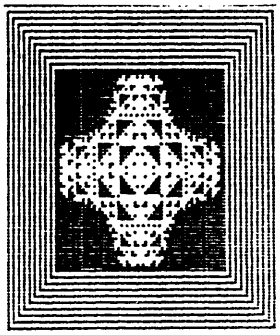
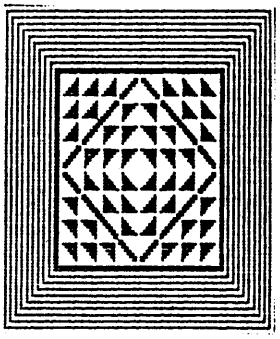


DE TEKENSTOMEN



```
650   CURSOR 0,1:PRINT SPC(59):CURSOR 0,0:PRINT SPC(59);
660   CURSOR 10,0:POKE #75,95
670   PRINT "WILT U NOG EEN TEKENING ZIEN (J/N) ?";
680   G=GETC:WAIT TIME 3:IF G=0.0 THEN 680
690   IF G=ASC("J") THEN POKE #75,32:GOTO 330
700   IF G=ASC("N") THEN PRINT CHR$(12):PRINT :PRINT TAB(22);:END
710   GOTO 680
720   REM *** BOVEN LINKER HELFT OPTIE 2
730   CURSOR 21,1:PRINT "30";TAB(41);"1":Z=T:N=Z
740   FOR R=1.0 TO N:FOR C=1.0 TO Z-1.0
750   IF (R-1.0)*(C-1.0)=0.0 THEN 770
760   P(R,C)=P(R,C-1.0)+P(R-1.0,C):GOTO 780
770   P(R,C)=1.0
780   NEXT C:Z=Z-1.0:CURSOR 20,0:R%=INT(R):PRINT R%;:NEXT R
790   REM *** ONDER RECHTER HELFT OPTIE 2
800   Z=N:N=2.0:FOR R=Z TO 1.0 STEP -1.0:FOR C=Z TO N STEP -1.0
810   IF (R-Z)*(C-Z)=0.0 THEN 830
820   P(R,C)=P(R,C+1.0)+P(R+1.0,C):GOTO 840
830   P(R,C)=1.0
840   NEXT C:N=N+1.0:CURSOR 40,0:R%=INT(R)
850   PRINT R%;" ";:NEXT R:GOTO 590
860   REM *** BOVEN LINKER HELFT OPTIE 3
870   CURSOR 11,1:PRINT "15         15";TAB(41);"16         16"
880   M=Q:Y=T:Z=INT(Y/2.0):B5=Z*2.0:Z1=Z
890   Z2=Z1:Z3=Z2:X4=Z3:X5=X4
900   FOR I=1.0 TO Z1:FOR J=1.0 TO Z
910   IF (J-1.0)*(I-1.0)=0.0 THEN 930
920   P(I,J)=P(I,J-1.0)+P(I-1.0,J):GOTO 940
930   P(I,J)=1.0
940   NEXT J:Z=Z-1.0:CURSOR 10,0:I%=INT(I)
950   PRINT I%;:NEXT I:N=Z1
960   REM *** BOVEN RECHTER HELFT OPTIE 3
970   FOR I=1.0 TO Z1:FOR J=Y TO X5+1.0 STEP -1.0
980   IF I=1.0 OR J=Y THEN 1000
990   P(I,J)=P(I,J+1.0)+P(I-1.0,J):GOTO 1010
1000  P(I,J)=1.0
1010  NEXT J:X5=X5+1.0:CURSOR 20,0:I%=INT(I)
1020  PRINT I%;:NEXT I:N=Z2
1030  REM *** ONDER LINKER HELFT OPTIE 3
1040  FOR I=Y TO X4+1.0 STEP -1.0:FOR J=1.0 TO Z2
1050  IF J=1.0 OR I=Y THEN 1070
1060  P(I,J)=P(I,J-1.0)+P(I+1.0,J):GOTO 1080
1070  P(I,J)=1.0
1080  NEXT J:Z2=Z2-1.0:CURSOR 40,0:I%=INT(I)
1090  PRINT I%;:NEXT I:N=Z3
1100  REM *** ONDER RECHTER HELFT OPTIE 3
1110  FOR I=Y TO N+1.0 STEP -1.0:FOR J=Y TO Z3+1.0 STEP -1.0
1120  IF J=Y OR I=J THEN 1140
1130  P(I,J)=P(I+1.0,J)+P(I,J+1.0):GOTO 1150
1140  P(I,J)=1.0
1150  NEXT J:Z3=Z3+1.0:CURSOR 50,0:I%=INT(I)
1160  PRINT I%;:NEXT I:GOTO 590
```

Interface Age

8080 2707 EPROM programming software
assembled listing December 1978 p 104

8080 A tabcounter for your edit buffer
assembled listing August 1978 p 151

8080 An 8080 binary tape monitor
assembled listing February 1978 p 153

8080 An Intel hex-format paper tape monitor
assembled listing March 1978 p 162

8080 Blockade—VDM-1 video game
assembled listing with object code November 1977 p 167

8080 Cassette Operating System
assembled subroutine listings April 1977 p 129

8080 Convert Motorola 6800 hex format to Intel format
assembled listing with object code May 1977 p 109

8080 Conway's Game of Life
assembled listing for Processor Technology's video display board
May 1977 p 138

8080 Copy/Sort/Search data manipulation
assembled listing September 1978 p 136

8080 Cromemco Dazzler graphics interface driver
assembled listing January 1978 p 153

8080 Diablo printer drive routine
assembled listing July 1977 p 25

8080 Dr. Wang's tiny BASIC
assembled listing December 1976 p 89

8080 Generalized string sorting routine
BASIC and 6 assembled listings November 1978 p 131

8080 I/O driver for PERSCI 1070 intelligent disk controller
assembled listing September 1977 p 153

8080 IAPS™International ASCII Publication Standard conversion
asembled listing May 1978 p 79

8080 IAPS™ International ASCII Publication Standard conversion
assembled load-create-verify-dump listings June 1978 p 148

8080 Intel hex format paper-tape punch program
assembled listing with object code July 1977 p 155

8080 Intelligent terminal program
assembled listing with object code September 1977 p 75

8080 Interval timer design
5 assembled routines January 1978 p 127

8080 Intrupped driven floppy disk controller for the S-100 bus
assembled listing May 1978 p 152

8080 Livermore BASIC Interpreter Part I
BASIC and assembled plot functions December 1976 p 115

8080 Livermore BASIC Interpreter Part II
assembled listing January 1977 p 97

8080 Livermore BASIC Interpreter Part III
assembled floating point math package listing February 1977 p 104

8080 Livermore BASIC Interpreter Part IV
assembled octal debugging listing March 1977 p 121

8080 Look—byte look-up program
assembled listing May 1978 p 167

8080 Memory catalog program
assembled listing May 1978 p 170

8080 Memory object code search routine
assembled routine February 1977 p 121

8080 Monitor initialization and printer control
2 assembled Daisywheel listings October 1978 p 111

8080 Morse Code generator
assembled listing October 1978 p 89

8080 Octal Monitor Program
assembled listing February 1977 p 13

8080 Piranha—game
assembled listing with object code December 1977 p 166

8080 Polymorphic Ideaboard software
2 assembled routines May 1977 p 98

8080 Punch and read Intel formatted tape
assembled listing December 1977 p 152

8080 Random Number Generator Program
assembled listing with a BASIC Chi-square subroutine February 1977 p 100

8080 Robot's random programming approach
assembled listing April 1978 p 158

8080 Text editor for XEK and PTCo Assemblers
assembled listing October 1978 p 140

8080 Video Chase—a game
assembled listing with object code October 1977 p 167

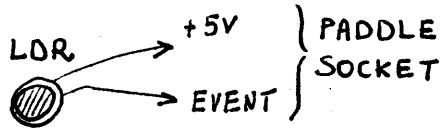
8080A Resident Operating System
1K SOL assembled listing January 1977 p 90

8080/8085/8086 boeken
D2 8080A/8085 ASSEMBLY LANGUAGE PROGRAMMING (Leventhal/Osborne)
D3 8080 PROGRAMMING FOR A LOGIC DESIGN (Osborne/Osborne)
D4 8080 SOFTWARE GOURMET GUIDE & COOKBOOK (Findley/Scelbi)
D5 EDITOR/ASSEMBLER SYSTEM FOR 8080/8085 BASED COMPUTERS (Weller)
D6 DBUG: AN 8080 INTERPRETIVE DEBUGGER (Titus/Sams)
D7 KF2DOS: A FLOPPY DISK OPERATING SYSTEM FOR THE 8080 (Wellus)
D8 8080 GALAXY GAME (Edwards/Scelbi)
D9 8080 STANDARD EDITOR (Edwards/Scelbi)
D10 8080 STANDARD ASSEMBLER (Edwards/Scelbi)
D11 8080 STANDARD MONITOR (Edwards/Scelbi)
D12 THE 8086 PRIMER (Morse/Hayden)
D13 TEA: AN 8080/8085 CO-RESIDENT EDITOR/ASSEMBLER (Titus/Sams)
D16 THE 8086 BOOK (Rector & Alexy/ Osborne)

```
2       REM TITEL MENU MET LICHTPEN V0.0
4       REM DATUM 26-12-81
6       REM BRON (AUTEUR) M.J.BERKX
8       REM OPSLAG BAND NR.16; CODE LP1;NR A1
10      MODE 0:PRINT CHR$(12):COLORT 0 8 15 0
15      REM *******OPBOUW VAN MENU**********
20      CURSOR 6,20:PRINT "Eerste keuze"
25      POKE #BE4D+1,#B:POKE #BE4D-3,#B
30      CURSOR 6,15:PRINT "Tweede keuze"
35      POKE #BBAF+1,#B:POKE #BBAF-3,#B
40      CURSOR 6,10:PRINT "Derde keuze"
45      POKE #B911+1,#B:POKE #B911-3,#B
60      K1=0.0:K2=0.0:K3=0.0
65      CURSOR 3,3:PRINT "Plaats de lichtpen TUSSEN DE STREEPJES voor Uw keuze"
70      WAIT TIME 200
80      REM ***** DOORLOPEN VAN MENU MET WIT BLOKJE*********
100     FOR N=1.0 TO 10.0
110     POKE #BE4D-3,#FF:POKE #BE4D-1,#FF:FOR T=0.0 TO 100.0:NEXT
112     IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 500
115     POKE #BE4D-1,0:POKE #BE4D-3,0
120     POKE #BBAF-3,#FF:POKE #BBAF-1,#FF:FOR T=0.0 TO 100.0:NEXT
122     IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 600
125     POKE #BBAF-1,0:POKE #BBAF-3,0
130     POKE #B911-3,#FF:POKE #B911-1,#FF:FOR T=0.0 TO 100.0:NEXT
132     IF PEEK(#FD00) IAND #20=#20 THEN GOSUB 700
135     POKE #B911-1,0:POKE #B911-3,0
160     NEXT
165     REM *******RESULTAAT VAN KEUZE OF GEEN KEUZE*******
170     PRINT CHR$(12):CURSOR 10,12:PRINT "GEEN KEUZE TE MAKEN":END
500     PRINT CHR$(12):CURSOR 10,12:PRINT "U MAAKTE KEUZE EEN":END
510     RETURN
600     PRINT CHR$(12.0):CURSOR 10,12:PRINT "U MAAKTE KEUZE TWEE":END
610     RETURN
700     PRINT CHR$(12):CURSOR 10,12:PRINT "U MAAKTE KEUZE DRIE":END
710     RETURN
1000    POKE #BE4D+1,#A:POKE #BE4D-3,#A
*
*
*
*
*
```

CENTRONICS

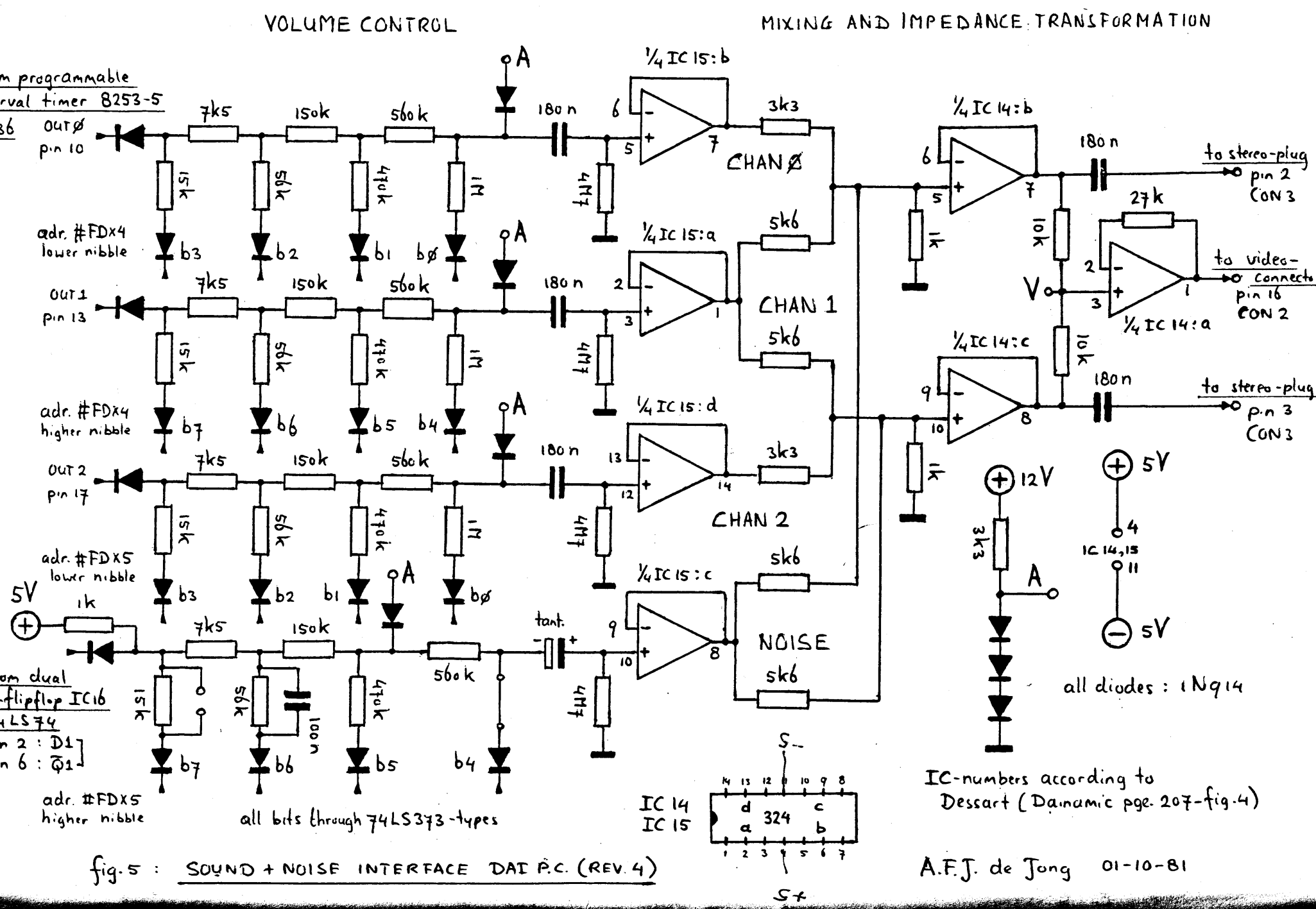
LICHTPEN




```
*LOAD:LIST
10      MODE 0
20      PRINT CHR$(12):CURSOR 10,12
30      PRINT "PAK ME DAN, ALS JE KAN!"
40      WAIT TIME 200
50      MODE 4
60      COLORG 0 5 10 15
90      FOR N=1.0 TO 5.0
100     X1=10.0+RND(XMAX-20.0):Y1=10.0+RND(YMAX-20.0)
110     FILL X1,Y1 X1+10,Y1+10 15
120     WAIT MEM #FD00,#10
130     X2=10.0+RND(XMAX-20.0):Y2=10.0+RND(YMAX-20.0)
140     FILL X2,Y2 X2+10,Y2+10 15
150     FILL X1,Y1 X1+10,Y1+10 0
160     WAIT MEM #FD00,#10
170     FILL X2,Y2 X2+10,Y2+10 0
180     NEXT
190     K=K+1.0
195     ON K GOTO 200,210,220,230,240
200     PRINT "JE KRIJGT ME TOCH NIET TE PAKKEN!":WAIT TIME 200:GOTO 50
210     PRINT "IK BEN LEKKER SNELLER!":WAIT TIME 200:GOTO 50
220     PRINT "VOLHOUDEN! JE HAD ME BIJNA!":WAIT TIME 200:GOTO 50
230     PRINT "EEN BEETJE VLUGGER EN JE HEBT ME!":WAIT TIME 200:GOTO 50
240     PRINT "HOU MAAR OP IK BEN STEEDS SNELLER DAN JIJ!":END
*
*
*
*
*
```

LUXAFLEX

```
*LOAD:LIST
1       REM TITEL "LUXAFLEX"
2       REM BRON/AURTEUR: M.J. BERKX (27-12-81)
3       REM OPSLAG: BAND NR.14; CODE P01
5       COLORT 8 0 0 0
6       READ C
7       IF C<0.0 THEN END
10      FOR A=1.0 TO 23.0:FOR B=1.0 TO 60.0:PRINT CHR$(C);:NEXT B:PRINT :NEXT A
15      FOR P=1.0 TO 3.0
20      K=#F
25      FOR N=0.0 TO 23.0
30      POKE #BFEF-N*#86,#70+K
40      WAIT TIME 2
50      NEXT N
60      K=#17-K
80      WAIT TIME 20
90      FOR N=23.0 TO 0.0 STEP -1.0
100     POKE #BFEF-N*#86,#70+K
110     WAIT TIME 2
120     NEXT N
125     WAIT TIME 20
130     NEXT P
140     GOTO 6
150     DATA 1,2,6,15,29,22,-1
```


VOLUME CONTROL
MIXING AND IMPEDANCE TRANSFORMATION
from programmable interval timer 8253-5
IC36
OUT Ø pin 10
OUT 1 pin 13
OUT 2 pin 17
adr. #FDx4 lower nibble
adr. #FDx4 higher nibble
adr. #FDx5 lower nibble
adr. #FDx5 higher nibble
from dual D-flipflop IC16 74LS74
pin 2 : D1
pin 6 : Q̄1
7k5
150k
560k
180n
15k
56k
470k
1M
47n
1/4 IC 15:b
1/4 IC 15:a
1/4 IC 15:d
1/4 IC 15:c
3k3
5k6
CHAN Ø
CHAN 1
CHAN 2
NOISE
tant.
1k
100n
all bits through 74LS373-types
1/4 IC 14:b
1/4 IC 14:a
1/4 IC 14:c
27k
10k
to stereo-plug pin 2 CON 3
to video-connector pin 16 CON 2
to stereo-plug pin 3 CON 3
12V
5V
IC 14,15
all diodes : 1N914
IC-numbers according to Dessart (Dynamic pge. 207-fig.4)
IC 14
IC 15
fig. 5 : SOUND + NOISE INTERFACE DAI P.C. (REV. 4)
A.F.J. de Jong 01-10-81

DAI FIRMWARE

RESIDENT SOFTWARE IN THE DAI pC

The resident software of your DAI (the 'firmware') is stored in 24K ROM. The memory addresses where this software can be found is CØØØ - EFFF.
As this is only a address range of 12K, bank switching is used to enable the use of 24K ROM in this address area. Bank switching is done on the addresses EØØØ - EFFF. That gives the following situation in the memory map:

CØØØ - CFFF	math. functions; machine control routines
DØØØ - DFFF	machine control routines; Basic routines.

This part of the memory is not-switched.

ØEØØØ - ØEFFF	Bank Ø : Basic handling routines.
1EØØØ - 1EFFF	Bank 1 : Math. routines; sound functions.
2EØØØ - 2EFFF	Bank 2 : Screen driving package.
3EØØØ - 3EFFF	Bank 3 : Basic encoding; utility package.

The switching of these banks is done under software control via the RST commands to the 8080 processor (see DAInamic newsletter 81/p.243).

When you write programs in machine language, a lot of routines available in this firmware can be usefully employed. An example:

Output of a character to screen/RS232:

MVI A, xx	xx is ASCII-code for the character
CALL :DD6Ø	On DD6Ø, the output routine to screen/RS232 can be found.

Do you have questions on the firmware? Write them to me. In each Newsletter from now on space is reserved for answering your questions (if the answer is known). When we don't have an answer on your question, we will communicate it to the other clubmembers in the hope to get an answer.
So don't hesitate. Don't think your question is too simple because you are just a beginner in the DAI-world. Remember: A experienced DAI-user is a beginner who started a little time earlier!!
Write your questions to:

B.J.Boerrigter
Fabritiusstraat 15
6174 RG Sweikhuizen - Nederland.

WHERE TO STORE MACHINE LANGUAGE PROGRAMS

It has become a practice to write M.L.programs with startaddress Ø3ØØ, the start of the Heap is then moved to an address after the M.L.program. When accidently a Reset happens, you may find your program destroyed. This is done by the reset routine: It defaults the HEAP to its original position (start Ø2EC) and length (Ø1ØØH). On the adresses Ø2EC/ED and Ø3EA-ED Heap pointers are written. The last ones destroy your M.L. program, which is for the rest not affected by a reset.
If you start a M.L.program on Ø4ØØH, it can be restarted with a simple 'GO' after an accidently occured reset.

Jan Boerrigter

SAVING and LOADING of STRING ARRAYS.

Especially in word processing programs, string arrays (e.g. A$(N)) are used to store the data. These arrays can be saved on tape via the Basic instruction SAVEA.
During saving (and/or loading) of these arrays, several problems may occur, as for instance a 'OUT OF MEMORY' error report.

This article will describe what happens during saving and loading of string arrays. Read before what is written in Newsletter 1981, page 188,189 about string arrays and the Heap.

In the Heap, string arrays are stored in two parts: 1. the address where the strings can be found, and 2. the string itself. Strings are stored in the sequence they are assigned, so they do not have a fixed location inside the Heap. Before saving a string array, it is necessary to get al the data of the particular array together. Via the next program, the whole sequence is demonstrated.

```
10    CLEAR 2500
20    DIM A$(5.0),B$(5.0),N$(5.0),M$(5.0)
30    FOR N=0.0 TO 5.0:READ A$(N):READ B$(N):NEXT
40    FOR N=0.0 TO 5.0
50    FOR X=0.0 TO 9.0
60    N$(N)=N$(N)+A$(N):M$(N)=M$(N)+B$(N)
70    NEXT X:NEXT N                                        SAVE
80    STOP
90    SAVEA N$ "TEST 1"
100   STOP
110   SAVEA M$ "TEST 2"
120   END
130   DATA A,G,B,H,C,I,D,J,E,K,F,L
-------------------------------------------------------------
200   CLEAR 2500
210   DIM N$(5.0),M$(5.0)
220   PRINT "START TAPE"                                   LOAD
230   LOADA N$ "TEST 1"
240   PRINT "START TAPE"
250   LOADA M$ "TEST 2"
260   END
```

The two string arrays N$(5) and M$(5) are filled with the contents of 10 times a letter. N$(0)="AAAAAAAAAA"; M$(5)="LLLLLLLLLL".

When the program stops in line 80, the contents of the Heap is as follows:

```
02EC | 00 0E 01 05
02F0 | 2D 03 33 03 39 03 3F 03 45 03 4B 03 00 0E 01 05
0300 | 30 03 36 03 3C 03 42 03 48 03 4E 03 00 0E 01 05   string addresses
0310 | 5C 03 7F 03 97 03 AF 03 C7 03 DF 03 00 0E 01 05
0320 | 68 03 8B 03 A3 03 BB 03 D3 03 EB 03 00 01 41 00
-----+-------------------------------------------------
0330 | 01 47 00 01 42 00 01 48 00 01 43 00 01 49 00 01   A$(n), B$(n)
0340 | 44 00 01 4A 00 01 45 00 01 4B 00 01 46 00 01 4C
-----+-------------------------------------------------
0350 | 80 09 46 46 46 46 46 46 46 46 46 00 0A 41 41 41
0360 | 41 41 41 41 41 41 41 00 0A 47 47 47 47 47 47 47
0370 | 47 47 47 80 09 46 46 46 46 46 46 46 46 46 00 0A
0380 | 42 42 42 42 42 42 42 42 42 42 00 0A 48 48 48 48
0390 | 48 48 48 48 48 48 00 0A 43 43 43 43 43 43 43 43   N$(n), M$(n)
03A0 | 43 43 00 0A 49 49 49 49 49 49 49 49 49 49 00 0A
03B0 | 44 44 44 44 44 44 44 44 44 44 00 0A 4A 4A 4A 4A
03C0 | 4A 4A 4A 4A 4A 4A 00 0A 45 45 45 45 45 45 45 45
03D0 | 45 45 00 0A 4B 4B 4B 4B 4B 4B 4B 4B 4B 4B 00 0A
03E0 | 46 46 46 46 46 46 46 46 46 46 00 0A 4C 4C 4C 4C
03F0 | 4C 4C 4C 4C 4C 4C 88 B6
```

In the Heap, the addresspointers to the strings can be found, and the particular strings theirselves. All strings are found in the Heap in the sequence they are loaded. For instance parts of strings can be found from line 60 in the program, where the N$ and the M$ increase every time.

When now line 90 is runned, the following happens. The ROM-software dictates the DAI to order all the data belonging to the string N$. This is done in the free RAM space! If line 90 has been executed, you will find directly after the symboltable (ending on ØE27) all the string data in a neat order.

```
0E28|00 0C 0A 41 41 41 41 41
0E30|41 41 41 41 41 0A 42 42 42 42 42 42 42 42 42 42
0E40|0A 43 43 43 43 43 43 43 43 43 43 0A 44 44 44 44
0E50|44 44 44 44 44 44 0A 45 45 45 45 45 45 45 45 45
0E60|45 0A 46 46 46 46 46 46 46 46 46 46
```

The same happens after executing line 110. But no additional RAM space is used; the data of N$ is overwritten by the data of M$:

```
0E28|00 0C 0A 47 47 47 47 47
0E30|47 47 47 47 47 0A 48 48 48 48 48 48 48 48 48 48
0E40|0A 49 49 49 49 49 49 49 49 49 49 0A 4A 4A 4A 4A
0E50|4A 4A 4A 4A 4A 4A 0A 4B 4B 4B 4B 4B 4B 4B 4B 4B
0E60|4B 0A 4C 4C 4C 4C 4C 4C 4C 4C 4C 4C
```

Maybe now is clear why a 'OUT OF MEMORY' error report occur when a string array is to be saved on tape. When a large Heap is cleared, the program itself uses a lot of memory space, and the stringarray contains a lot of data, the free RAM space may be to small to contain all the string data to be saved.

When LOADING string arrays from tape, the same sequence is used. The data, read from tape, is stored in the free RAM space above textbuffer and symbol table.
When all reading has been done, the ROM-firmware initiates the re-organisation of the Heap: the string data is moved from the free RAM space into the Heap and the address pointers are updated.

This is the reason the DAI uses some time after reading string arrays from tape. You remember: after reading a string array (cursor do not flash), the cursor starts flashing, but the DAI is still busy for some time.
It is a failure in the software design that the cassette motor is not switched of after the end of the reading, but after the organisation of the Heap.

Final conclusion: When you want to use string arrays for saving data on tape, be careful in the memory space you use. For both saving and reading, you need almost the same memory space in the free RAM space as you need in the Heap for the particular string array. Every obstruction against this rule is fined with a 'OUT OF MEMORY' error report.

So take care!

Jan Boerrigter/jan.'82

DAINAMIC BOOTSTRAP LOADER

```
J#L.
PAGE 01     DBL 1.1   30DEC81
```


```
002  ********************************************************
003  *                                                      *
004  *       DAINAMIC  BOOTSTRAP  LOADER    ( DBL )        *
005  *       ---------------------------------------        *
006  *                                                      *
007  *  DBL IS USED FOR EASY LOADING OF PROGRAMS WHICH      *
008  *  HAVE A MACHINE LANGUAGE AND A BASIC PART.           *
009  *                                                      *
010  *  VERSIONS :                                          *
011  *     DBL 1.* : DELIVERED AS A BASIC PROGRAM AND       *
012  *               IS USABLE WITH CASSETTE, DCR AND       *
013  *               DISK BY TYPING *LOAD:RUN .             *
014  *     DBL 2.* : DELIVERED AS A UTILITY FILE AND        *
015  *               ONLY USABLE WITH DCR AS A 'USER'       *
016  *               FILE AND EXECUTED AFTER DAI RESET.     *
017  *                                                      *
018  ********************************************************
019  *
020  ********************************************************
021  *                                                      *
022  *                  D B L   V 1 . *                     *
023  *                  ---------------                     *
024  *                                                      *
025  *  THIS MACHINE LANGUAGE PROGRAM IS LOADED IN THE      *
026  *  STACKAREA OF THE RAM BY A BASIC PROGRAM.            *
027  *  CONTROL IS TRANSFERED TO IT WITH A CALLM #F000.     *
028  *                                                      *
029  *  FUNCTIONS :                                         *
030  *                                                      *
031  *    1) INIT STACKPOINTER.                             *
032  *                                                      *
033  *    2) LOAD THE FIRST UTILITY FILE ON TAPE.           *
034  *                                                      *
035  *    3) SET BEGIN OF HEAP AFTER CODE OF THE            *
036  *       UTILITY FILE AND PERFORM A *NEW TO ADJUST      *
037  *       OTHER BASIC POINTERS.                          *
038  *                                                      *
039  *    4) PRINT MESSAGE [LOAD BASIC] AND LOAD THE        *
040  *       FIRST BASIC PROGRAM ON TAPE.                   *
041  *                                                      *
042  *    5) FORCE A RUN OF THE LOADED BASIC PROGRAM.       *
043  *                                                      *
044  *  NOTES :                                             *
045  *                                                      *
046  *    1) TO START WITH A CLEAN SYSTEM, DO A RESET       *
047  *       BEFORE STARTING THE BOOTSTRAP LOADING          *
048  *                                                      *
049  *    2) IF THE BREAKKEY IS PRESSED :                   *
050  *       - BEFORE CONTROL IS TRANSFERD TO THIS MLP      *
051  *         ( = BEFORE OR DURING POKING OF THE CODE)     *
052  *         THEN TYPE *RUN TO RESTART                    *
053  *       - AFTER LOADING OF THE UTILITY FILE            *
054  *         ( = AFTER FUNCTION 1,2,3 )                   *
055  *         THEN TYPE *LOAD:RUN TO CONTINUE              *
```

```
056                     *       - AFTER LOADING THE BASIC                       *
057                     *         THEN TYPE *RUN TO START THE PROGRAM           *
058                     *                                                       *
059                     *    3) IF LOADING ERRORS OCCUR DURING :                *
060                     *       - LOAD OF BASIC PROGRAM DBL OR                  *
061                     *         LOAD OF UTILITY FILE                          *
062                     *         THEN RESTART THE WHOLE PROCESS                *
063                     *       - LOAD OF THE BASIC PROGRAM THEN RESTART        *
064                     *         ONLY LOAD OF BASIC : SET TAPE BEFORE          *
065                     *         BASIC AND TYPE *LOAD:RUN                      *
066                     *       LOADING ERRORS ARE REPORTED IN THE NORMAL       *
067                     *       DAI ERROR FORMAT.                               *
068                     *                                                       *
069                     *    4) AS DBL READS THE TAPE FILES WITHOUT             *
070                     *       FILENAMES, PUT THE FILES IN CORRECT             *
071                     *       SEQUENCE ON THE TAPE :                          *
072                     *       1- ODBL 1.* : BASIC TO POKE DBL IN RAM          *
073                     *       2- 1UT-FILE : YOUR MLP CODE                     *
074                     *       3- OBASIC   : YOUR BASIC PROGRAM                *
075                     *                                                       *
076                     *********************************************************
077                     *
078                     * DESCRIBES OF DAI REFERENCES
079                     * --------------------------
080                     CURRNT  EQU   :100      START OF CURRENT LINE
081                     HEAP    EQU   :29B      BEGIN HEAP
082                     SRBOT   EQU   :F800     BOTTOM OF STACK RAM
083                     SSTOP   EQU   :F900     TOP OF STACK RAM
084                     ROPEN   EQU   :2CE      SEARCH TAPE FILE,SYNC,CHECK
085                                             FILE TYPE,DISPL/CHECK NAME
086                     RBLK    EQU   :2D1      TRANSFER BLOCK OF BYTES FROM
087                                             TAPE TO RAM, EXIT :
088                                             IF NO LOADING ERRORS
089                                             THEN CY=1
090                                             ELSE CY=0 AND A=ERRORCODE
091                     RCLO    EQU   :2D4      STOP CASSETTE
092                     FORCEB  EQU   :C892     FORCE BASIC RUN IF NO BREAK
093                     ERRLD   EQU   :D2A8     ENTRY FOR TAPE ERROR MSG,
094                                             EXITS TO BASIC COMMAND MODE
095                     CRLF    EQU   :DD5E     PRINT A CAR RETURN
096                     RNEW    EQU   :DEB8     DELETE BASIC PROGRAM AND SET
097                                             UP BASIC POINTERS
098                     *
099                             ORG   SRBOT
100                     *
101 F800 3100F9         ENTRY   LXI   SP,SSTOP  INIT STACKPOINTER
102 F803 210000                 LXI   H,:0      READ WITHOUT NAME
103 F806 220001                 SHLD  CURRNT    SET NO PROGRAM RUNNING
104 F809 01FF31                 LXI   B,:31FF   FILETYPE 1 + PRINT NAMES
105 F80C CDCE02                 CALL  ROPEN
106 F80F 2154F8                 LXI   H,TEMP    TEMPORY SAVE AREA
107 F812 1156F8                 LXI   D,TEMP+2
108 F815 CDD102                 CALL  RBLK      READ BEGINADDR UT CODE
109 F818 2100B0                 LXI   H,:B000   MAX ADDR UT FILE
110 F81B EB                     XCHG
111 F81C 2A54F8                 LHLD  TEMP
112 F81F DCD102                 CC    RBLK      IF NO ERRORS READ BLOCK
113                                             WITH CONTENTS UT FILE
```

```
114 F822 CDD402              CALL  RCLO       ALWAYS STOP CASSETTE
115 F825 D2A8D2              JNC   ERRLD      ABORT IF ERRORS
116 F828 229B02              SHLD  HEAP       ADJUST HEAPBEGIN
117 F82B CDB8DE              CALL  RNEW       ADJUST OTHER POINTERS
118 F82E CD5EDD              CALL  CRLF
119 F831 0137F8              LXI   B,BASLIN   POINTER BASIC CMD LINE
120 F834 C392C8              JMP   :C892      START BASIC RUN
121                  *
122                  * FOLLOWING DATA IS EXECUTED AS A BASIC COMMAND LINE
123                  TRUN    EQU   :87        RUN (DAI INTERNAL CODE)
124                  TLOAD   EQU   :8B        LOAD
125                  TPRINT  EQU   :AD        PRINT
126                  *
127 F837 AD          BASLIN  DATA  TPRINT     PRINT MESSAGE
128 F838 01                  DATA  :01        1 EXPRESSION
129 F839 20                  DATA  :20        SEPARATOR
130 F83A 18                  DATA  :18        STRINGCONSTANT
131 F83B 11                  DATA  :11        STRINGLENGTH
132 F83C 5B204C              ASC   '[ LOADING BASIC ]'
133 F84D FF                  DATA  :FF        END PRINT WITH CR
134 F84E 8B                  DATA  TLOAD      LOAD BASIC PROGRAM
135 F84F 1900                DATA  :19,:00    DUMMY STRINGCONSTANT
136 F851 AD00                DATA  TPRINT,:00 PRINT ONLY CR
137 F853 87                  DATA  TRUN       START BASIC PROGRAM
138                  *
139 F854 0000        TEMP    DBL   0
140                  *
141 F856                     END

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

BASLIN F837   CRLF   DD5E   CURRNT 0100   ENTRY  F800
ERRLD  D2A8   FORCEB C892   HEAP   029B   RBLK   02D1
RCLO   02D4   RNEW   DEB8   ROPEN  02CE   SRBOT  F800
SSTOP  F900   TEMP   F854   TLOAD  008B   TPRINT 00AD
TRUN   0087
```

```
100  REM <<< DAINAMIC BOOTSTRAP LOADER  V1.1  30DEC81 >>>
110  REM /* THIS PROGRAM LOADS A BOOTSTRAP MLP IN RAM
120  REM /* THE MLP LOADS A UTILITY FILE AND A BASIC PROGRAM
130  REM /* BEGIN OF HEAP IS ADJUSTED AFTER THE MLP
140  REM /* THE BASIC PROGRAM IS ENTERED WITH A RUN
150  REM /* CONTENTS TAPE : DBL, UT-FILE, BASIC PROGRAM
160  REM /* SET TAPE BEFORE DBL, TYPE *LOAD:RUN, START CASSETTE
200  MODE 0:PRINT CHR$(12):COLORT 8 0 0 8
210  POKE #75,#5F:CLEAR #100
220  PRINT "*** DAINAMIC BOOTSTRAP LOADER V1.1 ***":PRINT
230  READ BGNADDR,NMBR
240  FOR ADDR=BGNADDR TO BGNADDR+NMBR-1
250  READ BYTE:POKE ADDR,BYTE
260  NEXT:READ C:IF C<>#FFFF THEN PRINT "DATA READ ERROR":PRINT :END
270  PRINT "[ LOADING UTILITY FILE ]"
280  CALLM BGNADDR
300  REM /* MLP BYTES
310  DATA #F800,86
320  DATA #31,#00,#F9,#21,  #00,#00,#22,#00,  #01,#01,#FF,#31,  #CD,#CE,#02,#21
330  DATA #54,#F8,#11,#56,  #F8,#CD,#D1,#02,  #21,#00,#B0,#EB,  #2A,#54,#F8,#DC
340  DATA #D1,#02,#CD,#D4,  #02,#D2,#A8,#D2,  #22,#9B,#02,#CD,  #B8,#DE,#CD,#5E
350  DATA #DD,#01,#37,#F8,  #C3,#92,#C8,#AD,  #01,#20,#18,#11,  #5B,#20,#4C,#4F
360  DATA #41,#44,#49,#4E,  #47,#20,#42,#41,  #53,#49,#43,#20,  #5D,#FF,#8B,#19
370  DATA #00,#AD,#00,#87,  #00,#00,#FFFF
```

BASIC HANDBOEKEN—MANUALS

Voorbeelden en informatie vind je ondermeer in volgende boeken :

I DAI personal computer manual - DAI microcomputer engeneering company.
II Basic voor je personal computer - Stichting BASICned Hollandsche Rading.
III Cursusboek bij Teleaccursus MICROPROCESSORS 2 - Stichting Teleac Utrecht.
IV User's manual for level 1 Basic (TRS-80)
V Handleiding voor Basic level 2 "

		I	II	III	IV	V
01	CHECK	74		75		
02	CLEAR	78		191		4/2
03	COLORG	82		280		
04	COLORT	82		279		
05	CONT	77			53	2/3
06	CURSOR	87		283		
07	DATA	71	110	92	85	3/8
08	DIM	78	158	100		4/3
09	DOT	83	249	281		
10	DRAW	83	250	281		
11	EDIT	61		43		9/1
12	END	64	53	75	13	4/4
13	ENVELOPE	92		284		
14	FILL	83	250	282		
15	FOR NEXT	64	134	67	45	4/8
16	GPSUB	65	227	81	81	4/6
17	GOTO	66	60	40	29	4/5
18	IF GOTO	66		59	29	
19	IF THEN	66	99	52	29	4/12
20	IMP	48				
21	INPUT	72	64	34	33	3/7
22	LET	79	47	31	18	4/4
23	LIST	62	63	44	13	2/4
24	LOAD	74		74		
25	LOADA	75		107		
26	MODE	80	249	280		
27	NEW	63	51	42	7	
28	NEXT	65	145	71	45	4/8
29	NOISE	93		285		
30	ON GOSUB	66		84	78	4/7
31	ON GOTO	67	97	77	78	4/6
32	OUT	69		261		8/4
33	POKE	70		267		8/5
34	PRINT	73	62	32	9	3/1
35	READ	73	118	94	85	3/9
36	REM	77	98	45		4/12
37	RESTORE	73	217	98	87	
38	RETURN	67	232	81	82	4/6
39	RUN	63	66	45	9	
40	SAVE	75		74		
41	SAVEA	76		107		
42	SOUND	92		284		
43	STOP	68	104	81	56	4/5
44	TALK	94				
45	TROFF	78				2/5
46	TRON	78				2/5
47	WAIT	68		271		
48	UT	71		191		
49	ABS	96	239	113	93	7/1
50	ACOS	96		123		
51	ALOG	96				
52	ASC	96	215	157		5/3
53	ASIN	96		122		
54	ATN	96		123		7/1
55	CHR $	97	209	159		5/4
56	COS	97		120		7/2
57	CURX	88		283		
58	CURY	88		283		
59	EXP	97		124		7/2
60	FRAC	97		115		
61	FRE	79				5/5
62	FREQ	93		285		
63	GETC	72		301		
64	HEX $	97		191		
65	INP	69		261		8/4
66	INT	98	90	113	69	7/3
67	LEFT $	98	221	155		5/6
68	LEN	98	221	154		5/6
69	LOG	98		124		7/3
70	LOGT	98		124		
71	MID $	99	219	156		5/6
72	PDL	70				
73	PEEK	70		266		8/5
74	PI	99		121		
75	RIGHT $	99	221	156		5/7
76	RND	99	87	116	99	7/3
77	SCRN	87		283		
78	SGN	100		115		7/4
79	SIN	100		120		7/4
80	SPC	100	249	160		
81	SQR	100		116		7/7
82	STR $	100	213	158		5/7
83	TAB	101	248	160	61	
84	TAN	101		121		7/4
85	VAL	101	212	159		5/8
86	VARPTR	79		270		8/8
87	XMAX	87		282		
88	YMAX	87		282		
89	MOD	47				
90/91	IOR/IAND	104		251		
92/93	IXOR/INOT	104		251		
94/95	SHL/SHR	102		263		

CARPENTERS SOLUTION

```
T
2010  COMMAND$="515005152271140413603625050336263141143422737227372050
0533626311414360502714041140412272 70"
2020  COMMAND$=COMMAND$+"0515036336052271243362633626052273714061114 04
1270051500515"
2030  COMMAND$=COMMAND$+"360361211427050503631411427372273705250053362
631143140631600515022 7"
```

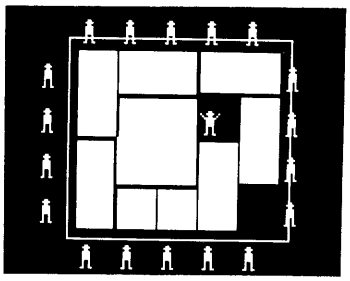
```
2010  COMMAND$="515005152271140413603625/033626314114342273722737205005"
2020  COMMAND$=COMMAND$+"33626311434142700515036/140411404127005150362 4/22737205336263362614227372053636163114041272270051500515"
2030  COMMAND$=COMMAND$+"271140436250535036263141142737227370525005333626311414363600515022 7"
```

```
2010  COMMAND$="505115052271140413636050250336263141143422727337272050053362631 1
414360502714140041 4/"
2020  COMMAND$=COMMAND$+"122727005150363360522721433626336260522737140611140412 7
005051150536114 04/"
2030  COMMAND$=COMMAND$+"273505250336263141142737227370525005333626311414363600515
50227"
*
```

```
2010  COMMAND$="515005152271140413603625050336263143114242737227372050053362631 1
414360502714041140413600515002735222737214 3362633626052271436363114041272270051500515271143605053062 6
314114273722737052500533626311414363600515022 7"
```

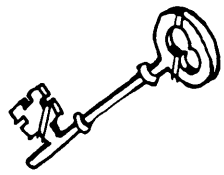
```
515005152271140413603625050336263143114242737227372050053362631141436050271404 11
404136005150027352227372143362633626052271436363114041272270051500515271143605 05
306263141142737227370525005336263114143636005150227
 211
```

```
3299  REM CORRECTIONS AS SUGGESTED BY K..VERBEKE
3300  FOR Y=0 TO 19:GOSUB 4020
3320  DRAW A*20,B*20+YO+Y+1 A*20+XO,B*20+YO+Y+1 CO
3450  FOR Y=0 TO 19:GOSUB 4020
3470  DRAW A*20,B*20-Y-1 A*20+XO,B*20-Y-1 CO
3620  FOR Y=1 TO 20:GOSUB 4020
3630  DRAW A*20+Y+1+XO,B*20 A*20-Y+XO+1,B*20+Y CO
3830  FOR Y=1 TO 20:GOSUB 4020
3840  DRAW A*20+Y-1,B*20 A*20+Y-1,B*20+YO CO
```



```
T
3299  REM CORRECTIONS AS SUGGESTED BY M.STREICHER
3310  DRAW A*20,B*20+Y-1 A*20+XO,B*20+Y-1 20
3460  DRAW A*20,B*20+YO-Y+1 A*20+XO,B*20+YO-Y+1 20
3630  DRAW A*20-Y+XO+1,B*20 A*20-Y+XO+1,B*20+YO 20
3850  DRAW A*20+Y-1,B*20 A*20+Y-1,B*20+YO 20
```

SIP TRICKS



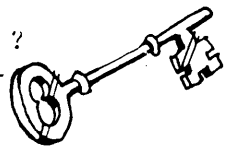
Un petit programme qui intéressera les possesseurs d'une imprimante.
Vous avez sûrement déjà remarqué qu'avec un "check" le titre des programmes n'apparaît pas sur l'inprimante.
Ce programme vous permettra de lister les titres des programmes contenus sur une cassette digitale (pour un DCR, tapez d'abord DCR(REW) ou CAS au choix).
Quand vous voulez arrêter l'impression, tapez espace.

```
PAGE  1  --   CASSETTE LIST                                    --  FLST V2.2


5          REM CASSETTE LIST
10         MODE 0:PRINT CHR$(12);:POKE #131,1:AD$=""
30         FOR A=1 TO 21:READ B:AD$=AD$+CHR$(B):NEXT
40         AD=PEEK(VARPTR(AD$)) IOR (PEEK(VARPTR(AD$)+1) SHL 8)+1
100        INPUT "CASSET'S NAME";CN$:PRINT
120        POKE #131,0:PRINT CN$:FOR A=0 TO LEN(CN$)-1:PRINT "-";:NEXT:
C          PRINT :CPT=1
1000       PRINT CHR$(12);:CALLM AD:PRINT :PRINT CPT;" - ";
1030       FOR A=#BFE7 TO #BF85 STEP -2:PRINT CHR$(PEEK(A));:NEXT
1035       IF GETC<>0 THEN 2000
1040       CPT=CPT+1:GOTO 1000
2000       POKE #131,1:END
10000      DATA #F5,#C5,#D5,#E5,#01,#40,#00,#11,#B1,#80,#21,#9E,#E6,
C          #CD,#CE,#02,#E1,#D1,#C1,#F1,#C9

PAGE  1  --   WHAT TO DO IF A CRASH OCCURS....                  --  FLST V2.2


1          REM WHAT TO DO IF A CRASH OCCURS....
2          INPUT "START OF HEAP (#2EC)";START:PRINT :DEP=START
10         COM=DEP:DIZ=0
20         OF=PEEK(COM):NUM=(PEEK(COM+1) SHL 8) IOR PEEK(COM+2):IF
C          NUM>2000.0 OR NUM<1.0 THEN DEP=DEP+1:IF DEP<#A000 THEN 10
40         IF NUM/10=INT(NUM/10.0) THEN DIZ=DIZ+1
50         IF DIZ>10 THEN 70
60         COM=COM+1+OF:GOTO 20
70         STEXT=DEP
80         COM=DEP
90         OF=PEEK(COM):NUM=(PEEK(COM+1) SHL 8) IOR PEEK(COM+2):IF
C          NUM>#FFFF OR NUM=0 OR OF=0 THEN 120
110        COM=COM+1+OF:GOTO 90
120        COM=COM+1:FINP=COM
130        A=PEEK(COM) IAND #F:IF A=0 THEN 200
140        COM=COM+A+1:B=PEEK(COM) IAND #F:COM=COM+B+1:GOTO 130
200        ESY=COM+1
210        PRINT "HEAP:";HEX$(STEXT-START)
220        PRINT "!!! START OF TEXT:";HEX$(STEXT);" FIRST LINE=
C          ";(PEEK(DEP+1) SHL 8) IOR PEEK(DEP+2)
230        PRINT "END OF TEXT:";HEX$(FINP)
240        PRINT "END OF SYMBOL TABLE:";HEX$(ESY)
```



Beaucoup d'entre vous ont sûrement déjà été "ennuyés" par des "resets intempestifs".

Ces accidents, qu'ils soient dus a la machine ou a l'homme-qui-programme-la-machine, arrivent toujours au début d'un programme (vous savez, après la 400ème ligne environ).

Voici donc un petit programme qui vous permettra de "récupérer" à 99,999 % des cas un programme "planté" (à condition qu'il ne se soit pas auto-détruit).

Voici la procédure à suivre :

1 - Ne pas paniquer ni jurer.

2-- Faire reset (si ce n'est pas déjà fait).

3 - Taper UT
Z3
S29B 02-00 EC-A0
B
NEW

4 - Charger le programme "What to do when a crash occurs".

5 - Faire "RUN".

6 - A la question "START OF HEAP?", taper- 2EC pour un programme normal;
-un autre chiffre si la mémoire était protégée par du langage machine
ex : 900 pour FGT.

7 - Prier.

8 - Le programme donne : HEAP
START OF TEXT
START OF SYMBOL TABLE
END OF SYMBOL TABLE

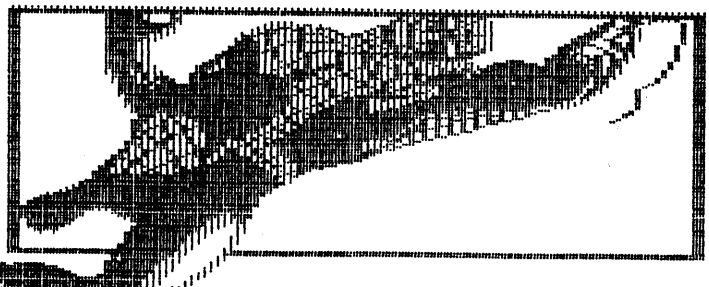
9 - Remplacer les pointeurs de 29B et 2A4 en n'oubliant pas d'inverser LSB et MSB.

10- Revenir au basic.

11- Faire "LIST".

12- Si rien ne se passe, jurer un bon coup!

Robert SIP.



```
5     CLEAR 100
6     POKE #75,32
10    ENVELOPE 0 15:ENVELOPE 1 15:MODE 0:PRINT CHR$(12);:COLORT 8 8 8 8:F$="  format :"
110   PRINT "          ***  D A I   B A S I C  T U T O R   ***":GOSUB 600:PRINT
140   PRINT " select a number   ...   (2 digits)":GOSUB 600
150   PRINT " 01 CHECK    02 CLEAR    03 COLORG   04 COLORT   05 CONT"
160   PRINT " 06 CURSOR   07 DATA     08 DIM      09 DOT      10 DRAW"
170   PRINT " 11 EDIT     12 END      13 ENVELOPE 14 FILL     15 FOR NEXT"
180   PRINT " 16 GOSUB    17 GOTO     18 IF GOTO  19 IF THEN  20 IMP"
190   PRINT " 21 INPUT    22 LET      23 LIST     24 LOAD     25 LOADA"
200   PRINT " 26 MODE     27 NEW      28 NEXT     29 NOISE    30 ON GOSUB"
210   PRINT " 31 ON GOTO  32 OUT      33 POKE     34 PRINT    35 READ"
220   PRINT " 36 REM      37 RESTORE  38 RETURN   39 RUN      40 SAVE"
230   PRINT " 41 SAVEA    42 SOUND    43 STOP     44 TALK     45 TROFF"
240   PRINT " 46 TRON     47 WAIT     48 UT       49 ABS      50 ACOS"
250   PRINT " 51 ALOG     52 ASC      53 ASIN     54 ATN      55 CHR$"
260   PRINT " 56 COS      57 CURX     58 CURY     59 EXP      60 FRAC"
270   PRINT " 61 FRE      62 FREQ     63 GETC     64 HEX$     65 INP"
280   PRINT " 66 INT      67 LEFT$    68 LEN      69 LOG      70 LOGT"
290   PRINT " 71 MID$     72 PDL      73 PEEK     74 PI       75 RIGHT$"
300   PRINT " 76 RND      77 SCRN     78 SGN      79 SIN      80 SPC"
310   PRINT " 81 SQR      82 STR$     83 TAB      84 TAN      85 VAL"
320   PRINT " 86 VARPTR   87 XMAX     88 YMAX     89 MOD      90 IOR"
330   PRINT " 91 IAND     92 IXOR     93 INOT     94 SHL      95 SHR";
332   POKE #BEE2,204:POKE #BDD6,200:POKE #BFEE,205
340   COLORT 8 0 12 0
405   M$="":FOR X=1.0 TO 2.0
420   G=GETC:IF G=0.0 THEN 420
430   IF G<48.0 OR G>57.0 THEN 420:M$=M$+CHR$(G):WAIT TIME 3
440   SOUND 1 0 15 0 FREQ(400.0*X):G=0.0:WAIT TIME 3:G=0.0:SOUND OFF :NEXT:M=VAL(M$):IF M<1 THEN 405
460   P%=#BFEE+10-(5*#86)-(INT(M/5.0))*#86-(INT(M MOD 5.0)*24.0):IF M MOD 5.0=0.0 THEN P%=P%+14
470   FOR X%=P%-18 TO P%+2 STEP 2:POKE X%,#FF:NEXT:WAIT TIME 100:MODE 0:PRINT CHR$(12);
510   ON M GOTO 1100,1200,1300,1400,1500,1600,1700,1800,1900,2000
520   ON M-10 GOTO 2100,2200,2300,2400,2500,2600,2700,2800,2800,3000
530   ON M-20 GOTO 3100,3200,3300,3400,3500,3600,3700,2500,3900,4000
540   ON M-30 GOTO 4100,4200,4300,4400,1700,4600,1700,2600,4900,5000
550   ON M-40 GOTO 5100,5200,5300,5400,5500,5500,5700,5800,5900,6000
560   ON M-50 GOTO 6100,6200,6000,6000,6500,6600,6700,6800,6900,7000
570   ON M-60 GOTO 7100,5200,7300,7400,7500,7600,7700,7700,7900,7900
580   ON M-70 GOTO 7700,8200,8300,8400,7700,8600,8700,8800,6600,9000
590   ON M-80 GOTO 9100,9200,9300,6600,9200,9600,3600,3600,9900,10000
595   IF M>90 GOTO 10000
600   FOR X=1.0 TO 60.0:PRINT CHR$(11);:NEXT:PRINT :RETURN
800   CURSOR 30,1:PRINT "spacebar to continue ..."
830   G=GETC:IF G<>32.0 THEN 830
840   GOTO 10
850   CURSOR 30,1:PRINT "spacebar to continue...";
860   G=GETC:IF G=0.0 THEN 860
870   RETURN
900   POKE #BFEE,198:POKE #BE5C,200:GOSUB 600:PRINT TAB(25);A$:GOSUB 600:RETURN
```

CHECK

```
1100  A$="CHECK":GOSUB 900:PRINT "The CHECK command scans a cassette tape or disc and "
1120  PRINT "examines all the files.The type of each is printed":PRINT "followed by the word Ok or B
AD."
1130  PRINT :PRINT TAB(20);"0 = BASIC PROGRAM":PRINT TAB(20);"1 = MACHINE LANGUAGE FILE"
1140  PRINT TAB(20);"2 = ARRAY":GOSUB 600:GOTO 800
```

CLEAR

```
1200  A$="CLEAR":GOSUB 900:PRINT F$;TAB(25);"10 CLEAR 4000"
1217  PRINT TAB(25);"CLEAR 4000":GOSUB 600
1220  PRINT "Resets all variables to 0 or null string,and returns"
1230  PRINT "all space assigned to arrays.The size of the HEAP "
1240  PRINT "is set to the number specified by the CLEAR statement."
1250  PRINT "minimum = 4 , maximum = 32767"
1255  PRINT "If you get  'OUT OF STRING SPACE',the HEAP is too small":GOSUB 600:GOTO 800
```

COLORG

```
1300  A$="COLORG":GOSUB 900:PRINT F$;TAB(20);"10 COLORG 0 5 10 15 (default values)"
1317  PRINT TAB(20);"COLORG 1 3 5 8":GOSUB 600
1320  PRINT "Sets the 4 colours available in 4 COLOR modes (2,4,6)"
1330  PRINT "If the screen is already in 4 COLOR mode,"
1340  PRINT "then the colour change will be immediate"
1345  PRINT "The first colour is background colour":GOSUB 600:GOSUB 850
1360  MODE 2A:PRINT CHR$(12);:COLORG 0 1 3 5
1370  FOR X=0.0 TO 10.0:FOR Y=0.0 TO 10.0:FILL Y*4,X*4 Y*4+2,X*4+2 21+RND(3.0):NEXT:NEXT
1375  PRINT CHR$(12);TAB(10);"COLORG 0 1 3 5":GOSUB 850
1377  PRINT CHR$(12);TAB(10);"COLORG 1 5 12 14":COLORG 1 5 12 14:GOSUB 850
1379  PRINT CHR$(12);TAB(10);"COLORG 5 0 1 13":COLORG 5 0 1 13:GOTO 800
```

COLORT

```
1400  GOSUB 600:PRINT TAB(25);"COLORT":GOSUB 600
1410  PRINT "Sets up 4 colours for character mode (MODE 0)"
1420  PRINT "First colour is background colour":PRINT "Second colour is character colour"
1430  PRINT "Colours nrs 3 and 4 are used if you POKE #FF"
1435  PRINT "into the colour byte of the character":GOSUB 600
1450  PRINT F$;TAB(25);"10 COLORT 8 0 12 0":PRINT TAB(25);"COLORT 8 0 12 0":GOSUB 600:PRINT :PRINT
1460  FOR X=#BC68 TO #BC7A STEP 2.0:POKE X,#FF:NEXT:LIST 1460:GOSUB 850
1465  CURSOR 5,5:PRINT "COLORT 12 0 13 1  ":COLORT 12 0 13 1:GOSUB 850
1480  CURSOR 5,5:PRINT "COLORT 5 0 7 0   ":COLORT 5 0 7 0:GOTO 800
```

CONT

```
1500  A$="CONT":GOSUB 900:PRINT "Continues a BASIC program execution "
1520  PRINT "with the next statement following ":PRINT "the STOP statement or BREAK position"
1540  PRINT "You can't CONT if you used UT,EDIT"
1550  PRINT "You can  LIST or PRINT variables and then CONT":GOSUB 600:GOTO 800
```

CURSOR

```
1600  A$="CURSOR":GOSUB 900:PRINT "Moves the CURSOR to the X,Y position"
1620  PRINT "given in the statement":PRINT " maximum X = 59 , maximum Y = 23":GOSUB 600:GOSUB 850
1630  CURSOR 10,4:PRINT CHR$(1);"CURSOR 10,4":GOSUB 850:CURSOR 25,10:PRINT CHR$(1);"CURSOR 25,10":GO
SUB 850
1645  CURSOR 40,15:PRINT CHR$(1);"CURSOR 40,14":GOTO 800
```

DATA READ RESTORE

```
1700  A$="DATA":GOSUB 900:PRINT "Specifies information to be used by READ"
1720  PRINT "DATA must give the same type of information":PRINT "as READ is asking for "
1735  PRINT "RESTORE points back to the first DATA-information":GOSUB 600:RESTORE
1750  DATA 999,VIS,75,FELLOW,10000,HERE YOU ARE
```

```
1760  LIST 1750:LIST 1770:PRINT :PRINT
1770  FOR X%=1 TO 3:READ A%,A$:PRINT TAB(10);A%,A$:NEXT
1772  P%=#BFEE-(#86*8):P1%=P%-30:P2%=P%-58-#86
1774  FOR X%=P1% TO P1%+10 STEP 2:POKE X%,#FF:NEXT
1776  FOR X%=P2%+2 TO P2%+10 STEP 2:POKE X%,#FF:NEXT:GOTO 800
```

DIM

```
1800  A$="DIM":GOSUB 900:PRINT "DIM allocates space for arrays          A(250)"
1802  PRINT "arrays can have more than one suscript eg:A$(10,5,20)"
1803  PRINT "the HEAP is used to store the arrays":PRINT "EDIT or RUN clears the arrays":GOSUB 600:G
OTO 800
```

DOT

```
1900  A$="DOT":GOSUB 900:PRINT "Places a dot of colour C at the position X,Y"
1912  PRINT "The size of the DOT will depend upon the current MODE"
1914  GOSUB 600:PRINT :PRINT F$;TAB(25);"DOT X,Y C":GOSUB 600:GOSUB 850
1920  COLORG 0 1 3 5:MODE 1A:PRINT CHR$(12):CURSOR 25,3:PRINT "MODE 1A":GOSUB 1980
1930  MODE 3A:PRINT CHR$(12);:CURSOR 25,3:PRINT "MODE 3A":GOSUB 1980
1940  MODE 5A:PRINT CHR$(12);:CURSOR 25,3:PRINT "MODE 5A":GOSUB 1980:GOTO 800
1980  GOSUB 600:A%=10:B%=10:IF M=10 THEN 1990
1981  FOR X%=1 TO 15
1982  DOT A%,B% X%:CURSOR 5,1:PRINT "DOT ";A%;",";B%;X%;"    "
1983  A%=A%+4:B%=B%+3:GOSUB 850:NEXT:RETURN
1990  FOR X%=1 TO 15
1992  DRAW A%,B% A%+5,B% X%:CURSOR 5,1:PRINT "DRAW ";A%;",";B%;A%+5;",";B%;X%;"      "
1994  A%=A%+4:B%=B%+3:GOSUB 850:NEXT:RETURN
```

DRAW

```
2000  A$="DRAW":GOSUB 900:PRINT "Draws a line in colour C between X1,Y1 and X2,Y2"
2020  PRINT "There is no restriction on the order of the coordinates":GOSUB 600
2040  PRINT F$;TAB(25);"DRAW X1,Y1 X2,Y2 C":GOSUB 600:GOSUB 850:GOTO 1920
```

EDIT

```
2100  A$="EDIT":GOSUB 900:PRINT "the program has stopped"
2120  PRINT "You are in command now":PRINT "Go to EDIT MODE with : EDIT 2110-2190 <return>"
2140  PRINT "Use the cursorkeys to make modifications to this text"
2150  PRINT "If you end with BREAK BREAK,":PRINT "Then no modifications are effective"
2170  PRINT "If you want some modifications then do BREAK SPACE"
2180  PRINT "To see the results type LIST 2110-2190 <return>"
2190  PRINT "Type RUN <return> to continue the program":GOSUB 600:END
```

END

```
2200  A$="END":GOSUB 900:PRINT "Terminates the execution of a BASIC program."
2220  PRINT "An 'END PROGRAM' message is displayed"
2230  PRINT "Type RUN <return> to start again":GOSUB 600:END
```

ENVELOPE

```
2300  A$="ENVELOPE":GOSUB 900:PRINT "format : ENVELOPE nr (V,T);(V,T);(V,T);V"
2320  PRINT "          ENVELOPE nr (V,T);(V,T);(V,T);":GOSUB 600
2330  PRINT "nr is an expression in the range 0 to 1"
```

```
2335  PRINT "V is a volume level in the range 0 to 15"
2340  PRINT "T is the time for volume V in the range 1 to 254"
2345  PRINT "T is in units of 3.2 milliseconds":GOSUB 600:GOSUB 850
2355  MODE 6A:COLORG 0 15 12 14:FOR X%=21 TO 23:A%=100
2361  PRINT CHR$(12);:PRINT "ENVELOPE 1 ";
2362  Y%=(X%-20)*50:Q%=Y%:DRAW 100,Y% 235,Y% X%
2364  V%=RND(16.0):T%=RND(30.0)
2365  Z%=Y%+V%*2:DRAW A%,Z% A%+T%,Z% X%:DRAW A%,Z% A%,Q% X%:Q%=Z%
2367  PRINT "(";V%;",";T%;");";:SOUND 1 0 V% 0 FREQ(800.0):WAIT TIME T%
2370  A%=A%+T%:IF A%+20>=XMAX-100 THEN GOSUB 850:GOTO 2380
2372  GOTO 2364
2380  NEXT:SOUND OFF :GOTO 800
```

FILL

```
2400  A$="FILL":GOSUB 900:PRINT "format : FILL X1,Y1 X2,Y2 C":GOSUB 600
2420  PRINT "Fills the rectangle with opposite corners "
2425  PRINT "There is no restriction on the order of the points":GOSUB 600:GOSUB 850
2440  COLORG 8 1 3 5:MODE 3A:FOR X%=0 TO 15:FILL 0,X%*6 XMAX,X%*6+4 X%
2465  CURSOR 4,2:PRINT "FILL 0,";X%*6;XMAX;",";X%*6+4;X%;" ":GOSUB 850
2477  IF X%=0.0 THEN FILL 0,0 XMAX,YMAX 0
2480  NEXT:GOTO 800
```

FOR NEXT

```
2500  A$="FOR    NEXT":GOSUB 900:PRINT "To set up a loop a number of times"
2520  PRINT "A STEP value is optional":GOSUB 600:GOSUB 850
2540  PRINT CHR$(12);:A$="SOME EXAMPLES...":GOSUB 900:LIST 2560:GOSUB 600
2560  FOR X%=1 TO 10:PRINT X%:NEXT:GOSUB 850
2562  PRINT CHR$(12);:GOSUB 600:LIST 2564:GOSUB 600
2564  FOR X%=10 TO 200 STEP 10:PRINT X%:NEXT:GOSUB 850
2567  PRINT CHR$(12);:GOSUB 600:LIST 2569:GOSUB 600
2569  FOR X%=30 TO 13 STEP -1:FOR Y%=5 TO 20:PRINT X%+Y%;:NEXT:PRINT :NEXT:GOTO 800
```

GOSUB RETURN

```
2600  A$="GOSUB":GOSUB 900:PRINT F$;TAB(25);"40  GOSUB 1000":GOSUB 600
2620  PRINT "Branches to the specified linenumber.":PRINT "When a return statement is encountered th
e next statement"
2624  PRINT "executed is the statement following the GOSUB."
2626  PRINT "To be safe,only leave the subroutine with RETURN":GOSUB 600:LIST 2630-2640
2629  PRINT :PRINT :PRINT CHR$(1);:LIST 2670-2674
2630  FOR X=1.0 TO 3.0
2634  CURSOR 1,1:INPUT "type a name <return>";A$
2635  GOSUB 2670:REM to the subroutine
2640  NEXT:GOTO 800
2670  CURSOR CURX-LEN(A$),CURY-1
2672  FOR X=LEN(A$)-1.0 TO 0.0 STEP -1.0:PRINT MID$(A$,X,1);
2674  NEXT:RETURN
```

GOTO

```
2700  A$="GOTO":GOSUB 900:PRINT F$;TAB(25);"20 GOTO 1000":GOSUB 600
2720  PRINT "Branches to the specified linenumber":GOSUB 600:GOTO 800
```

IF GOTO / IF THEN

```
2800  A$="IF THEN / IF GOTO":GOSUB 900
2805  PRINT F$;TAB(25);"20 IF A=5 THEN SOUND OFF"
2806  PRINT TAB(25);"40 IF A=5 THEN 100":PRINT TAB(25);"60 IF A=5 GOTO 100":GOSUB 600
2808  PRINT "If the condition is true,execute following statement"
2809  PRINT "IF THEN can be followed by a statement or a linenumber"
2810  PRINT "IF GOTO must be followed by a linenumber":GOSUB 600:PRINT "attention":PRINT "---------"
2830  PRINT "with IF GOTO lnr. and IF THEN lnr.,if the condition is false":PRINT "the next statement
 on the same line is executed"
2850  PRINT "This does not apply for IF THEN GOTO":GOSUB 600:GOTO 800
```

IMP

```
3000  A$="IMP":GOSUB 900:PRINT F$;TAB(25);"IMP INT":PRINT TAB(25);"IMP INT M-Z"
3007  PRINT TAB(25);"IMP FPT":PRINT TAB(25);"IMP FPT L-O":PRINT TAB(25);"IMP STR A-D":GOSUB 600
3012  PRINT "If the IMP statement is never used,":PRINT "DAIpc is in IMPLIED FLOATING POINT"
3016  PRINT "In this situation A=A! I=I! Z=Z!":PRINT "I% , A% , Z% are integer and distinct from I ,
 A , Z"
3020  PRINT "I$ , A$ , Z$ are string variables":GOSUB 600
3030  PRINT "note the different formats :":PRINT TAB(10);"FPT";TAB(25);"INT";TAB(40);"STR"
3040  FOR X%=1 TO 5:PRINT TAB(10);A;TAB(25);A%;TAB(40);A$
3050  A=A+1.23456E5:A%=A%+12345669:A$=A$+CHR$(65+RND(20.0)):NEXT:GOTO 800
```

INPUT

```
3100  A$="INPUT":GOSUB 900:PRINT F$;TAB(25);"20 INPUT A"
3115  PRINT TAB(25);"30 INPUT A$":PRINT TAB(25);"40 INPUT B,B$,B%"
3125  PRINT TAB(25);"50 INPUT 'HOW MANY TIMES';T":PRINT TAB(25);"60 INPUT 'WHAT'S YOUR NAME';NAME$":
GOSUB 600
3150  PRINT "Requests data from the terminal to be typed in."
3160  PRINT "Data typed in must be of the same type as the variable(s)":GOTO 800
```

LET

```
3200  A$="LET":GOSUB 900:PRINT F$;TAB(25);"LET W=1234"
3220  PRINT TAB(25);"LET O$='?CROCODILE'":PRINT TAB(25);"M=9876"
3240  PRINT TAB(25);"HOME$='DAInamic'":GOSUB 600
3250  PRINT "Assigns a value to a variable":PRINT "LET is optional":GOSUB 600:GOTO 800
```

LIST

```
3300  A$="LIST":GOSUB 900:PRINT F$;TAB(15);"LIST :         display the entire program"
3315  PRINT TAB(15);"LIST 100";TAB(30);"display line 100"
3320  PRINT TAB(15);"LIST 100-";TAB(30);"display from 100 to the end"
3325  PRINT TAB(15);"LIST 100-110";TAB(30);"display from 100 to 110"
3330  PRINT TAB(15);"LIST -100";TAB(30);"display from start to 100"
3335  PRINT TAB(15);"30 LIST 999";TAB(30);"with linenumber":GOSUB 600
3340  PRINT "Display can be made to pause by pressing any key"
3350  PRINT "Pressing spacebar will continue listing":GOSUB 600:GOTO 800
```

LOAD

```
3400  A$="LOAD":GOSUB 900:PRINT F$;TAB(25);"LOAD":PRINT TAB(25);"LOAD:RUN"
3420  PRINT TAB(25);"10 LOAD":PRINT TAB(25);"10 LOAD 'TEST'":GOSUB 600
3445  PRINT "When BASIC encounters a LOAD command,":PRINT "It loads the first program from tape"
3449  PRINT "The old program is deleted ":PRINT "and the new one starts when loading is finished":GO
SUB 600:GOTO 800
```

LOADA

```
3500  A$="LOADA":GOSUB 900:PRINT F$;TAB(25);"10 LOADA A$ 'APPLE'"
3517  PRINT TAB(25);"20 LOADA M$":PRINT TAB(25);"LOADA M$ 'TRIAL'":GOSUB 600
3520  PRINT "Loads an array from tape/disc":GOTO 800
```

MODE XMAX YMAX

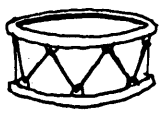
```
3600  A$="MODE":GOSUB 900:PRINT F$;TAB(25);"MODE 5A"
3607  PRINT TAB(25);"100 MODE 3":GOSUB 600:COLORG 0 1 3 12
3610  PRINT "Sets up display format":PRINT "There are 13 modes available :":GOSUB 600
3625  PRINT "MODE  GRAPHIC RESOLUTION";TAB(30);"4/16 COLOR      REQUIRED SPACE":PRINT
3627  PRINT "      XMAX YMAX"
3630  PRINT " 0    24 x 60 characters";TAB(30);"both            3.5K "
3632  PRINT " 1    72 x 65";TAB(30);"16              1.5K"
3636  PRINT " 2    72 x 65";TAB(30);"4               1.5K"
3640  PRINT " 3    160 x 130";TAB(30);"16              5.8K"
3644  PRINT " 4    160 x 130";TAB(30);"4               5.8K"
3648  PRINT " 5    336 x 256";TAB(30);"16             22.8K"
3652  PRINT " 6    336 x 256";TAB(30);"4              22.8K"
3654  PRINT :PRINT " 1A,2A,3A,4A,5A,6A same as 1,2,3,4,5,6 but split MODE":GOSUB 600:GOSUB 850
3660  PRINT CHR$(12):PRINT TAB(25);"MODE 1A":MODE 1A:GOSUB 3690:REM 16 C
3662  PRINT CHR$(12):PRINT TAB(25);"MODE 2A":MODE 2A:GOSUB 3695:REM 4C
3664  PRINT CHR$(12):PRINT TAB(25);"MODE 3A":MODE 3A:GOSUB 3690:PRINT CHR$(12):PRINT TAB(25);"MODE 4
A ":MODE 4A:GOSUB 3695
3666  PRINT CHR$(12):PRINT TAB(25);"MODE 5A":MODE 5A:GOSUB 3690:PRINT CHR$(12):PRINT TAB(25);"MODE 6
A":MODE 6A:GOSUB 3695
3670  GOTO 800
3690  CURSOR 2,1:PRINT "XMAX =";XMAX;" YMAX =";YMAX;"   "
3692  FOR X%=0 TO YMAX STEP 2:DRAW 0,X% XMAX,X% RND(16.0):NEXT:GOSUB 850:RETURN
3695  CURSOR 2,1:PRINT "XMAX =";XMAX;" YMAX =";YMAX;"    "
3696  FOR X%=1 TO 10:M%=XMAX/20:FILL XMAX-3*M%,X%*M% XMAX,X%*M%+M%/2 RND(3)+21:NEXT
3697  FOR X%=0 TO XMAX STEP 3:DRAW 0,0 X%,3*YMAX/4 RND(3.0)+21.0:NEXT:GOSUB 850:RETURN
```

NEW

```
3700  A$="NEW":GOSUB 900:PRINT "Deletes current BASIC program and resets all variables":GOSUB 600:GO
TO 800
```

NOISE

```
3900  A$="NOISE":GOSUB 900:PRINT F$;TAB(25);"NOISE E V"
3915  PRINT TAB(25);"NOISE OFF":GOSUB 600:PRINT "Gives white noise on both channels"
3927  PRINT "following ENVELOPE E(0/1) with volume V":GOSUB 600:GOSUB 850
3930  PRINT CHR$(12):LIST 3935-3940
3935  ENVELOPE 0 15,5;0,15;
3937  NOISE 0 15:WAIT TIME 200
3940  NOISE OFF
3945  GOSUB 850:CURSOR 0,15:GOSUB 600:LIST 3955-3960
3955  ENVELOPE 1 15
3957  FOR Z%=1 TO 10:FOR X%=5 TO 15
3958  NOISE 1 X%:WAIT TIME 5
3960  NEXT:NEXT:NOISE OFF
3970  GOTO 800
```



ON GOSUB

```
4000  A$=" ON   GOSUB":GOSUB 900:PRINT F$;TAB(25);":   ON A GOSUB 100,150,200,250,....":GOSUB 600
4030  CURSOR 0,15:PRINT "If A=1 then GOSUB 100   ( GOTO 100)"
4035  PRINT "If A=2 then GOSUB 150":PRINT "If A=3 then GOSUB 200"
4040  PRINT "If A=4 then GOSUB 250":GOSUB 600:PRINT "If A<=0 or A>number of line numbers"
4060  PRINT "then the following statement is executed":GOTO 800
```

ON GOTO

```
4100  A$=" ON GOTO ":GOSUB 900:PRINT F$;TAB(25);"ON A GOTO 100,150,200,250"
4120  GOSUB 600:GOSUB 850:GOTO 4030
```

OUT

```
4200  A$="OUT":GOSUB 900:PRINT F$;TAB(25);"OUT A,B":GOSUB 600
4220  PRINT "Sends the number in variable B to the DCE-bus"
4230  PRINT "The 8255 IC is used following the DCE-concept":GOSUB 600:GOTO 800
```

POKE

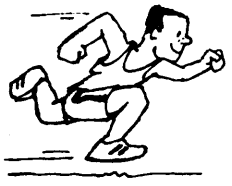
```
4300  A$="POKE":GOSUB 900:PRINT F$;TAB(25);"POKE #BFE0,#FF"
4315  PRINT TAB(25);"POKE A,B":PRINT TAB(25);"POKE 40000,216":GOSUB 600
4330  PRINT "Stores value B into memory location A":GOSUB 600:LIST 4350:GOSUB 850
4350  POKE #B935,65
4360  GOSUB 850:CURSOR 0,10:LIST 4370:GOSUB 850:CURSOR 0,8
4370  POKE #B932,#FF
4380  GOTO 800
```

PRINT

```
4400  A$="PRINT":GOSUB 900:PRINT F$;TAB(25);"PRINT A,B,C"
4420  PRINT TAB(25);"PRINT":PRINT TAB(25);"PRINT 'TOTAL '; T"
4440  PRINT TAB(25);"? TOTAL":GOSUB 600:LIST 4450-4470
4450  PRINT "4450":FOR X%=1 TO 5:PRINT X%:NEXT
4460  PRINT "4460":FOR X%=1 TO 5:PRINT X%;:NEXT:PRINT
4470  PRINT "4470":FOR X%=1 TO 5:PRINT X%,:NEXT
4480  GOTO 800
```

REM

```
4600  A$="REM":GOSUB 900:PRINT F$;TAB(25);"350 INPUT A$:REM ASK NAME of player":GOSUB 600
4610  PRINT "Allows comments inside BASIC programs"
4620  PRINT "REM statements are not executed":GOSUB 600:GOTO 800
```



RUN

```
4900  A$="RUN":GOSUB 900:PRINT F$;TAB(25);"RUN":PRINT TAB(25);"RUN 100":GOSUB 600
4930  PRINT "Starts execution of a BASIC program im memory":PRINT "All variables and arrays are clea
red":GOTO 800
```

SAVE

```
5000  A$="SAVE":GOSUB 900:PRINT F$;TAB(25);"SAVE"
5020  PRINT TAB(25);"SAVE 'OTHELLO by JUL KABAS'":PRINT TAB(25);"SAVE M$":GOSUB 600
5040  PRINT "Saves on cassette or disc the current program in memory"
5050  PRINT "You get the message : SET RECORD,START'TAPE,TYPE SPACE":GOTO 800
```

SAVEA

```
5100  A$="SAVEA":GOSUB 900:PRINT F$;TAB(25);"SAVEA A$ 'TUNE'"
5115  PRINT TAB(25);"10 SAVEA A$ 'TUNE'":PRINT TAB(25);"10 SAVEA A$":GOSUB 600
5130  PRINT "Saves an array on tape or disc":GOSUB 600:GOTO 800
```

SOUND FREQ

```
5200  A$="SOUND":GOSUB 900:PRINT :PRINT F$;TAB(11);" SOUND  1 0 5 0 FREQ(440)"
5230  FOR X%=18.0 TO 5.0 STEP -1.0:CURSOR 19,X%:PRINT CHR$(10);:NEXT
5235  PRINT "Oscillator number : 0 , 1 or 2"
5240  FOR X%=18.0 TO 8.0 STEP -1.0:CURSOR 21,X%:PRINT CHR$(10);:NEXT
5245  PRINT "Envelope number : 0 or 1"
5247  FOR X%=18.0 TO 11.0 STEP -1.0:CURSOR 23,X%:PRINT CHR$(10);:NEXT
5249  PRINT "Volume 0.....15"
5251  FOR X%=18.0 TO 14.0 STEP -1.0:CURSOR 25,X%:PRINT CHR$(10);:NEXT
5253  PRINT "Mode 0,1,2,3"
5255  FOR X%=18.0 TO 17.0 STEP -1.0:CURSOR 27,X%:PRINT CHR$(10);:NEXT
5257  PRINT "Frequency in Hertz":GOSUB 850:PRINT CHR$(12);
5259  A$="SOUND 2":GOSUB 900
5260  SOUND 1 0 15 0 FREQ(440,0)
5261  LIST 5260:GOSUB 5299
5262  SOUND 1 0 15 1 FREQ(440.0)
5264  LIST 5262:GOSUB 5299
5266  SOUND 1 0 15 2 FREQ(31.0):SOUND 1 0 15 2 FREQ(15000.0)
5268  LIST 5266:GOSUB 5299:GOTO 800
5299  WAIT TIME 100:SOUND OFF :RETURN
```

STOP

```
5300  A$="STOP":GOSUB 900:PRINT F$;TAB(25);"100 STOP":GOSUB 600
5320  PRINT "Basic suspends execution of the program":PRINT "and enters the command mode"
5340  PRINT "To continue type CONT":GOSUB 600:GOTO 800
```

TALK

```
5400  A$="TALK":GOSUB 900:PRINT F$;TAB(25);"TALK #400":GOSUB 600
5420  PRINT " By building up a table of parameters in memory"
5430  PRINT " and sending this information to the oscillators"
5440  PRINT " by the TALK command,":PRINT " It is possible to create vocal sounds":GOTO 800
```

TROFF TRON

```
5500  A$="TRON TROFF":GOSUB 900:PRINT F$;TAB(25);"10 TRON":PRINT TAB(25);"100 TROFF":GOSUB 600
5515  TRON
5520  PRINT "When BASIC encounters a TRON command,"
5530  PRINT "all lines in execution are displayed"
5540  PRINT "until a TROFF command is given"
5550  TROFF
5555  GOSUB 600:PRINT "TRACE WAS ON  for linenumbers 5520-5540":GOTO 800
```

WAIT

```
5700  A$="WAIT":GOSUB 900
5710  PRINT F$;TAB(25);"WAIT I,J,k (1)":PRINT TAB(25);"WAIT MEM I,J,k (2)"
5715  PRINT TAB(25);"WAIT TIME I (3)":PRINT TAB(25);"K is optional":GOSUB 600
5720  PRINT "(1) Read status of DCE port I,esclusive OR's with k"
5730  PRINT "    and AND's the result with J until result equals J":GOSUB 600
5740  PRINT "(2) As (1), but I is a memory loction"
5750  PRINT "    example : event check :WAIT MEM #FD00,#10":GOSUB 600
5760  PRINT "(3) Delays program execution for a time I"
5780  PRINT "    Time is in units of 20 milliseconds":GOSUB 600:GOTO 800
```

UT

```
5800  A$="UT":GOSUB 900:PRINT F$;TAB(25);"UT (command mode)":GOSUB 600
5820  PRINT "Calls the Machine Language Monitor":PRINT "with this facilities:";TAB(30);"L LOOK"
5840  PRINT TAB(30);"D DISPLAY":PRINT TAB(30);"G GO":PRINT TAB(30);"F FILL"
5850  PRINT TAB(30);"S SUBSITUTE":PRINT TAB(30);"M MOVE":PRINT TAB(30);"X EXAMINE"
5860  PRINT TAB(30);"V VECTOR EXAMINE":PRINT TAB(30);"R READ":PRINT TAB(30);"W WRITE"
5870  PRINT TAB(30);"Z INITIALISE":GOSUB 600:GOTO 800
```

ABS

```
5900  A$="ABS":GOSUB 900:PRINT F$;TAB(25);"A=ABS(B)":GOSUB 600:PRINT :PRINT
5920  FOR X%=1 TO 10:A%=RND(7657):IF RND(1)<0.5 THEN A%=A%*(-1)
5940  PRINT TAB(10);"A = ";A%;TAB(22);CHR$(10);TAB(25);" ABS(A) =";ABS(A%):NEXT:GOTO 800
```

ACOS ASIN ATN

```
6000  A$="ASIN ACOS ATN":GOSUB 900:PRINT F$;TAB(25);"A=ACOS(B) A=ASIN(B) A=ATN(B)":GOSUB 600
6020  PRINT "Returns arc sine,cosine,tangent of argument":PRINT "Result is between -PI/2 and PI/2":G
OSUB 600
6030  PRINT "argument";TAB(15);"arc sine";TAB(30);"arc cosine";TAB(45);"arc tangent":GOSUB 600
6040  FOR X%=1 TO 10:A=RND(1):IF RND(1)<0.5 THEN A=A*(-1)
6050  PRINT A;TAB(15);ASIN(A);TAB(30);ACOS(A);TAB(45);ATN(A):NEXT:GOTO 800
```

ALOG

```
6100  A$="ALOG":GOSUB 900:PRINT F$;TAB(25);"A=ALOG(B)":GOSUB 600
6120  PRINT "Returns antilog base 10 of argument":GOSUB 600
6130  PRINT :PRINT :FOR X%=1 TO 10:A=RND(7)
6140  PRINT TAB(3);"argument = ";A;TAB(25);CHR$(10);TAB(28);"antilog = ";ALOG(A):NEXT:GOTO 800
```

ASC

```
6200  A$="ASC":GOSUB 900:PRINT F$;TAB(25);"A = ASC(X$)":GOSUB 600
6220  PRINT "Returns the integer ASCII value of the first character"
6230  PRINT "of the string X":GOSUB 600
6235  FOR X%=0 TO 3:PRINT TAB(X%*15);"str";TAB(X%*15+5);"asc";:NEXT:PRINT :GOSUB 600
6240  FOR X%=48 TO 91 STEP 4:FOR Y%=0 TO 3
6260  PRINT TAB(Y%*15);CHR$(34);CHR$(X%+Y%);CHR$(34);TAB(Y%*15+5);X%+Y%;
6270  NEXT:PRINT :NEXT:GOTO 800
```

CHR$

```
6500  A$="CHR$":GOSUB 900:PRINT F$;TAB(25);"A$=CHR$(B)":GOSUB 600
6520  PRINT "Returns a character whose value is B":GOSUB 600:GOTO 6235
```

COS SIN TAN

```
6600  A$="SIN COS TAN ":GOSUB 900:PRINT F$;TAB(25);"A = COS(B) A = SIN(B) A=TAN(B)":GOSUB 600
6620  PRINT "Returns the (co)sine ,tangent of the argument":GOSUB 600
6630  PRINT :PRINT "argument";TAB(15);"sine";TAB(30);"cosine";TAB(45);"tangent":GOSUB 600
6635  FOR X%=1 TO 10:A=RND(2*PI):IF RND(1)<0.5 THEN A=A*(-1)
6640  PRINT A;TAB(15);SIN(A);TAB(30);COS(A);TAB(45);TAN(A):NEXT:GOTO 800
```

CURX

```
6700  A$="CURX":GOSUB 900:PRINT F$;TAB(25);"A=CURX":GOSUB 600
6720  PRINT "Sets A to the X position of the CURSOR":GOSUB 600:GOSUB 850
6730  FOR X%=0 TO 59:CURSOR X%,10:PRINT CHR$(255):CURSOR 25,5:PRINT "CURX = ";X%:SOUND 1 0 15 0 FREQ
(31+X%*10)
6740  WAIT TIME 2:SOUND OFF :WAIT TIME 10:CURSOR X%,10:PRINT CHR$(32):NEXT:GOTO 800
```

CURY

```
6800  A$="CURY":GOSUB 900:PRINT F$;TAB(25);"A = CURY":GOSUB 600
6820  PRINT "Sets A to the Y position of the cursor":GOSUB 600:GOSUB 850
6830  PRINT CHR$(12);:FOR X%=1 TO 23:CURSOR 5,X%:PRINT CHR$(255)
6840  SOUND 1 0 15 0 FREQ(800):WAIT TIME 2:SOUND OFF
6850  CURSOR 40,10:PRINT " CURY = ";X%:WAIT TIME 10:CURSOR 5,X%:PRINT CHR$(32):NEXT:GOTO 800
```

EXP

```
6900  A$="EXP":GOSUB 900:PRINT F$;TAB(25);"A = EXP(X)":GOSUB 600
6920  PRINT "Returns the value 'e' (2.71828) to the power X":GOSUB 600
6930  FOR X%=1 TO 10:A=RND(20):IF RND(1)<0.5 THEN A1%=A*(-1)
6940  PRINT TAB(5);"   X = ";A;TAB(25);CHR$(10);TAB(27);"  e ^ X = ";EXP(A):NEXT:GOTO 800
```

FRAC

```
7000  A$="FRAC":GOSUB 900:PRINT F$;TAB(25);"A = FRAC(X)":GOSUB 600
7020  PRINT "Returns the flaotimg point fractional part of the argument":GOSUB 600
7030  PRINT :PRINT :FOR X%=1 TO 10:A=RND(10000):IF RND(2)<1 THEN A=A*(-1)
7040  PRINT TAB(2);" argument = ";A;TAB(25);CHR$(10);TAB(27);" fraction = ";FRAC(A):NEXT:GOTO 800
```

FRE

```
7100  A$="FRE":GOSUB 900:PRINT F$;TAB(25);"B = FRE":GOSUB 600
7210  PRINT " Gives the number of bytes currently unused by BASIC":GOSUB 600:GOTO 800
```

GETC

```
7300  A$="GETC":GOSUB 900:PRINT F$;TAB(25);"G=GETC":GOSUB 600
7320  PRINT "G is set to the ASCII value of the character typed":PRINT "on the keyboard":GOSUB 600
7330  PRINT :PRINT :PRINT "Hit some keys....":PRINT :FOR X%=1 TO 10
7334  G%=GETC:IF G%=0 THEN 7334
7340  PRINT TAB(5);" you typed ";CHR$(G%);" ASCII value = ";G%:NEXT:GOTO 800
```

HEX$

```
7400  A$="HEX$":GOSUB 900:PRINT F$;TAB(25);"A$=HEX$(X)":GOSUB 600
7420  PRINT "Returns a string of characters representing ":PRINT "the hexadecimal value of the numbe
r X":GOSUB 600
7430  PRINT :PRINT :FOR X%=1 TO 10:A%=RND(65000)
7440  PRINT TAB(5);" value = ";A%;TAB(25);CHR$(10);TAB(27);" HEX$ = #";HEX$(A%):NEXT:GOTO 800
```

INP

```
7500  A$="INP":GOSUB 900:PRINT F$;TAB(25);"A= INP(31)":GOSUB 600
7520  PRINT "Reads the byte present in the DCE-bus CARD 3 PORT 1":PRINT "and assigns it to the varia
ble A":GOSUB 600:GOTO 800
```

INT

```
7600  A$="INT":GOSUB 900:PRINT F$;TAB(25);"A=INT(B)":GOSUB 600
7620  PRINT "Returns the largest integral value less than":PRINT "or equal to the argument B":GOSUB
600
7630  FOR X%=1 TO 10:B=RND(1000)
7640  PRINT TAB(5);"value = ";B:TAB(25):CHR$(10);TAB(27);" integer = ";INT(B):NEXT:GOTO 800
```

LEFT$ MID$ RIGHT$ LEN

```
7700  A$="LEFT$ MID$ RIGHT$ LEN":GOSUB 900
7710  PRINT F$;TAB(25);"B$=LEFT$(A$,3)":PRINT TAB(25);"B$=MID$(A$,3,2)"
7720  PRINT TAB(25);"B$=RIGHT$(A$,4)":PRINT TAB(25);"B=LEN(A$)":GOSUB 600
7730  PRINT "Returns a number of characters from B$":GOSUB 600:A$="SOUNDGENERATOR"
7750  CURSOR 0,13:PRINT "LEN(A$)=14":CURSOR 40,13:PRINT A$
7755  CURSOR 20,12:PRINT "LEFT$(A$,5) = ";TAB(40);:FOR X%=1 TO 5:PRINT CHR$(94);:NEXT
7757  CURSOR 40,11:PRINT LEFT$(A$,5):CURSOR 40,9:PRINT A$
7765  CURSOR 20,8:PRINT "MID$(A$,3,6) = ";TAB(43);:FOR X%=1 TO 6:PRINT CHR$(94);:NEXT
7766  CURSOR 43,7:PRINT MID$(A$,3,6):CURSOR 40,5:PRINT A$
7775  CURSOR 20,4:PRINT "RIGHT$(A$,4) = ";TAB(50);:FOR X%=1 TO 4:PRINT CHR$(94);:NEXT
7777  CURSOR 50,3:PRINT RIGHT$(A$,4):GOTO 800
```

LOG LOGT

```
7900  A$="LOG LOGT":GOSUB 900:PRINT F$;TAB(25);"A=LOG(X)":PRINT TAB(25);"A=LOGT(X)":GOSUB 600
7930  PRINT " LOG returns the natural logarithm of the argument":PRINT " LOGT calculates the logarit
hm base 10":GOSUB 600
7950  FOR X%=1 TO 10:A%=RND(100000):PRINT TAB(1);" argument = ";A%;TAB(20);" LOG = ";LOG(A%);TAB(40)
;" LOGT = ";LOGT(A%):NEXT:GOTO 800
```

PDL

```
8200  A$="PDL":GOSUB 900:PRINT F$;TAB(25);"A=PDL(2)":GOSUB 600
8210  PRINT "Sets A to a number between 0 and 255,":PRINT "representing the position of potm. nr 2":
GOSUB 600:GOSUB 850
8230  MODE 6A:PRINT CHR$(12);:PRINT "you controll some of these lines..."
8235  FOR Y%=0 TO 5:FOR X%=1 TO 50
8240  A%=PDL(Y%):CURSOR Y%*10,2:PRINT "PDL";Y%;"=";A%:DRAW Y%*50,0 Y%*50,A% 23:WAIT TIME 5:DRAW Y%*5
0,0 Y%*50,A% 20
8250  NEXT:NEXT:GOTO 800
```

PEEK

```
8300  A$="PEEK":GOSUB 900:PRINT F$;TAB(25);"A=PEEK(B)":GOSUB 600
8310  PRINT "Sets A equal to the contents of adress B":GOSUB 600:GOTO 800
```

PI

```
8400  A$="PI":GOSUB 900:PRINT "Returns the floating point value 3.14159":GOTO 800
```

RND

```
8600  A$="RND":GOSUB 900:PRINT F$;TAB(25);"A=RND(100)":PRINT TAB(25);"A=RND(1)*100":PRINT TAB(25);"A
=RND(-1)":PRINT TAB(25);"A=RND(0)":GOSUB 600
8610  PRINT "Generates a software or hardware (RND(0)) random number":PRINT :FOR X%=1 TO 10:PRINT TA
B(10);"RND(100)=";TAB(20);RND(100):NEXT:GOTO 800
```

SCRN

```
8700  A$="SCRN":GOSUB 900:PRINT F$;TAB(25);"A=SCRN(X,Y)":GOSUB 600
8710  PRINT "Sets A equal to the colour number of the screen":PRINT "at coordinate X,Y":GOSUB 600
8720  GOSUB 850:MODE 1A:PRINT CHR$(12):FOR X%=0 TO 52 STEP 2:DOT 30,X% RND(15):NEXT
8730  FOR X%=0 TO 52 STEP 2:A%=SCRN(30,X%):DRAW 0,X% 25,X% A%:DRAW 35,X% XMAX,X% A%
8740  CURSOR 9,2:PRINT "SCRN (";30;",";X%;")= ";A%;"    ":WAIT TIME 50:NEXT:GOTO 800
```

SGN

```
8800  A$="SGN":GOSUB 900:PRINT F$;TAB(25);"A=SGN(X)":GOSUB 600
8810  PRINT "Sets A to 1 if X>0 , 0 if X=0 , -1 if X<0":GOSUB 600:GOTO 800
```

SPC

```
9000  A$="SPC":GOSUB 900:PRINT F$;TAB(25);"PRINT SPC(10);...":GOSUB 600
9010  PRINT "Returns a string of a number of spaces":GOSUB 600
9020  PRINT :LIST 9030:PRINT
9030  FOR X%=1 TO 6:PRINT TAB(RND(10));CHR$(10);:R%=RND(40):PRINT SPC(R%);CHR$(10);"SPC(";R%;")":PRI
NT :NEXT:GOTO 800
```

SQR

```
9100  A$="SQR":GOSUB 900:PRINT F$;TAB(25);"A=SQR(B)":GOSUB 600
9110  PRINT "Gives the square root of the argument":GOSUB 600
9120  PRINT TAB(20);"argument";TAB(40);"square root":GOSUB 600
9130  FOR X%=1 TO 10:B%=RND(100000):B=SQR(B%):PRINT TAB(20);B%;TAB(40);B:NEXT:GOTO 800
```

STR$ VAL

```
9200  A$="STR$ VAL":GOSUB 900:PRINT F$;TAB(25);"A$=STR$(X)":PRINT TAB(25);"A=VAL(A$)":GOSUB 600
9210  PRINT "STR$ returns a string which is the ASCII representation of x"
9220  PRINT "VAL returns the fpt value of the number in the string":GOSUB 600:GOTO 800
```

TAB

```
9300  A$="TAB":GOSUB 900:PRINT F$;TAB(25);"PRINT TAB(30);...":GOSUB 600
9310  PRINT "Moves the cursor to the desired column":GOSUB 600
9320  FOR Y%=15 TO 3 STEP -1:A%=RND(40):CURSOR A%,Y%:PRINT CHR$(255);"TAB(";A%;")":WAIT TIME 50:NEXT
:GOTO 800
```

VARPTR

```
9600  A$="VARPTR":GOSUB 900:PRINT F$;TAB(25);" A=VARPTR(B)":PRINT TAB(25);"A=VARPTR(B(3,4))":GOSUB 6
00
9610  PRINT "Variable A is set to the memory address of B":GOSUB 600:GOTO 800
```

MOD

```
9900  A$="MOD":GOSUB 900:PRINT F$;TAB(25);"A = X MOD B":GOSUB 600
9910  PRINT "A equals the remaining part of X/B":GOSUB 600
9920  PRINT :FOR X%=1 TO 10:A%=RND(100):B%=RND(6)+1:PRINT TAB(20);A%;" MOD";B%;" =";A% MOD B%:NEXT:G
OTO 800
```

EXPERIENCES WITH DAI VIDITEL

At the Utrecht meeting of HCC I bought the DAI VIDITEL program as developed by Hans de Vries c.s. and requested for a Viditel subscription from PTT.

In the weeks that followed I had to wait for my Viditel modem to be delivered by PTT as a part of the Viditel subscription. But the DAI VIDITEL program could be used already. In its local editing mode viditel-like screen lay-outs can be composed. This offers an easy to handle alternative for the preparation of text frames to be copied onto colour slides !

The modem came within three weeks. In that per iod I also assembled the connection cable, so the Viditel computer could be called immediately.

It is a pleasure for me to report a smoothly operating DAI VIDITEL program. By the fact that the complete DAI keyboard is used the potentials of Viditel are much greater for DAI users than for people using a special Viditel tv-set. By the so called Vidibus function I could sent a letter with my first impressions to the designers of DAI Viditel. Note, that this way of corresponding is cheaper than sending real mail !

Another facility made possible through DAI VIDITEL and the PTT modem is a connection with the Viewdata computer of NOVA Automation Conultants. This private viewdata system is accessible for HCC members (free of charge). They can both access it for retrieving stored information pages, but also for storing their own information pages onto it. Further, it offers program development and processing functions which ar e available in combination with the viewdata functions.

According to my 'software philosophy' programs should be seen as 'tools': simple products which perform a single well-defined task which results into output that can be processed further by another tool. Following this philosophy I would like to offer the following suggestions to the designers of DAI VIDITEL:

1. do NOT add any more bells and whistles to the program.
2. change the program into a module; i.e. make it posible to call the program as a module from within another module or basic program.

It is my strong opinion that by following these suggestions the designers of DAI VIDITEL will be able to provide the DAI users with a powerful tool that they can extend themselves according to a variety of wishes. This can lead to a stream of ideas and applications for which the DAI VIDITEL module is crucial. It frees the designers from the implementation and management of a never ending stream of 'improvements'. The latter in fact better can be programmed by other people, thus enabling the designers to devote themselves to more advanced problems.

Supplement to DAI VIDITEL documentation:

1. The english pound sign (ASCII 35) can be entered by SHIFT-TAB; the graphics symbol corresponding with this sign also can be entered by SHIFT-TAB.
2. The at-sign (ASCII 64) can be entered by SHIFT-0 (zero).
3. The right arrow and left arrow (ASCII 93 and 91) are entered by 1

and [respectively.
4. The release graphics sign (ASCII 95) is entered by CTRL-SHIFT-CHARDEL.
5. Other peculiarities are:

Sign:	ASCII:	How to enter:
1/2	92	TAB
US	96	CTRL-SPACE
1/4	123	CTRL-;
11	124	CTRL-<
3/4	125	CTRL-=

Recommended literature:
J.J.M.Bokland. Viditel: techniek voor de abonneeaparatuur; The Hague, 1981.

P. van der Hijden

Benodigde Viditel-apparatuur

1. Een aangepast TV-toestel met afstandsbediening, bureauterminal of hobby-computer.
2. Een telefoontoestel.
3. Een modem (dat door de PTT wordt verstrekt zonder extra kosten).

Viditel-randapparatuur

De PTT verhuurt Viditel-randapparatuur, waaronder:
- o de Viditel-bureauterminal (VBT-003), bestaande uit een combinatie van kleurenmonitor en alfa-numeriek toetsenpaneel;
- o de Viditel-printer (VPE-403), voor het maken van afdrukken van Viditel-beelden op papier;
- o de Viditel-printertussenschakeleenheid (VPI-403), om in de handel zijnde matrix-printers aan te sluiten op de Viditel-abonnee-modem;
- o de Viditel-automatische nummerzender (VCG-901), voor o.m. het automatisch uitzenden van het toegangsnummer, code en eventuele privé-code;
- o de Viditel-invoerterminal (VIT-003).

Viditel in de toekomst!

Tot nu toe kunt u met Viditel al diverse kanten uit. U kunt informatie opvragen die voor u belangrijk is. Daarnaast is het mogelijk om via antwoordbeelden bestellingen te plaatsen bij bedrijven en berichten te versturen aan andere Viditel-abonnees.

Begin 1982 worden de mogelijkheden met Viditel verder uitgebreid door de zgn. Gateway-functie. Een abonnee kan dan met deze functie via Viditel gekoppeld worden aan (externe) computers van bedrijven en instellingen, waardoor hij over steeds meer en betere informatie- en communicatiemogelijkheden kan beschikken, zoals: het boeken van reizen -met ontvangst van reserveringsbevestiging - , het overboeken van geldbedragen op rekeningen, het laten uitvoeren van berekeningen, het bestellen van produkten enz. enz.

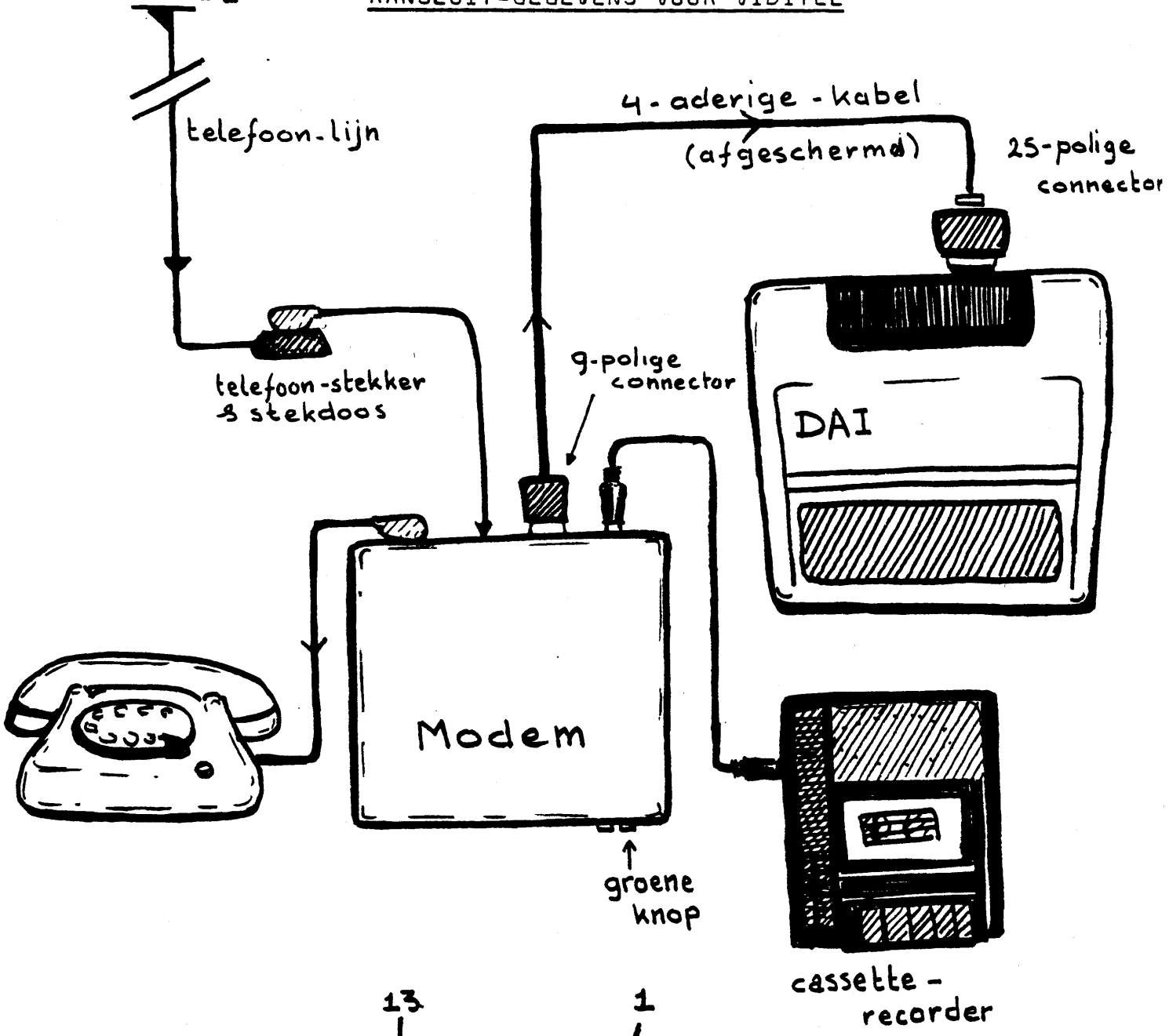
Wilt u meer weten over Viditel, vraag dan informatie bij:

Centrale Directie der PTT
Directoraat Commerciële Zaken Telecommunicatie
Bureel Viditel
Antwoordnummer 6000
2500 VB 's-Gravenhage
Een telefoontje kan natuurlijk ook:
(070) 75 32 69 / 75 33 92 / 75 40 74

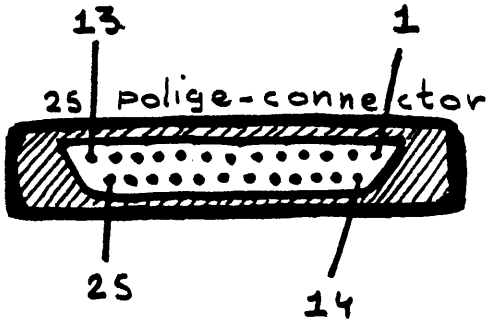
Wanneer men Viditel-abbonee wordt krijgt men automatisch het modem te huur à f10,- per maand, de 4-aderige kabel wordt NIET door de PTT geleverd.

VIDITEL

AANSLUIT-GEGEVENS VOOR VIDITEL

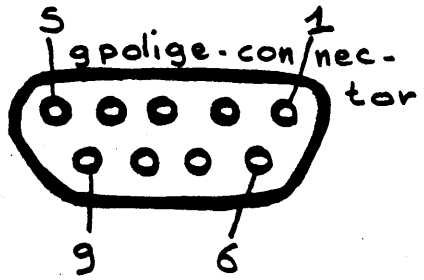


DAI:



Ampenol:
17-90250-16 &
17-1372

MODEM:



Amphenol:
17-90090-16 &
17-1370
leverancier RODELCO

AANSLUIT-GEGEVENS	MODEM	DAI
aarde	5	1
serie in	4	3
serie out	3	2
DTR	7	8

(DTR=Data Terminal Ready)

DAI-VIDITEL kent alle standaard VIDITEL-attributen.

Zoals bv.

: 8 kleuren.

: Grafische symbolen & 'gescheiden' grafische symbolen.

: Dubbele hoogte karakters.

: Flashing.

enz.

Er is geen enkele hardware-uitbreiding of aanpassing in de DAI nodig.

De beeld-opbouw geschied met behulp van een 16 kleuren grafische mode. (gelijk aan mode 5)

Het programma is geheel in machine-code geschreven en beslaat ongeveer $3\frac{1}{2}$K geheugenruimte.

Het toetsen-bord is zodanig aangepast dat het mogenlijk is de DAI te gebruiken voor het editen van viditel-pagina's.

Uiteraard kan men het toetsen-bord ook gebruiken voor het verzenden van 'electronische brieven' via de Viditel-VIDIBUS service, of het verzenden van 'Antwoord-pagina's aan Viditel informatie-leveranciers.

==========

Edit-mogenlijkheden:

Break: full duplex, Break Break: half duplex enz.

Ctrl Q : cursor on
Ctrl T : cursor off } volledige cursorbesturing
Ctrl L : clear
Ctrl ↑ : home

Ctrl [: ESCAPE

ESCAPE-functies

ESC A :	alfa-numeriek rood	;ESC Q :	grafisch rood
ESC B :	alfa-numeriek groen	;ESC R :	grafisch groen
ESC C :	alfa-numeriek geel	;ESC S :	grafisch geel
ESC D :	alfa-numeriek blauw	;ESC T :	grafisch blauw
ESC E :	alfa-numeriek magenta	;ESC U :	grafisch magenta
ESC F :	alfa-numeriek cyan	;ESC V :	grafisch cyan
ESC G :	alfa-numeriek wit	;ESC W :	grafisch wit
ESC H :	flash	;ESC Y :	continue grafisch
ESC I :	steady	;ESC Z :	gescheiden grafisch
ESC J :	End	;ESC L :	normale hoogte
ESC K :	Start	;ESC M :	dubbele hoogte
ESC] :	nieuwe achtergrond	;ESC ↑ :	grafisch 'houdend'
ESC TAB:	zwarte achtergrond	;ESC shift TAB:	release

Voor off-line editen (het maken van beelden zonder dat viditel is aangesloten):

1 toets de 'BREAK'toets in.
2 druk op de 'CTRL'toets en toets tegelijk de 'Q' in. (cursor on).

Enkele belangrijke geheugen-adressen :
301, : Geheugen-grootte : #C0 bij 48K en #80 bij 32K
303, : ASCII-pagina : #54 bij 48K en #14 bij 32K
304 t/m # 308, : Gebruikte kleuren voor achtereenvolgens: zwart, rood, groen, geel, blauw, magenta, cyaan en wit

==========

HANDLEIDING VOOR HET HELE GEZIN: HOE KRIJG IK VERBINDING MET VIDITEL?

stap 1. Is de DAI met de telefoon verbonden? (via de modem)
Zo nee: Maak de verbinding door de 25 polige 'stekker' vanuit de modem in de 'RS232 bus' van de DAI te steken, deze bevind zich aan de achterzijde.

stap 2. Zoek de Viditel-cassette en stop deze in de cassette-recorder.

stap 3. Zet de TV aan.

stap 4. Zet de DAI aan, de schakelaar bevind zich aan de rechter-achterzijde.

stap 5. U krijgt nu een groen (grijs) beeld met in grote letters: DAI PERSONAL COMPUTER. : druk een willekeurige toets in.

stap 6. Er is 'BASIC' op het scherm verschenen.
U geeft nu de computer z'n eerste instructie:
Typ UT in en druk op de 'RETURN'toets.

stap 7. Er is nu 'UTILITY' op het scherm verschenen;
geef de tweede instructie:
Typ Z3 in en druk op de 'RETURN'toets.

stap 8. Het moment is nu gekomen dat U het Viditel-programma vanaf de cassette in de DAI kunt gaan lezen:
1. Typ R in en druk op de 'RETURN'toets.
2. Start de cassette-recorder.

stap 9. Als het programma ingelezen is verschijnt er weer een '>'.
Het DAI-VIDITEL programma kan nu in werking worden gesteld: Typ G400 en druk de 'RETURN'toets in.

stap 10. Zo, het programma loopt, het beeld wordt zwart.
Bel nu naar VIDITEL:
020-318318 voor AMSTERDAM of
070-151515 voor DEN HAAG.

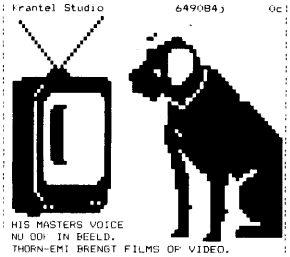
stap 11. Als u de hoge piep-toon hoort, druk dan op de groene knop aan de voorzijde van het modem, en leg de hoorn weer op de haak.

stap 12. VIDITEL presenteerd zich.
Typ uw toegangsnr. in : .....
Typ uw code-nummer in : .....
En typ eventueel uw prive-code-nummer in : .....

U bent nu met VIDITEL verbonden. Veel genoegen!!

***VIDITEL is standard in Belgium,Holland,UK & Germany.
***latest version has extra features for printer and cassette/DCR

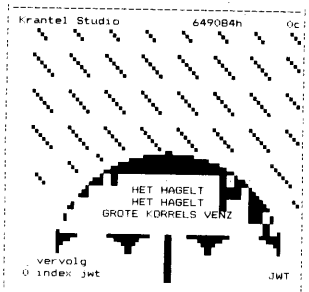
Krantel Studio 649084j 0c



HIS MASTERS VOICE
NU OOK IN BEELD.
THORN-EMI BRENGT FILMS OP VIDEO.
vervolg JWT

9 voor keuzepagina

Krantel Studio 649084h 0c



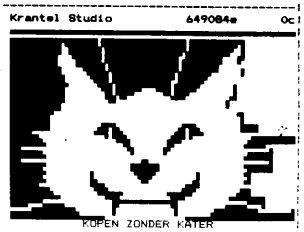
_ vervolg
0 index jwt JWT

0 index jwt

Krantel Studio 649084c 0c

ONZE NAAM KLEEFT OOK
AAN HET NIETIGSTE DRUKWERK.

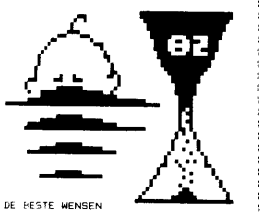
Krantel Studio 649084e 0c



VERVAARDIGD VOOR POSTBUS 51
CONSUMENTENINFORMATIE JWT

Viditel 180139a 0c

en 1982 zal ook wel lukken...



DE BESTE WENSEN

Krantel Studio 64908a 0c
J. WALTER THOMPSON COMPANY

_ voor kennismaking 1 index jwt

Pag 000 SYSTEM 005 min

ABC DATAVISION

***** HOOFDINDEX *****
1. WAT IS ABC-DATAVISION ??
■. ALGEMENE INDEX.
3. INFORMATIE ABC-Data B.V. (NIEUW!)
■. PRODUKT INFORMATIE.
5. GEBRUIKSAANWIJZING.
■. LIJST VAN ONDERWERPEN.
7. GRAFISCHE VOORBEELDEN.
8. HOBBY AANBIEDINGEN
9. EDITING, Alleen voor I.P.'s

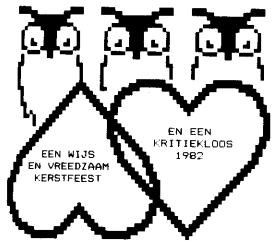
Toets het nummer van Uw keuze.

VIEWDATA 3000

Thank you for calling.

Please call again.

Viditel 180130a 0c
CEES P. RUIGROK'S UILEN WENSEN U
NE 400



Krantel Studio 649084b 0c

_ vervolg
0 index jwt JWT

Krantel Studio 649084m 0c

FORD, MEER WAARD OP DE WEG.

_ vervolg
0 index jwt JWT

Philips' Natuurkundig Laboratorium

Viewdata

Alleen voor besloten gebruikersgroepen
--- Voor inlichtingen: 040-743072 ---

Mag ik uw gebruikersnummer?

Vries Data 301065a 0c

DAI-NAMIC

DAI PERSONAL COMPUTER CLUB

CONTACT-ADRES: W.F.HERMANS
HEIDE 98
WESTMEERBEEK
BELGIE

toets....

Vries Data 301065b 0c

DAI-NAMIC

NIET OFFICIEEL TOETS

1. DAINAMIC COLOFON
2. TIPS DE DAINAMIC PERIODIEK
3. VOOR DAINAMIC PROGRAMMA'S
4. AGENDA & ROUTEBESCHRIJVING

Vries Data 30106a 27c

viditel

HCC telecommunicatie

VIDITEL OP DE DAI PC

Machine code programma start op 4400

Volledige Handleiding komt in NOVA en in DAI-NAMIC

Vries Data 3010651a 0c

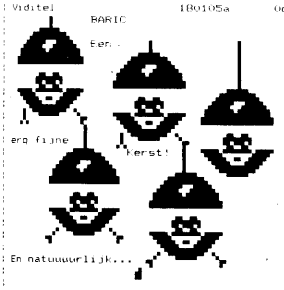
DAI-NAMIC

COLOFON

1. DAInamic = periodiek
2. Lid worden van DAInamic
3. Tel. contact & Correspondentie

Viditel 180105a 0c

BARIC



Een
erg fijne Kerst!
En natuuuurlijk...

Vries Data 3010600a 0c

***** H O O F D - M E N U *****

1 TELE-SOFTWARE (nog toekomstmuziek)
2 AGENDA
3 ACTUALITEITEN
4 GRAFISCHE POGINGEN
5 DAINAMIC
■ RESERVE
■ RESERVE
8 NIEUWTJES ALLERLEI
9 INFO VOOR DAI-GEBRUIKERS

VIDITEL NU OOK OP DAI

met dank aan J&JRG DE VRIES..
originele screen copies op
MX-80,gereduceerd formaat.
Laat uw abonneenummer kennen
als je er ook bij bent ...

Vries Data 3010691a 0c
DAI USER'S OP VIEWDATA/3000 Pag.1

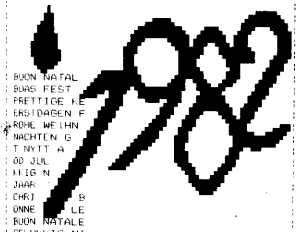
5506 J&JRG DE VRIES (IL):30106
BUNCHESTR.26
2286 GL RIJSWIJK (Z-H)
TEL. 070-931607
VID. 400003419

5525 R.DEN BREEMS
HROONSTADREEF 27
3067 RT ROTTERDAM
TEL. 010-551234
VID. 400004352

5539 SJEF VAN BERLO
G.GEZELLELAAN 5
2624 KT DELFT
TEL. 015-565440
VID. 222222222 Verv._

Zend Antwoordpagina: toets 9

Viditel 180107a 0c
DENI SPORT WENST U EEN 'SPEELS'



BUON NATAL
BONS FEST
PRETTIGE K
GEBTDAGEN F
FROHE WEIHN
NACHTEN G
T NYTT A
OD JUL
NIG N
JAAR
CHRI B
ONNE LE
BUON NATALE
GELUKKIG NI
EUWJAAR BUO

Pag 090 SYSTEM 006 min

ABC DATAVISION

HARTELIJK DANK VOOR UW INTERESSE

EN GRAAG TOT EEN VOLGENDE KEER

Krantel Studio 649084d 0c

FEMINA,
'N ZACHTHEID
WAAROP JE ALTIJD
MAG VERTROUWEN.

_ vervolg
0 index jwt JWT

Vries Data 3010691b 0c
DAI USER'S OP VIEWDATA/3000 Pag.2

5539 THEO VERBERKT (IL):3015539
v.BUERENSTRAAT 13
OUD HEUSDEN
TEL. 04162-2667
VID. 222222222

5548 PIETER v/d HEIDEN
VOORDORP 53
2352 BT LEIDERDORP
TEL. 071-411769
VID. 400004581

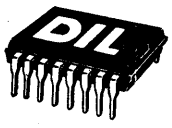
5522

Verv._

p.v.b.a. A.C.S.

Meensesteenweg 49
8800 ROESELARE
Tel. 051/21 30 89

Beenhouwersstraat 87
8000 BRUGGE
050/33 08 01



D.I.L.-ELEKTRONIKA,

MIJNSHERENLAAN 108,
3081 CH ROTTERDAM
Nederland
TELEFOON: 010 - 854213

Guibernau Electronica, s.a.

Sepulveda 104
BARCELONA-15 (SPAIN)
Tel. 243-34-32

IDS 2000

Rue de la Bonne Femme 11
4030 GRIVEGNEE
Tel. 041/41 32 20

LEGOTRONICS

Kon. Albert I laan 97
8800 ROESELARE
Tel. 051/22 01 03

MEMOCOM

Mini-digitale cassetterecorder

Postbus 2924
3000 CX ROTTERDAM - Nederland
Tel. 010-148284

MICRO SELECT

Toutes applications micro-électroniques
Vente de systèmes et composants micro-processeur

3, rue Delcloche
4020 LIÈGE
Tel. 041/41 28 10

Bennenbergweg 1
3221 NIEUWRODE (bij AARSCHOT)
Tel. 016/56 87 70
DAI - Epson Printers - Memocom digitale cassette recorder - Barco kleurenmonitor - Software - Microlectuut - Service

Publishing House J. VAN IN
att. : L. CAMPS

Educational Software
primary - secondary schools

Grote Markt 39
2500 LIER
Tel. 031/80 55 11

TEVETRONIC

Avenue Milcamps 57
1040 BRUSSEL
Tel. 02/736 61 24