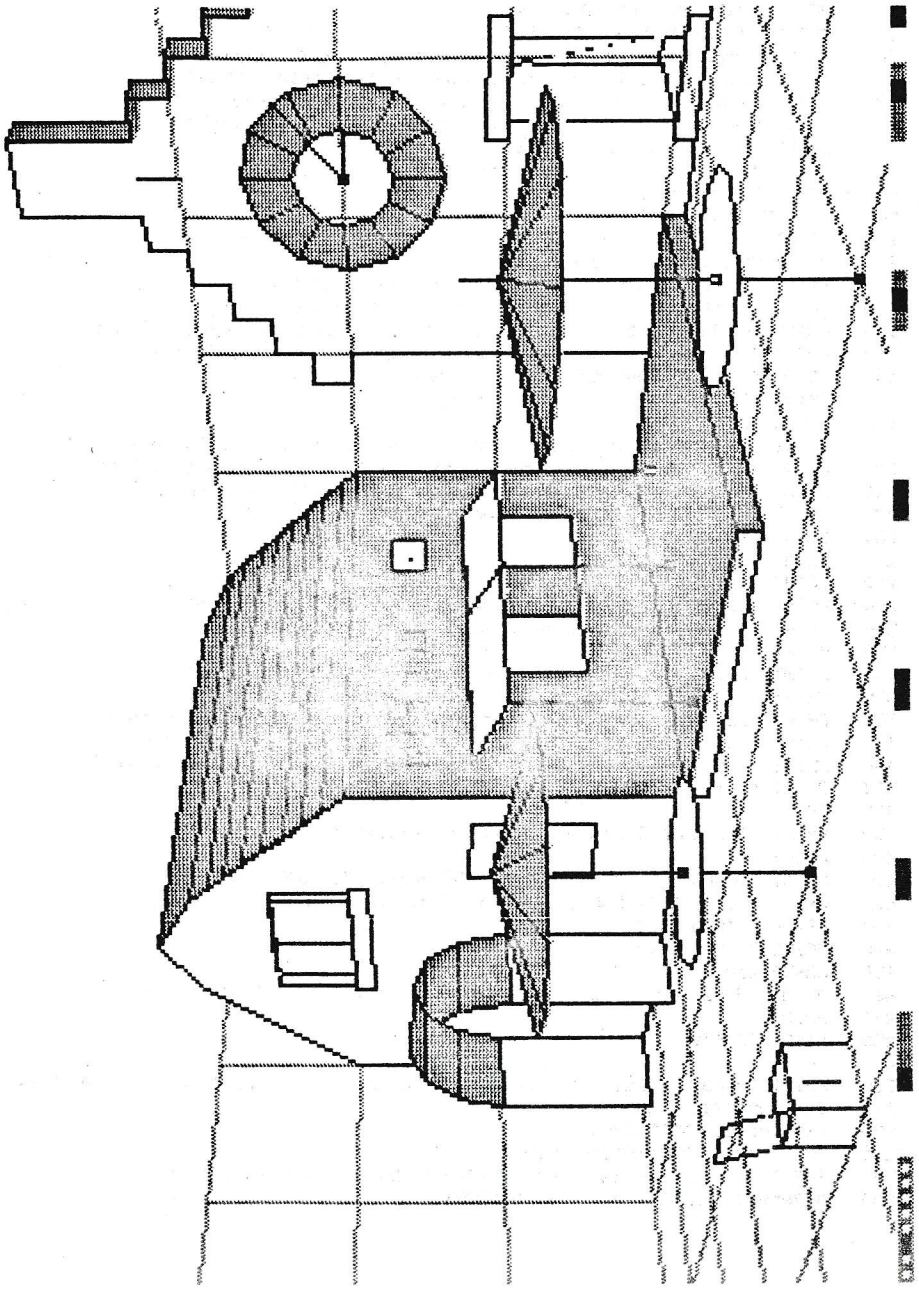
```
117 0360 23              INX   H
118 0361 0D              DCR   C
119 0362 7E              MOV   A,M       right byte of line-number
120 0363 BB              CMP   E         greater/eql. than right byte
121 0364 D26B03          JGE   GRT       .of first data line ?
122 0367 09              DAD   B         add length
123 0368 C34D03          JMP   NXTLIN    next line
124 036B 2B        GRT   DCX   H
125 036C 2B        GRT1  DCX   H
126 036D E5        GRT2  PUSH  H
127 036E D1              POP   D
128 036F C1              POP   B
129 0370 C5              PUSH  B
130 0371 09              DAD   B
131 0372 E5              PUSH  H
132 0373 C1              POP   B
133 0374 2AA302          LHLD  VAREND
134 0377 D5              PUSH  D
135 0378 CD4FDE          CALL  MOVE      move up rest of program
136                      ************************************************
137                      * Now there is enough place for the new data-lines.*
138                      * They are generated by the last program-part:     *
139                      ************************************************
140 037B 3A0403          LDA   NUMBER
141 037E 57              MOV   D,A       number of $-s
142 037F 2A0603          LHLD  VARPTR    adress of pointer to $
143 0382 4E        LINSET MOV   C,M
144 0383 23              INX   H
145 0384 46              MOV   B,M       adress of $ in BC
146 0385 23              INX   H
147 0386 78              MOV   A,B
148 0387 B1              ORA   C         null-$ ?
149 0388 CAF003          JZ    NULL$1
150 038B 0A              LDAX  B         length of $
151 038C 03              INX   B
152 038D FE7A            CPI   MAXLEN    $ too long ?
153 038F DA9403          JLS   OK2
154 0392 3E79            MVI   A,MAXLEN-1
155 0394 5F        OK2   MOV   E,A
156 0395 C604            ADI   4
157 0397 F5              PUSH  PSW
158 0398 3A0503          LDA   QMFLAG
159 039B B7              ORA   A
160 039C CAA303          JZ    NOQT2
161 039F F1              POP   PSW
162 03A0 3C              INR   A
163 03A1 3C              INR   A
164 03A2 F5              PUSH  PSW
165 03A3 F1        NOQT2 POP   PSW
166 03A4 CD0604          CALL  SET       set length of line
167 03A7 3A0103          LDA   LINE+1
168 03AA CD0604          CALL  SET
169 03AD 3A0003          LDA   LINE
170 03B0 CD0604          CALL  SET       set line-number
171 03B3 3EA2            MVI   A,:A2
172 03B5 CD0604          CALL  SET       set DATA-code
173 03B8 3A0503          LDA   QMFLAG
174 03BB B7              ORA   A
175 03BC CAC103          JZ    NOQT3
176 03BF 3E02            MVI   A,2
177 03C1 83        NOQT3 ADD   E
178 03C2 CD0604          CALL  SET       set length of data-part
179 03C5 3A0503          LDA   QMFLAG
```



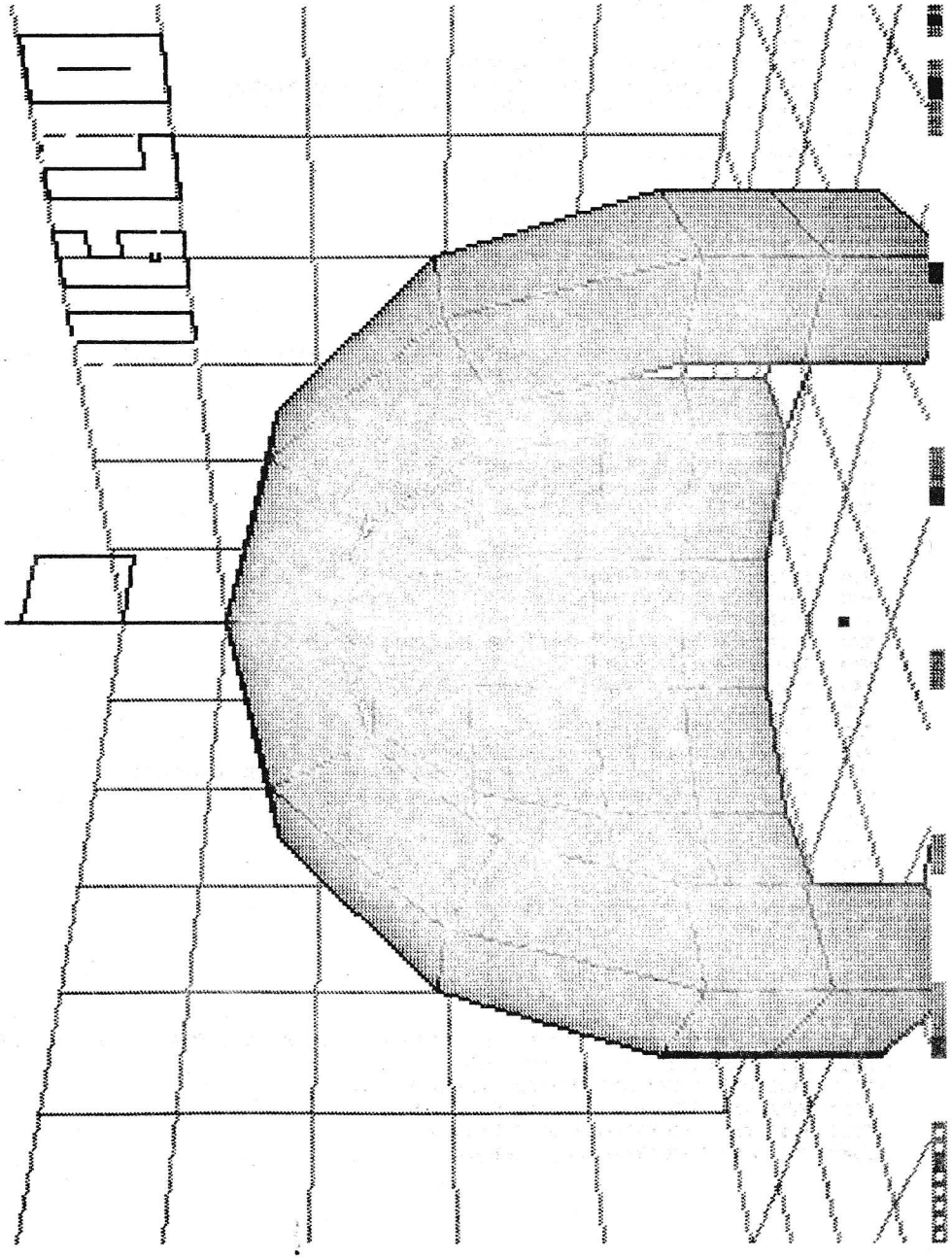
```
180 03C8 B7                ORA   A          quotation-marks ?
181 03C9 C40604            CNZ   SET
182 03CC 7B                MOV   A,E
183 03CD B7                ORA   A          no characters in DATA ?
184 03CE CADA03            JZ    ENDLIN
185 03D1 0A        DATSET  LDAX  B
186 03D2 03                INX   B
187 03D3 CD0604            CALL  SET        set data-character
188 03D6 1D                DCR   E          last character of line ?
189 03D7 C2D103            JNZ   DATSET
190 03DA 3A0503    ENDLIN  LDA   QMFLAG
191 03DD B7                ORA   A          quotations-marks ?
192 03DE C40604            CNZ   SET
193 03E1 E5                PUSH  H
194 03E2 D5                PUSH  D
195 03E3 2A0203            LHLD  STEP
196 03E6 EB                XCHG
197 03E7 2A0003            LHLD  LINE
198 03EA 19                DAD   D          add step to line-number
199 03EB 220003            SHLD  LINE
200 03EE D1                POP   D
201 03EF E1                POP   H
202 03F0 15        NULL$1  DCR   D          last line ?
203 03F1 C28203            JNZ   LINSET
204 03F4 D1                POP   D
205 03F5 C1                POP   B
206 03F6 2AA102            LHLD  VARBGN
207 03F9 09                DAD   B
208 03FA 22A102            SHLD  VARBGN
209 03FD 2AA302            LHLD  VAREND
210 0400 09                DAD   B
211 0401 22A302            SHLD  VAREND     update pointers
212 0404 C1                POP   B
213 0405 C9                RET
214                ********************************************************
215                * Subroutine to set one byte of data-line; current     *
216                * adress saved on stack, code in A.                    *
217                ********************************************************
218 0406 220803    SET     SHLD  HLSAVE
219 0409 E1                POP   H
220 040A E3                XTHL
221 040B 77                MOV   M,A
222 040C 23                INX   H
223 040D E3                XTHL
224 040E E5                PUSH  H
225 040F 2A0803            LHLD  HLSAVE
226 0412 C9                RET
227                *
228 0413                   END
```

```
****************************
* S Y M B O L   T A B L E *
****************************

BSCBGN 029F   DATSET 03D1   ENDLIN 03DA   GETLNG 031C
GRT    036B   GRT1   036C   GRT2   036D   HLSAVE 0308
LINE   0300   LINSET 0382   MAXLEN 007A   MOVE   DE4F
MSBGRT 035D   NOQT1  033F   NOQT2  03A3   NOQT3  03C1
NULL$  0340   NULL$1 03F0   NUMBER 0304   NXTLIN 034D
OK1    032D   OK2    0394   QMFLAG 0305   SET    0406
STEP   0302   VARBGN 02A1   VAREND 02A3   VARPTR 0306
```



```
1     REM +++ MARKUS SIGG  8/82 +++
10    CLEAR 1000:MODE 0:PRINT CHR$(12);
20    PRINT "This programm demonstrates with a little example"
30    PRINT "the facilities of the subroutine ARRAYDATA.":PRINT
40    PRINT "ARRAYDATA is a utility-program and can be used if"
50    PRINT "the content of a string-array shall be transferred"
60    PRINT "into data-lines.":GOSUB 1100
70    PRINT "The ML-program can be called by the following"
80    PRINT "instructions:":PRINT
90    LIST 5000-5040:PRINT
100   PRINT "LINE:   line-number of first data-line"
110   PRINT "STP:    line-step"
120   PRINT "NUMBER: number of strings to transfer,0=256"
130   PRINT "FIRST:  index of first string to transfer"
140   PRINT "QMFLAG: set QMFLAG=34 (";CHR$(34);") when you want to transfer"
150   PRINT "        strings which contain ',' and QMFLAG=0 when you"
160   PRINT "        want to transfer numbers"
170   PRINT "ARRAY$: array containing the strings"
180   GOSUB 1100
190   PRINT "ATTENTION: The subroutine may not be called within normal"
200   PRINT "           loops, because the pointers to the loop-variables"
210   PRINT "           can't be updated !! Loops with subscripted"
220   PRINT "           variables must be used."
230   GOSUB 1100
240   PRINT "Here our example:"
250   PRINT
260   PRINT "We want to type in a list of 5 persons, using the"
270   PRINT "following characteristics:"
280   PRINT
290   PRINT "last name,first name,land,town,street,house-number"
310   PRINT
320   PRINT "Last name and first name shall be stored together in one"
330   PRINT "data-element in the first DATA-line."
340   PRINT "The second line shall contain land,town,street and house-"
350   PRINT "number, stored as seperated elements."
370   PRINT
380   PRINT "The DATA-lines will look as following:"
390   PRINT
400   PRINT "10000 DATA ";CHR$(34);"LAST NAME,FIRST NAME";CHR$(34)
410   PRINT "10010 DATA LAND,TOWN,STREET,HOUSENUMBER"
430   PRINT "11000 ..."
440   PRINT "11010 ..."
450   GOSUB 1100
460   PRINT "Here are the instructions to generate this list:"
470   PRINT
480   LIST 510-670
490   PRINT
500   PRINT "PRESS 'SPACE' TO RUN THESE INSTRUCTIONS !":CALLM #D6DA:PRINT CHR$(12);
510   DIM ARRAY$(1),I%(0)
520   LINE%=10000:NUMBER%=1
530   FOR I%(0)=1 TO 5:PRINT "Person no.";I%(0);":"
540   INPUT "First name,last name ";FIRSTNAME$,LASTNAME$:PRINT
550   ARRAY$(0)=LASTNAME$+","+FIRSTNAME$
```



```
560   INPUT "Land ";LAND$:PRINT
570   INPUT "Town ";TOWN$:PRINT
580   INPUT "Street ";STREET$:PRINT
590   INPUT "Housenumber ";HOUSENUMBER$:PRINT
600   ARRAY$(1)=LAND$+","+TOWN$+","+STREET$+","+HOUSENUMBER$
650   FIRST%=0:QMFLAG%=34:GOSUB 5000:LINE%=LINE%+10
660   FIRST%=1:QMFLAG%=0:GOSUB 5000
670   PRINT :LINE%=LINE%+990:NEXT
680   PRINT CHR$(12);"Here is the result:":LIST 10000-
690   GOSUB 1100
700   PRINT "We can execute a CLEAR:":TRON
710   CLEAR 1000
720   TROFF
730   PRINT "and our data is not lost:"
740   LIST 10000-
750   GOSUB 1100
760   PRINT "Now we want to lay out a second list which presents the"
770   PRINT "data sorted in names,lands,towns,..."
780   PRINT
790   PRINT "This list shall look as followiing:"
800   PRINT
810   PRINT "20000 DATA NAME 1,NAME 2,...,NAME 5"
820   PRINT "20010 DATA LAND 1,LAND 2,...,LAND 5"
830   PRINT "20020  .     .     .          ."
870   GOSUB 1100
880   PRINT "Following program solves our problem:"
890   PRINT
900   LIST 930-1010
910   PRINT
920   PRINT "PRESS 'SPACE' TO START THIS PROGRAM-PART !"
930   DIM ARRAY$(4)
940   LINE%=20000:STP%=10:NUMBER%=5:FIRST%=0:QMFLAG%=0
950   RESTORE
960   FOR I%=1 TO 5
970   FOR J%=0 TO 4
980   READ ELEMENT$:IF J%=0 THEN ELEMENT$=CHR$(34)+ELEMENT$+CHR$(34)
990   ARRAY$(J%)=ARRAY$(J%)+ELEMENT$:IF I%<>5 THEN ARRAY$(J%)=ARRAY$(J%)+","
1000  NEXT:NEXT
1010  GOSUB 5000
1020  GOSUB 1100
1030  PRINT "Here the new list:"
1040  PRINT
1050  LIST 20000-
1060  GOSUB 1100
1070  PRINT "And here the whole list:"
1080  PRINT
1090  LIST 10000-:END
1100  PRINT :PRINT "PRESS 'SPACE' TO CONTINUE !":CALLM #D6DA:PRINT CHR$(12);:RETURN
5000  REM Subroutine to call ARRAYDATA
5010  POKE #300,LINE% MOD 256:POKE #301,LINE%/256
5020  POKE #302,STP% MOD 256:POKE #303,STP%/256
5030  POKE #304,NUMBER%:POKE #305,QMFLAG%
5040  CALLM #310,ARRAY$(FIRST%):RETURN
```



```
001                 ****************************************************
002                 *                                                  *
003                 * EEEEEE  DDDDD      I   TTTTTTT     >             *
004                 * E       D    D     I      T         >            *
005                 * E       D    D     I      T          >           *
006                 * EEEE    D    D     I      T     >>>>>>>          *
007                 * E       D    D     I      T          >           *
008                 * E       D    D     I      T         >            *
009                 * EEEEEE  DDDDD      I      T        >             *
010                 *                                                  *
011                 * DDDDD    AAAA   TTTTTTT  AAAA                    *
012                 * D    D  A    A     T    A    A                   *
013                 * D    D  A    A     T    A    A                   *
014                 * D    D  AAAAAA     T    AAAAAA                   *
015                 * D    D  A    A     T    A    A                   *
016                 * D    D  A    A     T    A    A      Markus Sigg  *
017                 * DDDDD   A    A     T    A    A           8/82    *
018                 *                                                  *
019                 ****************************************************
020                 * This subroutine can be used to transfer data     *
021                 * from the EDIT-buffer to data lines. The call     *
022                 * of the routine looks as following:               *
023                 * POKE #300,LINE MOD 256:POKE #301,LINE/256        *
024                 * POKE #302,STEP MOD 256:POKE #303,STEP/256        *
025                 * POKE #304,QMFLAG:CALLM #310                      *
026                 * The whole EDIT-buffer is transferred to data     *
027                 * lines, starting with the line-number given       *
028                 * by LINE and the line-step specified by STEP.     *
029                 * QMFLAG=0:quotation-marks are not transferred     *
030                 * QMFLAG<>0: the specified character is            *
031                 *            transferred if a quotation-mark       *
032                 *            occures. (example:39 for ')           *
033                 * The end of a line to be read out is marked       *
034                 * by the carriage-return character.                *
035                 * No line may contain more than 121 characters,    *
036                 * otherwise errors will occure.                    *
037                 * The line from which the subroutine is called     *
038                 * must be smaller than the first data line.        *
039                 * The subroutine should not be called by           *
040                 * FOR var= loops, but by FOR var(x)= loops !       *
041                 ****************************************************
042                 BUFBGN  EQU   :A2       start of edit-buffer
043                 BSCBGN  EQU   :29F      start of basic-memory
044                 VARBGN  EQU   :2A1      start of variable-table
045                 VAREND  EQU   :2A3      end of variable-table
046                 MOVE    EQU   :DE4F     memory-move-routine
047                 *
048                         ORG   :300
049                 *
050 0300 EBFD       LINE    DBL   65000     first data-line
051 0302 0A00       STEP    DBL   10        line-step
052 0304 00         QMFLAG  DATA  0         quotation-mark-flag
053 0305 0000       HLSAVE  DBL   0
```

Tkv/

```
054 0307 0000       ADRSVE  DBL   0
055                 *
056                         ORG   :310      startadress
057                 ************************************************
058                 * The first part of the program calculates the *
059                 *       number of bytes to be inserted         *
060                 ************************************************
061 0310 C5                 PUSH  B
062 0311 2AA200             LHLD  BUFBGN
063 0314 E5                 PUSH  H
064 0315 C1                 POP   B         start of buffer in BC
065 0316 210000             LXI   H,0       number of bytes to insert
066 0319 110600             LXI   D,6
067 031C 0A         GETCD1  LDAX  B
068 031D 03                 INX   B
069 031E B7                 ORA   A         end of buffer ?
070 031F C22F03             JNZ   NOEND
071 0322 0B                 DCX   B
072 0323 0B                 DCX   B
073 0324 0A                 LDAX  B
074 0325 FE0D               CPI   :D
075 0327 CA4503             JZ    BFEND1
076 032A 23                 INX   H
077 032B 19                 DAD   D
078 032C C34503             JMP   BFEND1
079 032F FE22       NOEND   CPI   34        quotation-mark ?
080 0331 C23B03             JNZ   NOQT1
081 0334 3A0403             LDA   QMFLAG
082 0337 B7                 ORA   A
083 0338 CA1C03             JZ    GETCD1
084 033B 23         NOQT1   INX   H
085 033C FE0D               CPI   :D        end of line ?
086 033E C21C03             JNZ   GETCD1
087 0341 19                 DAD   D
088 0342 C31C03             JMP   GETCD1
089 0345 E5         BFEND1  PUSH  H
090                 ************************************************
091                 * The number of bytes to insert is calculated now *
092                 * and saved on stack.  The following instructions *
093                 * search the location where the data-lines are to *
094                 * be set in, and move the following lines and the *
095                 *              variable-table up:                 *
096                 ************************************************
097 0346 2A0003                LHLD  LINE
098 0349 EB                 XCHG            number of first data-line
099 034A 2A9F02             LHLD  BSCBGN    adress of first program-line
100 034D 0600               MVI   B,0
101 034F 7E         NXTLIN  MOV   A,M       length of whole line
102 0350 B7                 ORA   A         end of program ?
103 0351 CA6F03             JZ    GRT2
104 0354 4F                 MOV   C,A
105 0355 23                 INX   H
106 0356 7E                 MOV   A,M       left byte of line-number
107 0357 BA                 CMP   D         greater/equal than left byte
108 0358 D25F03             JGE   MSBGRT    .of first data line ?
109 035B 09                 DAD   B         add length
110 035C C34F03             JMP   NXTLIN    next line
111 035F C26E03     MSBGRT  JNZ   GRT1      greater than left byte
112 0362 23                 INX   H
113 0363 0D                 DCR   C
114 0364 7E                 MOV   A,M       right byte of line-number
115 0365 BB                 CMP   E         greater/eql. than right byte
116 0366 D26D03             JGE   GRT       .of first data line ?
```



```
117 0369 09                 DAD   B         add length
118 036A C34F03             JMP   NXTLIN    next line
119 036D 2B         GRT     DCX   H
120 036E 2B         GRT1    DCX   H
121 036F E5         GRT2    PUSH  H
122 0370 D1                 POP   D
123 0371 C1                 POP   B
124 0372 C5                 PUSH  B
125 0373 09                 DAD   B
126 0374 E5                 PUSH  H
127 0375 C1                 POP   B
128 0376 2AA302             LHLD  VAREND
129 0379 D5                 PUSH  D
130 037A CD4FDE             CALL  MOVE      move up rest of program
131                 ************************************************
132                 * Now there is enough place for the new data-lines.*
133                 * They are generated by the last program-part :    *
134                 ************************************************
135 037D 2AA200             LHLD  BUFBGN
136 0380 7E         SETLIN  MOV   A,M
137 0381 B7                 ORA   A         end of buffer ?
138 0382 CAF203             JZ    BFEND2
139 0385 E3                 XTHL
140 0386 220703             SHLD  ADRSVE    adress of start of line
141 0389 E3                 XTHL
142 038A 3E2A               MVI   A,'*'
143 038C CD0404             CALL  SET       skip length-byte of line
144 038F 3A0103             LDA   LINE+1
145 0392 CD0404             CALL  SET
146 0395 3A0003             LDA   LINE
147 0398 CD0404             CALL  SET       set line-number
148 039B 3EA2               MVI   A,:A2
149 039D CD0404             CALL  SET       set DATA-code
150 03A0 3E2A               MVI   A,'*'
151 03A2 CD0404             CALL  SET       skip length of data-part
152 03A5 3E22               MVI   A,'"'
153 03A7 CD0404             CALL  SET       set start-quotation-mark
154 03AA 0600               MVI   B,0       length of data-part
155 03AC 7E         GET     MOV   A,M
156 03AD 23                 INX   H
157 03AE FE0D               CPI   :D        end of line ?
158 03B0 CACA03             JZ    ENDLIN
159 03B3 B7                 ORA   A
160 03B4 CACA03             JZ    ENDLIN
161 03B7 FE22               CPI   34        quotation-mark ?
162 03B9 C2C303             JNZ   NOQT2
163 03BC 3A0403             LDA   QMFLAG
164 03BF B7                 ORA   A
165 03C0 CAAC03             JZ    GET
166 03C3 CD0404     NOQT2   CALL  SET       set character
167 03C6 04                 INR   B         increment data-length
168 03C7 C3AC03             JMP   GET
169 03CA F5         ENDLIN  PUSH  PSW
170 03CB D1                 POP   D
171 03CC 3E22               MVI   A,'"'
172 03CE CD0404             CALL  SET       set end-quotation-mark
173 03D1 D5                 PUSH  D
174 03D2 E5                 PUSH  H
175 03D3 2A0703             LHLD  ADRSVE
176 03D6 3E06               MVI   A,6
177 03D8 80                 ADD   B
178 03D9 77                 MOV   M,A
179 03DA 23                 INX   H
```


TKY/35

```
180 03DB 23              INX   H
181 03DC 23              INX   H
182 03DD 23              INX   H
183 03DE 04              INR   B
184 03DF 04              INR   B
185 03E0 70              MOV   M,B
186 03E1 2A0203          LHLD  STEP
187 03E4 EB              XCHG
188 03E5 2A0003          LHLD  LINE
189 03E8 19              DAD   D
190 03E9 220003          SHLD  LINE
191 03EC E1              POP   H
192 03ED F1              POP   PSW
193 03EE B7              ORA   A
194 03EF C28003          JNZ   SETLIN
195 03F2 D1       BFEND2 POP   D
196 03F3 C1              POP   B
197 03F4 2AA102          LHLD  VARBGN
198 03F7 09              DAD   B
199 03F8 22A102          SHLD  VARBGN
200 03FB 2AA302          LHLD  VAREND
201 03FE 09              DAD   B
202 03FF 22A302          SHLD  VAREND
203 0402 C1              POP   B
204 0403 C9              RET
205                ***************************************************
206                * Subroutine to set one byte of data-line; current *
207                * adress saved on stack, code in A.                *
208                ***************************************************
209 0404 220503    SET   SHLD  HLSAVE
210 0407 E1              POP   H
211 0408 E3              XTHL
212 0409 77              MOV   M,A
213 040A 23              INX   H
214 040B E3              XTHL
215 040C E5              PUSH  H
216 040D 2A0503          LHLD  HLSAVE
217 0410 C9              RET
218                *
219 0411                 END

*****************************
* S Y M B O L   T A B L E *
*****************************

ADRSVE 0307   BFEND1 0345   BFEND2 03F2   BSCBGN 029F
BUFBGN 00A2   ENDLIN 03CA   GET    03AC   GETCD1 031C
GRT    036D   GRT1   036E   GRT2   036F   HLSAVE 0305
LINE   0300   MOVE   DE4F   MSBGRT 035F   NOEND  032F
NOQT1  033B   NOQT2  03C3   NXTLIN 034F   QMFLAG 0304
SET    0404   SETLIN 0380   STEP   0302   VARBGN 02A1
VAREND 02A3
```



```
1     REM +++ MARKUS SIGG  8/82 +++
10    MODE 0:PRINT CHR$(12);
20    PRINT "This program demonstrates the use of the subroutine"
30    PRINT "EDITDATA. EDITDATA can be used, when you want to transfer"
40    PRINT "the content of the EDIT-buffer into DATA-lines."
50    PRINT
60    PRINT "EDITDATA is called by the following instructions:"
70    LIST 5000-5030
80    PRINT "LINE:    line-number of first DATA-line"
90    PRINT "STP:     line-step"
100   PRINT "QMFLAG: 0=do not transfer quotation-marks"
110   PRINT "       <>0=use character(QMFLAG) instead of quot.mark"
120   PRINT
130   CLEAR 10000
140   PRINT "Now type EDIT,BREAK-BREAK,CONT !"
150   STOP
200   PRINT "Now the EDIT-buffer will be read out:"
210   TRON
220   LINE%=10000:STP%=5:QMFLAG%=39:GOSUB 5000
230   TROFF
240   PRINT
250   PRINT "PRESS 'SPACE' TO CONTINUE !"
260   CALLM #D6DA:PRINT CHR$(12);
270   LIST 10000-:END
5000  REM Subroutine to call EDITDATA
5010  POKE #300,LINE% MOD 256:POKE #301,LINE%/256
5020  POKE #302,STP% MOD 256:POKE #303,STP%/256
5030  POKE #304,QMFLAG%:CALLM #310:RETURN
```

TOOLKIT 1 appendix A

60–28

audio :.

the programs on APPENDIX A-TOOLKIT 1

```
*     Obj ARRAY-DATA (H300-H41F)
*     ARRAY-DATA DEMO 1
*     ARRAY-DATA DEMO 2
*     ARRAY-DATA DEMO 3
*     Obj EDIT-DATA  (H300-H41F)
*     EDIT-DATA DEMO 1
*     EDIT-DATA DEMO 2
*     SOURCE ARRAY-DATA
*     SOURCE EDIT-DATA
```

dcr :

```
1-ASS - ARRAYDATA
2-OBJ - ARRAYDATA 298-41F
3-BAS - ARRAY-DATA DEMO 1
4-BAS - ARRAY-DATA DEMO 2
5-BAS - ARRAY-DATA DEMO 3
6-ASS - EDITDATA
7-OBJ - EDITDATA 298-410
8-BAS - EDIT-DATA DEMO 1
9-BAS - EDIT-DATA DEMO 2
```