```
                +
                ;
C003                    ORG     0C003H
                ;
C003            XMINIT: DS      3       ; PACKAGE INIT
                ;
C006            XFINM:  DS      3       ; INCR FPT NUMBER IN MEM
C009            XFDCM:  DS      3       ; DECR FPT NUMBER IN MEM
                ;
C00C            XFCOMP: DS      3       ; FLOATING POINT COMPARE
                ;
C00F            XIINM:  DS      3       ; INCR INT NUMBER IN MEM
C012            XIDCM:  DS      3       ; DECR INT NUMBER IN MEM
                ;
C015            XICOMP: DS      3       ; INTEGER COMPARE
                ;
C018            XPUSH:  DS      3       ; SAVE FPAC ON STACK
C01B            XPOP:   DS      3       ; RETRIEVE FPAC FROM STACK
                ;
                ; IO FUNCTIONS
                ;
C01E            XFCB:   DS      3       ; INPUT A FPT NUMBER TO FPAC
C021            XFBC:   DS      3       ; CONVERT A FPT NUMBER FOR OUTPUT
C024            XICB:   DS      3       ; INPUT INTEGER NUMBER TO IAC
C027            XIBC:   DS      3       ; CONVERT INTEGER FOR OUTPUT
C02A            XHCB:   DS      3       ; INPUT HEX NUMBER TO IAC
C02D            XHBC:   DS      3       ; CONVERT IAC TO HEX FOR OUTPUT
C030            XPRTY:  DS      3       ; PRETTIES UP FPT OR INTEGER NUMB
                ;
C033            DECBUF: DS      2       ; LOCATION OF OUTPUT BUFFER
                ;
                +       PAGE
```



```
                +
                ;
                ; MEMORY + IO MAP
                ;
                ; DEFINES WHERE TO FIND THE HARDWARE
                ;
FB00            MTHAD   EQU     0FB00H  ; MATH CHIP (IF FITTED)
                ;
FC00            SNDAD   EQU     0FC00H  ; 8253 ADDRESS (IF FITTED)
                ;
FC00                    SND0    EQU     SNDAD   ; CHAN 0
FC02                    SND1    EQU     SNDAD+2 ; CHAN 1
FC04                    SND2    EQU     SNDAD+4 ; CHAN 2
FC06                    SNDC    EQU     SNDAD+6 ; CONTROL
FC00                    PDLCH   EQU     SND0    ; PADDLE READING CHANNEL
                ;
                        ; 8253 MODE BYTES
                ;
0032                    COM1    EQU     032H    ; CHAN 0,MODE 1,2 BYTE OPERA
                ;
0036                    COM3    EQU     036H    ; CHAN 0, MODE 3, 2 BYTE
0076                    C1M3    EQU     076H
00B6                    C2M3    EQU     0B6H
                ;
0030                    COM0    EQU     030H    ; CHAN 0, MODE 0, 2 BYTE OP
                ;
0000                    COFIX   EQU     0       ; FIX COUNT ON CHANNEL 0
                ;
FD00            PORI    EQU     0FD00H  ; INPUT PORT
                ;
0004                    PIPGE   EQU     04H     ; PAGE SIGNAL
                ;
0008                    PIDTR   EQU     08H     ; SERIAL OP READY
                ;
0010                    PIBU1   EQU     10H     ; BUTTON ON PADDLE 1
                ;
0020                    PIBU2   EQU     20H     ; BUTTON ON PADDLE 2
                ;
0040                    PIRPI   EQU     40H     ; RANDOM BITS
                ;
0080                    PICAI   EQU     80H     ; CASSETTE INPUT DATA
                ;
FD01            PDLST   EQU     0FD01H  ; PADDLE SAMPLING START
                ;
FD04            PORO    EQU     0FD04H  ; VOLUME OUTPUTS CHANS 0,1
                ;
FD05            POR1    EQU     PORO+1  ; VOLUMES CHAN 2 AND NOISE
                ;
```

```
FD06            PORO    EQU     OFD06H  ; OUTPUT PORT
                ;
0001                    POCAS   EQU     01H     ; CASSETTE OUTPUT BIT
0007                    PDLMSK  EQU     7       ; PADDLE SELECT BITS
                ;
0008                    POPNA   EQU     08H     ; PADDLE ENABLE BIT
                ;
0010                    POCM1   EQU     10H     ; CASSETTE MOTOR CONTROL 1
0020                    POCM2   EQU     20H     ;       "       "       " 2
                ;
                        ; TOP 2 BITS ARE BANK SWITCHING
                ;
FE00            GIC     EQU     OFE00H  ; RWBUS GIC ADDRESS
                ;
0080                    RWMOP   EQU     080H    ; RW OUTPUT MODE
                ;
0090                    RWMIP   EQU     090H    ; RW INPUT MODE
                ;
FFF0            TICC    EQU     OFFF0H  ; TICC ADDRESS
                ;
F900            STTOP   EQU     OF900H  ; TOP OF STACK RAM
                ;
F800            SRBOT   EQU     OF800H  ; BOTTOM OF STACK RAM
                ;
                +       PAGE
```



```
                +
                ;
                ; VARIABLES:-
                ;
0100                    ORG     0100H
                ;
                ; USER STATE:
                ;
                ; FOLLOWING ARE SAVED BY SOFT BREAK
                ;
                SYSBOT:
                ;
0100            CURRNT: DS      2       ; START OF CURRENT LINE
                ;
0102            BRKPT:  DS      2       ; START OF CURRENT COMMAND
                ;
0104            LOPVAR: DS      2       ; POINTS TO CURRENT LOOP VARIAB
                                        ; 0 IF NO RUNNING LOOP
                ;
0106            LSTPF:  DS      1       ; FLAG FOR INTEGER/FPT LOOP
                                        ; AND IMPLICIT/EXPLICIT STEP
                ;
0107            LSTEP:  DS      4       ; STEP VALUE IF EXPLICIT
                ;
010B            LCOUNT: DS      4       ; LOOP ITERATION COUNT
                ;
010F            LOPPT:  DS      2       ; POINTER TO START LOOP
                ;
0111            LOPLN:  DS      2       ; POINTER TO START LOOP LINE
                ;
0010            FRAME   EQU     $-LOPVAR+1 ; ALLOW FOR FLAGS WHEN PUSHI
                ;
0113            STKGOS: DS      2       ; STACK LEVEL AT LAST GOSUB
                                        ; 0 IF NO ACTIVE CALL
                SYSTOP:
                ;
                STRFL:                  ; TRACE/STEP FLAGS TOGETHER
                ;
0115            TRAFL:  DS      1       ; TRACE FLAG
0116            STEPF:  DS      1       ; STEP FLAG
                ;
0117            RDIPF:  DS      1       ; FLAG SET WHILE RUNNING INPUT
0118            RUNF:   DS      1       ;   "   "   "       "   PROGRA
                ;
                ; PREVIOUS 2 BYTES MUST BE CONSECUTIVE
                +       PAGE
```

```
+
                ;
                ; RUNTIME SCRATCH AREA
                ;
                GSNWK:                  ; SCRATCH AREA FOR GOSUB/NEXT (2 BYTES
                LISW1:                  ; START OF LISTED AREA
                ;
0119            COLWK:  DS      2       ; SCRATCH AREA FOR SCOLG,SCOLT (4 BYTE
                ;
011B            LISW2:  DS      2       ; END LISTED AREA
                ;
                ; SAVE AREA FOR RESTART ON ERROR.
                ;
011D            ERSSP:  DS      2       ; STACK POINTER
                ;
011F                    DS      3       ;*
                ;*
0122            ERSFL:  DS      1       ; SET IF ENCODING A STORED LINE
                ;
                ; DATA/READ VARIABLES
                ;
0123            DATAC:  DS      1       ; OFFSET OF NEXT CH TO ENCODE IN "DATA
                ;
0124            DATAP:  DS      2       ; POINTER TO CURRENT DATA LINE
                ;!DATAQ: DS     2       ; POINTER AFTER CURRENT D. LINE IF ANY
                ;
0126            CONFL:  DS      1       ; SET IF THERE IS A SUSPENDED PROGRAM
                ;
0127            STACK:  DS      2       ; CURRENT BASE STACK LEVEL
                ;
0015            SFRAME  EQU     SYSTOP-SYSBOT
                ;
                ; SCRATCH LOCN FOR EXPRESSION EVALUATION
                ;
0129            WORKE:  DS      4
                ;
                ; RANDOM NUMBER KERNEL
                ;
012D            RNUM:   DS      4
                ;
                ;!RNDLY: DS     1       ; RANDOM NUMBER DELAY COUNT
+                       PAGE
```



```
+
                ;
                ; OUTPUT SWITCHING
                ;
0131            OTSW:   DS      1       ; 0 TO OUTPUT TO SCREEN+RS232
                                        ; 1 OUTPUT TO SCREEN
                                        ; 2 TO EDIT BUFFER
                                        ; 3 TO DISK
                ;
                ; INPUT SWITCHING
                ;
                ;!INSW: DS      1
                                        ; 0 FROM KEYBOARD
                                        ; 1 FROM DISK
                ;
                ;
                ; ENCODING INPUT SOURCE SWITCHING
                ;
0132            EFEPT:  DS      2       ; POINTER
0134            EFECT:  DS      1       ; COUNT
                ;
0135            EFSW:   DS      1       ; SET 0:        INPUT FROM KB/SCREEN
                                        ;     1:          "     "   STRING
                                        ;     2:          "     "   EDIT BUF
                ;
                ; VARIABLES USED DURING EXPRESSION ENCODING
                ; (COULD OVERLAP WITH RUNTIME VARIABLES)
                ;
0136            TYPE:   DS      1       ; TYPE OF LATEST EXPRESSION OR ITEM
                ;
0137            RGTOP:  DS      1       ; LATEST PRIORITY OPERATOR
                ;
0138            OLDOP:  DS      1       ; OLD PRIORITY+OPERATOR
                ;
0139            HOPPT:  DS      2       ; PTR TO PLACE FOR OPERATOR
                ;
013B            RGTPT:  DS      2       ; PTR TO RGT OPERAND LATEST OPERATOR
                ;
                ; ORDER OF LAST 7 BYTES IS IMPORTANT
                ;
+               PAGE
```

```
                +
                ;
                ; MASK TO SELECT CASSETTE 1 OR 2
                ;
013D            CASSL:  DS      1       ; #10 FOR CASSETTE 1,#20 FOR 2
                ;
                ; ENCODED INPUT BUFFER
                ;
013E            EBUF:   DS      128     ; USED ALSO BY UTILITY
                ;
                ; INTERRUPT HANDLER VARIABLES
                ;
005F            TICIM   EQU     05FH    ; CURRENT INTERRUPT MASK
                ;
01BE            TIMER:  DS      2       ; TIMER LOCATION
                ;
01C0            CTIMR:  DS      1       ; CURSOR CLOCK
                ;
000F            CTIMV   EQU     15      ; FLASH TIME IN 20 MS UNITS
                ;
01C1            KBXCT:  DS      1       ; EXTEND KB SCAN TIME COUNTER
                ;
0002            KBXCK   EQU     2       ; KB SCAN TIME (UNITS OF 16 MS)
                                        ; RAND ROUTINE NEEDS THIS EVEN
                ;
                ; INTERRUPT MASKS DEFINITIONS
                ;
FFFB            SNDIAD  EQU     TICC+0BH ; SOUND TIMER ADDR
0008            SNDIM   EQU     08H     ; SOUND INT MASK BIT
                ;
FFFC            KBIAD   EQU     TICC+0CH ; KB TIMER ADDR
0040            KBIM    EQU     40H     ; KEYBOARD "    "    "
                ;
0080            CLKIM   EQU     080H    ; CLOCK   "    "    "
                ;
0004            STKIM   EQU     04H     ; STACK   "    "    "
                +       PAGE
```



```
                +
                ;
                ; IO LOCATIONS
                ;
                ; !POROM:       DS      1       ; MEMORY OF
                ; !POR1M:       DS      1       ; LAST OUTPUTS TO
0040            POROM   EQU     40H     ; OUTPUT PORTS
                ;
                ; SOUND CONTROL BLOCK STORAGE
                ;"
000E            SCBL    EQU     14      ; LENGTH OF A SOUND CONTROL BL
0009            NCBL    EQU     9       ;    "   "   NOISE      "
                ;
01C2            SCBO:   DS      3*SCBL+NCBL ; SOUND + NOISE CHANNELS
                ;
                ; ENVELOPE STORAGE
                ;
0040            ENVLL   EQU     64      ; NUMBER OF BYTES/ENVELOPE
                ;
0002            NUMENV  EQU     2       ; NUMBER OF ENVELOPES
                ;
01F5            ENVST:  DS      NUMENV*ENVLL ; ENVELOPE STORAGE
                ;
0275            IMPTAB: DS      'Z'-'A'+1 ; IMPLICIT TYPE TABLE
                ;
028F            IMPTYP: DS      1       ; DEFAULT NUMBER TYPE
                ;
0290            REQTYP: DS      1       ; REQUIRED NUMBER TYPE
                ;
                ; SPARE VARIABLE SPACE
                ;
0291                    DS      10
0291            DATA0   EQU     0291H   ;*
0293            RNDLY   EQU     0293H   ;*
0294            POROM   EQU     0294H   ;*
0295            POR1M   EQU     0295H   ;*
0296            INSW    EQU     0296H   ;*
                +       PAGE
```

```
                +
                ;
                ; HEAP/TEXT BUFFER/SYMTAB POINTERS
                ;
29B             HEAP:   DS      2       ; START OF HEAP
                ;
29D             HSIZE:  DS      2       ; SIZE OF HEAP
100             HSIZD   EQU     100H    ; DEFAULT SIZE
                ;
29F             TXTBGN: DS      2       ; START OF TEXT BUFFER
                ;
                TXTUSE:                 ; END TEXT AREA AND
2A1             STBBGN: DS      2       ; START SYMBOL TABLE
                ;
2A3             STBUSE: DS      2       ; END SYMBOL TABLE
                ;
2A5             SCRBOT: DS      2       ; BOTTOM OF SCREEN RAM AREA
                ;
                +       PAGE
```



```
                +
                ;
                ; KEYBOARD VARIABLES + CONSTANTS
                ;
02A7            KBTPT:  DS      2       ; POINTER TO CODE TABLE
                ;
02A9            MAP1:   DS      8       ; LATEST SCAN OF KEYS
                ;
02B1            MAP2:   DS      8       ; PREVIOUS SCAN
                ;
02B9            KNSCAN: DS      1       ; SET TO SCAN FOR BREAK ONLY
                ;
0004            KBLEN   EQU     4       ; LENGTH OF ROLLOVER BUFFER
                KEYL:
02BA            KLIND:  DS      KBLEN   ; CIRCULAR BUFFER FOR KEYS PRE
                ;
02BE            KLIIN:  DS      2       ; NEXT POSN FOR INPUT TO KLIND
02C0            KLIOU:  DS      2       ; NEXT POSN FOR OUTPUT FROM KL
                ;
02C2            RPCNT:  DS      1       ; COUNT FOR REPT
                ;
02C3            SHLK:   DS      1       ; SET IF "SHIFT INVERT"
                ;
                        IF SUSP
                ;
02C4            KBRFL:  DS      1       ; FLAG FOR "BREAK PRESSED"
                ;
                        ENDIF
                ;
02B0            SHLOC   EQU     MAP1+7  ; BYTE CONTAINING SHIFT
0040            SHMSK   EQU     040H    ; SHIFT KEY BIT
                ;
02AF            RPLOC   EQU     MAP1+6  ; BYTE CONTAINING REPT KEY
0020            RPMSK   EQU     020H    ; REPT KEY BIT
                ;
0002            RPLIM   EQU     2       ; TIMING FOR REPT
                ;
0040            BRSEL   EQU     040H    ; COLUMN SELECT MASK FOR BREAK
0040            BRMSK   EQU     040H    ; BREAK KEY BIT
                ;
0020            BRLIM   EQU     20H     ; TIMING FOR HARD BREAK
                ;
                +       PAGE
```

```
DAI 8080 ASSEMBLY SERVICE,D2.2                          PAGE 21
BASIC V1.0 DISK EDIT 7 2-MARCH-80


                +
                ;
                ;   DISC/CASSETTE SWITCHING VECTOR
                ;
                IOVEC:
                ;
02C5            WOPEN:  DS      3
                ;
02C8            WBLK:   DS      3
                ;
02CB            WCLOSE: DS      3
                ;
02CE            ROPEN:  DS      3
                ;
02D1            RBLK:   DS      3
                ;
                RCLOSE:
02D4            RCLO:   DS      3
                ;
02D7            MBLK:   DS      3
                ;
02DA            RESET:  DS      3
                ;
02DD            DOUTC:  DS      3
                ;
02E0            DINC:   DS      3
                ;
02E3                    DS      3       ; SPARE
                ;
02E6            TAPSL:  DS      2
                ;
02E8            TAPSD:  DS      2
                ;
02EA            TAPST:  DS      2
                ;
                VAREND:
                VARLAST:
                ;
02EC            RAM     SET     $
                ;
                +       PAGE
```



```
)AI 8080 ASSEMBLY SERVICE,D2.2                          PAGE 22
3ASIC V1.0 DISK EDIT 7 2-MARCH-80


                +
                ;
C6C0                    ORG     0C6C0H  ; START OF BASIC
                ;
                ; BANK SWITCHING RESTARTS
                ;
                ; THE FOLLOWING ROUTINES SWITCH THE PAGED
                ;  BANKS OF ROM.  THEY ARE ENTERED VIA RST INSTRUCTIONS
                ;
                MARST:
                ;
C6C0 E1                 POP     H
                ;
C6C1 F3                 DI
                ;
C6C2 224300             SHLD    RSWK2   ; SAVE HL
C6C5 F5                 PUSH    PSW
C6C6 E1                 POP     H
C6C7 224100             SHLD    RSWK1   ; PSW
                ;
C6CA 2640               MVI     H,040H  ; BANK SELECT BITS FOR MATH PA
C6CC 3AD400             LDA     MVECA   ; OFFSET OF START HW/SW VECTOR
                MRS10:
C6CF E3                 XTHL
C6D0 86                 ADD     M       ; ADD ENTRY NUMBER
C6D1 23                 INX     H
C6D2 E3                 XTHL
                ;
C6D3 6F                 MOV     L,A     ; COMPLETE ENTRYPOINT ADDRESS
C6D4 3A4000             LDA     POROM   ; BANK SELECT PORT STATUS
C6D7 F5                 PUSH    PSW     ; REMEMBER
C6D8 E63F               ANI     03FH    ; KEEP OTHER BITS
C6DA B4                 ORA     H       ; ADD NEW SELECT BITS
C6DB 324000             STA     POROM   ; UPDATE MEMORY
C6DE 3206FD             STA     PORO    ; AND PORT
                ;
C6E1 26E0               MVI     H,VECA SHR 8
C6E3 CDF2C6             CALL    MRDCL
                ;
C6E6 E3                 XTHL
C6E7 F5                 PUSH    PSW
C6E8 7C                 MOV     A,H
C6E9 324000             STA     POROM   ; REINSTATE MEMORY
C6EC 3206FD             STA     PORO    ; + PORT
C6EF F1                 POP     PSW
C6F0 E1                 POP     H
C6F1 C9                 RET             ; BACK TO CALLER
                ;
```

```
DAI 8080 ASSEMBLY SERVICE, D2.2                    PAGE 23
BASIC V1.0 DISK EDIT 7 2-MARCH-80

                MRDCL:
C6F2 E5                 PUSH    H
C6F3 2A4100             LHLD    RSWK1
C6F6 E5                 PUSH    H
C6F7 F1                 POP     PSW
C6F8 2A4300             LHLD    RSWK2
C6FB FB                 EI
C6FC C9                 RET

                        PAGE
```


THIS PROGRAM NAMED SUM IS CALLING A MACHINE LANGUAGE SUBROUTINE LOADED AS AN ARRAY ''A'' NAMED ''SUM A'' THE SUBROUTINE ,LOCATED AT #3FC , PERFORMS INTEGER CALCULATION WITH 64 DIGITS RESOLUTION. YOU MUST LOAD THE PROGRAM, STOP THE RECORDER IF YOU DO NOT USE THE REMOTE CONTROL, RUN THE PROGRAM WHAT IS NOW LOADING THE ROUTINE AS AN ARRAY AND ASK YOU THE OPERATION TO PERFORM I.E. 12345+432 <RETURN> AND GIVES THE RESULT. IF YOU PRESS THE BREAK KEY TO CONTINUE YOU HAVE NOW TO RUN 35 ,OR FIRST TYPE 1 <RETURN> TO 24 <RETURN> WHAT WILL ERASE THIS TEXT AND LOADA ROUTINE AND YOU CAN NOW MAKE A NORMAL RUN. IF YOU WANT TO SAVE THE PROGRAM AND THE ROUTINE YOU MUST SAVE''PROGRAM NAME'' STOP RECORDER, SAVEA A''ROUTINE NAME''

YOU WILL NOTICE IF YOU LIST THE PROGRAM THAT 3 FIRST LINES ARE CLEAR 2000, DIM A(20,20), LOADA A''SUM A'' AFTER YOU HAVE LOADED THE ARRAY YOU CANNOT EDIT NOR CLEAR NOR DIM ARRAYS ALREADY DIMENSIONED.

PRESS ANY KEY CONTINUE THE PROGRAM LOADING ROUTINE

```
10    CLEAR 2000
20    DIM A(20,0,20,0)
22    LOADA A "SUM A"
35    PRINT "WHAT IS YOUR SUM    ";
40    INPUT A$
45    PRINT
50    CALLM #3FC,A$
60    PRINT "HERE IS THE ANSWER!",A$
70    GOTO 35
```

```
03F0 00 00 00 00 00 00 00 00 00 00 00 00 00 F5 C5 D5 7E
0400 23 66 6F E5 4E 23 CD 2F 05 11 99 06 CD F1 04 CA
0410 99 04 78 32 5E 07 7E 36 20 32 60 07 23 0D CA 89
0420 04 11 DA 06 CD F1 04 78 32 5F 07 21 1B 07 11 99
0430 06 01 DA 06 3A 60 07 FE 2A CA 6B 04 FE 2F CA 7F
0440 04 FE 2B CA 52 04 FE 2D C2 89 04 3A 57 07 2F 32
0450 5F 07 3A 5E 07 A7 CC F1 05 C4 DA 05 C5 D1 3A 5F
0460 07 A7 CC F1 05 C4 DA 05 C3 92 04 CD 06 06 CA 89
0470 04 3A 5E 07 47 3A 5F 07 A8 32 5D 07 C3 92 04 CD
0480 22 06 CA 89 04 C3 71 04 E1 E1 D1 C1 F1 23 36 3F
0490 23 C9 CD 61 07 3A 5D 07 E1 E5 23 06 00 A7 CA A5
04A0 04 36 2D 23 04 11 5B 07 1B 1A A7 CA A8 04 E5 21
04B0 E5 F8 19 4D E1 1A F6 30 77 23 1B 0D FA EB 04 C2
04C0 B5 04 F1 5E 16 00 4A E5 19 7E FE 20 C2 D5 04 2B
04D0 C3 1D CD C9 04 79 D6 03 FA E6 04 0D 0D E1 E5 73
04E0 23 19 36 00 23 71 E1 D1 C1 F1 C9 2B 36 30 C3 C2
04F0 04 06 00 7E E6 30 FE 30 CA 15 05 2B 23 0D C8 7E
0500 36 20 FE 20 CA FC 04 23 FE 2B CA 15 05 FE 2D C2
0510 79 04 3E FF 47 7E FE 20 CA 29 05 E6 30 FE 30 C0
0520 C9 B5 05 7E 36 20 E6 0F 12 23 0D C2 15 05 C9 E5
0530 21 99 06 1E 40 36 FF 23 1D C2 35 05 21 DA 06 1E
0540 40 36 FF 23 1D C2 41 05 21 1B 07 1E 40 36 00 23
0550 1D C2 4D 05 AF 32 5D 07 32 5E 07 32 5F 07 E1 C9
0560 E5 D5 C5 F5 11 40 00 19 EB E1 CD 71 05 C1 D1 E1
0570 C9 D5 1B 1A A7 C2 83 05 7B B2 C2 72 05 D1 AF 32
0580 5D 07 C9 F2 9A 05 3A 5D 07 2F 32 5D 07 D5 13 1B
0590 1A 2F 3C 12 7B B2 C2 8F 05 D1 13 13 E5 06 00 7E
05A0 FE 06 00 F2 B6 05 95 C6 0A FA A6 05 77 23 7B B2
05B0 C2 9F 05 E1 D1 C9 04 D6 0A F2 B6 05 C3 A6 05 C5
05C0 D5 CD C9 05 D1 C1 FE 80 C9 1A F5 AF 12 F1 13 47
05D0 1A F5 78 12 F1 A7 F2 CE 05 C9 F5 D5 E5 CD E4 05
05E0 E1 D1 F1 C9 1A FE FF C8 2F 3C B6 77 13 23 C3 E4
05F0 05 F5 D5 E5 CD FB 05 E1 D1 F1 C9 1A FE FF C8 B6
0600 77 13 23 C3 FB 05 0A 3D 02 FA 12 06 CD F1 05 C3
0610 06 06 CD 60 05 03 3A 2F D6 01 D8 CD BF 05 C8 C3
0620 06 06 AF 32 5C 07 E5 21 D9 06 2B 7E FE FF C2 36
0630 06 36 00 C3 2A 06 E1 36 01 CD 83 06 FA 52 06 23
0640 D5 C5 D1 CD BF 05 D1 C8 3A 5C 07 3C 32 5C 07 C3
0650 37 06 35 CD 83 06 2B D5 C5 D1 CD 70 06 D1 3A 5C
0660 07 3D 32 5C 07 F8 34 CD 83 06 F4 52 06 C3 66 06
0670 E5 D5 CD 78 06 D1 C1 C9 13 1A 1B 12 13 FE FF C8
0680 03 78 06 E5 C5 D5 D5 E1 C5 D1 CD DA 05 CD 60 05
0690 D1 C1 F1 3A 5D 07 A7 C9 80 00 00 00 00 00 00 00
06A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0700 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0710 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0720 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0760 00 21 1B 07 C3 60 05 00 00 00 00 00 00 00 00 00
```


REAL TIME CLOCK

```
5     CLEAR 300
6     POKE #29C,3:POKE #29E,0:POKE #3EC,#80:POKE #3ED,#28
10    FOR T%=0 TO 11:READ D%
20    FOR T1%=0 TO 15:READ D1%
25    IF D1%>=#100 THEN D1%=(PEEK(#2A6) IAND #FE IOR #E)+D1%-#100
30    POKE D%,D1%:D%=D%+1:NEXT:NEXT
40    POKE #71,#3:POKE #70,#0
100   DATA #300,#C5,#D5,#E5,#F5,#21,#99,#03,#06,#0A,#0E,#06,#16,#00,#1E,#32,#
110   DATA #310,#7B,#BE,#C2,#57,#03,#72,#23,#34,#78,#BE,#C2,#5E,#03,#72,#23,#
120   DATA #320,#79,#BE,#C2,#5E,#03,#72,#23,#34,#78,#BE,#C2,#5E,#03,#72,#23,#
130   DATA #330,#79,#BE,#C2,#5E,#03,#72,#23,#34,#78,#BE,#C2,#5E,#03,#72,#23,#
140   DATA #340,#29,#23,#3E,#02,#BE,#C2,#5E,#03,#2B,#3E,#04,#BE,#C2,#5E,#03,#
150   DATA #350,#00,#23,#36,#00,#C3,#5E,#03,#F1,#E1,#D1,#C1,#C3,#49,#09,#3A,#
160   DATA #360,#101,#FE,#7A,#C2,#57,#03,#21,#BA,#03,#7E,#06,#30,#32,#F1,#100
170   DATA #370,#7E,#C6,#30,#32,#F3,#100,#23,#7E,#C6,#30,#32,#F7,#100,#23,#7E
180   DATA #380,#30,#32,#F9,#100,#23,#7E,#C6,#30,#32,#FD,#100,#23,#7E,#C6,#30
190   DATA #390,#FF,#100,#3E,#FF,#32,#FC,#100,#32,#EE,#100,#32,#F0,#100,#32,#
#100
200   DATA #3A0,#32,#F4,#100,#32,#F6,#100,#32,#F8,#100,#32,#FA,#100,#32,#FC,#
,#32
210   DATA #3B0,#F5,#100,#3C,#30,#101,#00,#C3,#5F,#03,#1A,#00,#00,#00,#00,#00
0
500   INPUT "INPUT THE TIME  < HH,MM,SS > "1T$:PRINT :A%=#3BF
510   FOR D%=0 TO LEN(T$)-1:T1$=MID$(T$,D%,1)
520   IF ASC(T1$)>47 AND ASC(T1$)<58 THEN POKE A%,VAL(T1$):A%=A%-1:IF A%=#3B
TH 530
550   NEXT:GOTO
```

```
AP UTILITY V2.3
 0700 3FF

0700 09 05 55 F5 21 B9 03 06 0A 0E 06 16 00 1E 32 34
0710 78 BE C2 5E 03 72 23 34 78 BE C2 5E 03 72 23 34
0720 78 BE C2 5E 03 72 23 34 78 BE C2 5E 03 72 23 34
0730 78 BE C2 5E 03 72 23 34 78 BE C2 5E 03 72 23 34
0740 28 23 3E 02 BE C2 5E 03 2B 3E 04 BE C2 5E 03 36
0750 00 23 36 00 C3 5E 03 F1 E1 D1 C1 C3 A9 D9 3A EF
0760 7F FE 7A C2 57 03 21 9A 03 7E C6 30 32 F1 7E 23
0770 7E C6 30 32 F3 7E 23 7E C6 30 32 F7 7E 23 7E C6
0780 30 32 F9 7E 23 7E C6 30 32 FD 7E 23 7E C6 30 32
0790 FF 7E 3E FF 32 EC 7E 32 EE 7E 32 F0 7E 32 F2 7E
07A0 32 F4 7E 32 F6 7E 32 F8 7E 32 FA 7E 32 FC 7E 32
07B0 FE 7E 32 00 7F 00 C3 5F 03 24 09 01 00 02 06 00
07C0 45 35 2C 23 46 35 2C 23 32 31 2C 23 42 39 2C 23
07D0 30 33 2C 23 30 36 2C 23 30 41 2C 23 30 45 2C 23
07E0 30 36 2C 23 31 36 2C 23 30 30 2C 23 00 07 30 36
07F0 30 31 35 32 35 00 01 35 80 01 32 80 19 18 00 00
```



```
*
R O T A T I N G   P Y R A M I D
================================


2     PRINT "ROTATING PYRAMIDE ,1,2,3 AND 4 ARE USED"
3     PRINT "WITH REPT KEY FOR ROTATION":WAIT TIME 400
5     MODE 6:MODE 6:SF=3.5:REM MODE +SCALING FACTOR
6     COLORG 0 15 0 15
7     GOSUB 2000:REM INITIALISE DATA
90    REM
92    GOSUB 800:REM DRAW NEW SHAPE
95    COLORG 0 15*(1-0) 15*0 15
96    GOSUB 900:REM ERASE OLD SHAPE
97    0=1.0-0
99    KS=ABS(KS)
100   A=GETC:IF A<ASC("0") THEN 100
120   FOR P=1.0 TO NP
130   XX(P)=X(P):YY(P)=Y(P)
140   NEXT
141   REM
150   ON A-ASC("0") GOTO 500,510,600,610,700,710
160   GOTO 100
161   REM
162   REM
500   KS=-KS
510   FOR P=1.0 TO NP
520   X=X(P):Y=Y(P)
530   X(P)=X*KC+Y*KS
540   Y(P)=Y*KC-X*KS
550   NEXT
560   GOTO 90
590   REM
591   REM
600   KS=-KS
610   FOR P=1.0 TO NP
620   Y=Y(P):Z=Z(P)
630   Y(P)=Y*KC+Z*KS
640   Z(P)=Z*KC-Y*KS
650   NEXT
660   GOTO 90
661   REM
662   REM
700   KS=-KS
710   FOR P=1.0 TO NP
720   Z=Z(P):X=X(P)
730   Z(P)=Z*KC+X*KS
740   X(P)=X*KC-Z*KS
```

```
750   NEXT
760   GOTO 90
800   REM
801   REM DRAW NEW PICTURE
802   REM
810   FOR L=1.0 TO NL
820   PA=LA(L)
830   PB=LB(L)
840   DRAW X(PA)+XC,V(PA)+VC X(PB)+XC,V(PB)+VC 17+0*2
850   NEXT
860   RETURN
900   REM
901   REM ERASE OLD PICTURE
902   REM
910   FOR L=1.0 TO NL
920   PA=LA(L)
930   PB=LB(L)
940   DRAW XX(PA)+XC,VV(PA)+VC XX(PB)+XC,VV(PB)+VC 18-2*0
950   NEXT
960   RETURN
990   REM
991   REM DATA SETUP ROUTINE
992   REM
2000  PHI=PI/20.0
2010  KS=SIN(PHI)
2020  KC=COS(PHI)
2030  XC=XMAX/2.0
2040  VC=VMAX/2.0
2050  0=1.0
2100  READ NP,NL
2110  DIM X(NP),V(NP),Z(NP)
2120  DIM XX(NP),VV(NP)
2130  DIM LA(NL),LB(NL)
2131  REM
2200  FOR P=1.0 TO NP
2210  READ X(P),V(P),Z(P)
2211  X(P)=X(P)*SF
2212  V(P)=V(P)*SF
2213  Z(P)=Z(P)*SF
2220  NEXT
2221  REM
2230  FOR L=1.0 TO NL
2240  READ LA(L),LB(L)
2250  NEXT
2251  REM
2260  GOSUB 800
2270  RETURN
2300  REM
2301  REM DATA
2302  REM
2800  REM NUMBER OF POINTS AND NUMBER OF LINES
2900  DATA 5,8
2901  REM
2903  DATA 0,0,20
2904  DATA 20,20,-20
2905  DATA 20,-20,-2
2906  DATA -20,20,-2
2907  DATA -20,-20,-
2909  REM
2910  DATA 1,2
2911  DATA 1,3
2912  DATA 1,4
```



```
2913  DATA 1,5
2914  DATA 2,3
2915  DATA 2,4
2916  DATA 3,5
2917  DATA 4,5
2999  DATA 8,12
4000  DATA 1,2
4001  REM DATA FOR SOMETHING ELSE!
4002  REM
4009  DATA 20,20,20
4010  DATA 20,20,-20
4020  DATA 20,-20,20
4030  DATA 20,-20,-20
4040  DATA -20,20,20
4050  DATA -20,20,-20
4060  DATA -20,-20,20
4070  DATA -20,-20,-20
4110  DATA 1,3
4120  DATA 1,5
4130  DATA 2,4
4140  DATA 2,6
4150  DATA 3,4
4160  DATA 3,7
4170  DATA 4,8
4180  DATA 5,6
4190  DATA 5,7
4210  DATA 7,8
9999  END
*
```

C R A P S

==========

```
1     C1=1.0
2     C2=0.0
3     C3=14.0
4     C0=13.0
10    COLORG C0 C1 C2 C3:COLORT C0 0 0 0
11    MODE 3A
12    H=GETC
100   REM DRAW 14,19 14,68 C1
110   REM DRAW 14,68 63,68 C1
120   REM DRAW 63,68 63,19 C1
130   REM DRAW 63,19 14,19 C1
140   FILL 15,20 62,67 C2
150   REM DRAW 94,19 94,68 C1
160   REM DRAW 94,68 143,68 C1
170   REM DRAW 143,68 143,19 C1
180   REM DRAW 143,19 94,19 C1
190   FILL 95,20 142,67 C2
200   GOSUB 1200
210   PFS=0.0:TOSS%=0
212   CURSOR 0,3:PRINT "              TO SHOOT CRAPS PRESS ANY KEY          ";
213   CURSOR 0,2:PRINT "     point                                    tosses";
214   CURSOR 0,1:PRINT "                                                    ";
215   CURSOR 0,0:PRINT "                                                    ";
216   CURSOR 28,2:PRINT "$";:CURSOR 28,2
220   GOSUB 1300
251   IF SUM%=7.0 OR SUM%=11.0 THEN CURSOR 25,1:GOSUB 1500:GOTO 210
252   IF SUM%=2.0 OR SUM%=3.0 OR SUM%=12.0 THEN CURSOR 24,1:GOSUB 1600:GOTO 2
253   POINT%=SUM%
254   GOSUB 1400:GOSUB 1300
255   IF POINT%=SUM% THEN CURSOR 25,1:GOSUB 1500:GOTO 210
260   IF SUM%=7 THEN CURSOR 25,1:GOSUB 1600:GOTO 210
280   GOTO 254
700   D=1.0+INT(10.0*RND(1.0)):IF D>6.0 GOTO 700
800   A=V+19.0
801   A1=A+7.0
802   B=V+35.0
803   B1=B+7.0
804   C=V+51.0
805   C1=C+7.0
810   IF D=1.0 OR D=3.0 OR D=5.0 THEN FILL B,40 B1,47 C3
820   IF D=1 THEN RETURN
830   FILL A,56 A1,63 C3
835   FILL C,24 C1,31 C3
```



```
840   IF D<4 THEN RETURN
850   FILL A,24 A1,31 C3
855   FILL C,56 C1,63 C3
860   IF D<6 THEN RETURN
870   FILL A,40 A1,47 C3
875   FILL C,40 C1,47 C3
880   RETURN
1200  FILL 19,24 58,63 C2
1210  FILL 99,24 138,63 C2
1220  V=0.0:GOSUB 700
1230  SUM%=INT(D)
1240  V=80.0:GOSUB 700
1245  SUM%=SUM%+INT(D)
1250  RETURN
1300  WAIT TIME 10:H=GETC:IF H=0.0 GOTO 1300:GOSUB 1200:RETURN
1400  CURSOR 6,1:IF POINT%<>0 THEN PRINT POINT%," ";
1401  TOSS%=TOSS%+1:CURSOR 47,1:PRINT TOSS%:CURSOR 28,2:RETURN
1500  PRINT "you win":JF=1.0:WAIT TIME 200:RETURN
1600  PRINT "you lose":JF=1.0:WAIT TIME 200:RETURN
```

*
R A N D O M L I N E S 3

```
=======================


5     COLORG 7 15 0 0
10    MODE 6
100   S%=X% MOD (XMAX):T%=Y% MOD (YMAX)
105   FOR A%=0 TO 60:X%=RND(XMAX):Y%=RND(YMAX)
110   DRAW S%,T% X%,Y% 15:DRAW S%,T% X%,Y% 0:S%=X%:T%=Y%
120   NEXT:WAIT TIME 100:GOTO 10
```

*
B U G

```
=====


5     MODE 5
10    X%=5:FOR 0%=YMAX-6 TO 0 STEP -1:X%=X%+1:GOSUB 100:NEXT
20    GOTO 5
100   DOT X%,0% 15
110   DOT X%-1,0%+1 13
120   DOT X%-2,0%+2 11
130   DOT X%-3,0%+3 8
140   DOT X%-4,0%+4 6
150   DOT X%-5,0%+5 3
160   DOT X%-6,Y%+6 1
170   RETURN
```

*
S O U N D S

```
===========


10    ENVELOPE 0 16:FOR A=0.0 TO 2.0:SOUND A 0 15 0 FREQ(33.0):NEXT
20    FOR A=5.0 TO 541.0 STEP A:GOSUB 100:NEXT
30    FOR Z=440.0 TO 33.0 STEP -(Z/100.0)
40    FOR G=0.0 TO 2.0:SOUND G 0 15 2 FREQ(Z+G)
50    NEXT G:WAIT TIME 5:NEXT Z:GOTO 10
100   Q=A MOD 3.0:R=(Q+1.0) MOD 3.0:S=(Q+2.0) MOD 3.0
110   SOUND Q 0 15 2 FREQ(A+32.0)
120   SOUND R 0 15 2 FREQ(A*A+32.0)
130   SOUND S 0 15 2 FREQ(A*A*A+32.0)
140   RETURN
```


*
C O L O R G R A P H I C S

```
==========================


10    MODE 2:GOSUB 20:MODE 4:GOSUB 20:MODE 6:GOSUB 20:GOTO 10
20    FOR A%=0 TO YMAX:DRAW 0,0 XMAX,A% 20+(A% MOD 3):NEXT
30    FOR A%=0 TO XMAX-1:DRAW 0,0 A%,YMAX 20+(A% MOD 3):NEXT
40    FOR S%=0 TO 20:COLORG RND(15) RND(15) RND(15) RND(15)
50    WAIT TIME 20:NEXT S%:RETURN
```

*
G R A P H I C S 2

```
==================


10    MODE 2:GOSUB 20:MODE 4:GOSUB 20:MODE 6:GOSUB 20:GOTO 10
20    FOR A%=0 TO YMAX STEP 3:W%=W%+1:DRAW 0,0 XMAX,A% 20+(W% MOD 3):NEXT
30    FOR A%=0 TO XMAX-1 STEP 3:W%=W%+1:DRAW 0,0 A%,YMAX 20+(W% MOD 3):NE
40    FOR A%=1 TO XMAX STEP 3:W%=W%+1:DRAW A%,0 XMAX,YMAX 20+(W% MOD 3):N
50    FOR A%=1 TO YMAX STEP 3:W%=W%+1:DRAW 0,A% XMAX,YMAX 20+(W% MOD 3):N
60    FOR S%=0 TO 20:COLORG RND(15) RND(15) RND(15) RND(15)
70    WAIT TIME 20:NEXT S%:RETURN
```

*
R A N D O M L I N E S

```
=======================


5     COLORG 7 15 0 0
10    MODE 4
100   S%=X% MOD (XMAX):T%=Y% MOD (YMAX)
105   FOR A%=0 TO 2:X%=RND(XMAX):Y%=RND(YMAX)
110   DRAW S%,T% X%,Y% 15:DRAW S%,T% X%,Y% 0:S%=X%:T%=Y%:NEXT:GOTO 10
```

```
5     ENVELOPE 0 15,2;10,2;15,2;10,2;0
6     ENVELOPE 1 15,5;12,5;10,100;0
10    REM music compose program
15    ENVELOPE 0 6
16    CLEAR 8000
17    DIM N$(50.0):DIM F%(50.0):DIM T(255.0):DIM E(255.0)
18    DIM V(255.0):DIM M(255.0):DIM D(255.0):DIM S(255.0)
20    DATA C0,65,C0+,69,D0,73,D0+,78,E0,82,F0,87,F0+,92,G0
21    DATA 98,G0+,104,A0,110,A0+,116,B0,123
30    DATA C,131,C+,138,D,147,D+,155,E,165,F,175,F+,185,G
31    DATA 196,G+,208,A,220,A+,233,B,247
40    DATA C1,262,C1+,277,D1,294,D1+,311,E1,330,F1,349,F1+
41    DATA 370,G1,392,G1+,415,A1,440,A1+,466,B1,494
50    DATA C2,523,C2+,554,D2,587,D2+,622,E2,659,F2,698,F2+
51    DATA 740,G2,784,G2+,831,A2,880,A2+,932,B2,988
60    FOR X=1.0 TO 48.0:READ N$(X):READ F%(X):NEXT
70    N$(0.0)="0":F%(0.0)=60000
75    N$(49.0)="C3":F%(49.0)=1046
90    PRINT CHR$(12)
100   REM compose
110   FOR X=1.0 TO 255.0
120   READ S(X):IF S(X)=999.0 THEN GOTO 190
125   READ E(X),NOTE$,V(X),D(X),M(X)
130   FOR V=0.0 TO 48.0
140   IF NOTE$=N$(V) THEN T(X)=F%(V):GOTO 180
150   NEXT V
180   NEXT
190   CURSOR 10,10
191   PRINT "from the motion picture ' THE STING '"
192   CURSOR 20,8:PRINT "THE ENTERTAINER "
194   CURSOR 30,6:PRINT "by SCOTT JOPLIN"
200   FOR P=1.0 TO X-1.0
210   SOUND S(P) E(P) V(P) M(P) FREQ(T(P))
211   WAIT TIME D(P)*5.0
220   NEXT
221   PRINT CHR$(12):SOUND OFF :WAIT TIME 10
225   CURSOR 10,10
226   PRINT "AFTER A BOTTLE OF WHISKY ....."
230   FOR P=1.0 TO X-1.0
240   SOUND S(P) E(P) V(P) M(P) FREQ(T(P)+RND(15.0))
241   WAIT TIME D(P)*5.0:NEXT
250   SOUND OFF :PRINT CHR$(12):POKE #7921,#56
251   CURSOR 2,10:PRINT "THANK YOU !"
300   DATA 0,1,D2,15,2,0,0,1,E2,15,2,0,0,1,C2,15,2,0
301   DATA 0,1,A1,15,4,0,0,1,B1,15,2,0,0,1,G1,15,4,0
302   DATA 2,1,D1,10,2,2,2,1,E1,10,2,0
303   DATA 2,1,C1,10,2,0,2,1,A,10,4,0,2,1,B,10,2,0
304   DATA 2,1,G,10,4,0
305   DATA 1,1,D,15,2,0,1,1,E,15,2,0,1,1,C,15,2,0
306   DATA 1,1,A0,15,4,0,1,1,B0,15,2,0,1,1,A0,15,2,0
307   DATA 1,1,G0+,15,2,0,1,1,G0,15,8,0
308   DATA 0,0,G,15,0,0,2,0,B,15,0,0,1,0,G1,15,4,0
309   DATA 0,0,0, 0,0,0,1,0,0,0,0,0,2,0,0,0,0,0
310   DATA 0, 0,D,10,2,0,0,0,D+,10,2,2,0,0,E,10,2,0
311   DATA 0,0,C1,10,5,0,0,0,E,10,2,0,0,0,C1,10,5,0
312   DATA 0,0,E,10,2,0,0,0,C1,10,8,0
313   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
314   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
315   DATA 0,0,D2+,12,0,0,2,0,F1+,12,2,0
```



```
316   DATA 0,0,E2,15,0,0,2,0,G1,15,2,0
317   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
318   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
319   DATA 0,0,E2,12,0,0,2,0,G1,12,4,0
320   DATA 0,0,B1,12,0,0,2,0,D1,12,2,0
321   DATA 0,0,D2,12,0,0,2,0,F1,12,4,0
322   DATA 0,0, C2,12,0,0,2,0,E1,12,8,0
323   DATA 2,0,0,0,0,0
324   DATA 0,0,D,12,2,0,0,0,D+,12,2,0
325   DATA 0,0,E,12,2,0,0,0,C1,12,5,0
326   DATA 0,0,E,12,2,0,0,0,C1,12,5,0
327   DATA 0,0,E,12,2,0,0,0,C1,12,10,0
328   DATA 0,0,A1,12,2,0,0,0,G1,12,2,0
329   DATA 0,0,F1+,12,0,0,2,0,C1,12,2,0
330   DATA 0,0,A1,12,2,0
331   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
332   DATA 0,0,E2,12,0,0,2,0,F1+,12,0,0,1,0,D0,12,3,0
333   DATA 0,0,D2,12,2,0,0,0,C2,12,2,0,0,0,A1,12,2,0
334   DATA 0,0,D2,12,0,0,2,0,F1,12,0,0,1,0,G0,12,8,0
335   DATA 0,0,0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,0
336   DATA 0,0,D,12,2,0,0,0,D+,12,2,0
337   DATA 0,0,E,12,2,0,0,0,C1,12,5,0
338   DATA 0,0,E,12,2,0,0,0,C1,12,5,0
339   DATA 0,0,E,12,2,0,0,0,C1,12,8,0
340   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
341   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
342   DATA 0,0,D2+,12,0,0,2,0,F1+,12,2,0
343   DATA 0,0,E2,12,0,0,2,0,G1,12,2,0
344   DATA 0,0,C2,12,0,0,2,0,E2,12,2,0
345   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
346   DATA 0,0,E2,12,0,0,2,0,G1,12,3,0
347   DATA 0,0,B1,12,0,0,2,0,D1,12,2,0
348   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
349   DATA 0,0,C2,12,0,0,2,0,E1,12,4,0
350   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
351   DATA 00,0,D2,12,0,0,2,0,F1,12,2,0
352   DATA 1,1,C,15,0,0,0,0,E2,12,0,0,2,0,G1,12,2,0
353   DATA 0,0,C2,12,0,0,2,0,E1,12,2,0
354   DATA 0,0,D2,12,0,0,2,0,F1,12,2,0
355   DATA 1,1,A0+,15,0,0,0,0,E2,12,0,0,2,0,G1,12,3,0
356   DATA 0,0,C2,12,0,0,2,0,G1,12,2,0
357   DATA 0,0,D2,12,0,0,2,0,G1,12,2,0
358   DATA 0,0,C2,12,0,0,2,0,G1,12,2,0
359   DATA 1,1,A0,15,0,0,0,0,E2,12,0,0,2,0,A1,12,2,0
360   DATA 0,0,C2,12,0,0,2,0,C2,12,2,0
361   DATA 0,0,D2,12,0,0,2,0,A1,12,2,0
362   DATA 1,1,G0+,15,0,0,0,0,E2,12,0,0,2,0,G1+,12,3,0
363   DATA 0,0,C2,12,0,0,2,0,A1,12,2,0
364   DATA 0,0,D2,12,0,0,2,0,A1,12,2,0
365   DATA 0,0,C2,12,0,0,2,0,A1,12,2,0
366   DATA 1,1,G0,15,0,0,0,0,E2,12,0,0,2,0,G1,12,2,0
367   DATA 0,0,C2,12,0,0,2,0,E1,12,0,0
368   DATA 0,0,D2,1,0,0,2,0,F1,12,2,0
369   DATA 1,1,G0,15,0,0,0,0,E2,12,0,0,2,0,G1,12,3,0
370   DATA 0,0,B1,12,0,0,2,0,D1,12,2,0
371   DATA 0,0,D2,12,0,0,2,0,F1,12,4,0
372   DATA 1,1,C0,15,0,0,0,0,C2,12,0,0,2,0,E1,12,4,0
1000  DATA 999
```

N E A R M B A N D I T

```
3     MODE 5A
3     COLORG 12 12 12 12
4     COLORT 12 0 0 0
9     CURSOR 0,3:PRINT "       Pralines         PRESS ANY KEY      Pralines";
0     CURSOR 0,2:PRINT "    red  red  red  = 10     WIN          x    x    -
0     CURSOR 0,1:PRINT "     x    x    x   =  3                  -    x    x
0     CURSOR 28,1:PRINT "$";:CURSOR 28,1
00    0%=64:GOSUB 1000
10    0%=160:GOSUB 1000
20    0%=256:GOSUB 1000
40    CURSOR 25,1:PRINT "        ";
41    CURSOR 28,1:PRINT "$";:CURSOR 28,1
42    A=GETC:IF A=0.0 GOTO 142
43    FOR Z=0.0 TO 15.0
44    Z1%=1+Z/6
45    ON Z1% GOTO 150,160,170
50    0%=64:GOSUB 900
55    NOE=K
60    0%=160:GOSUB 900
65    TWO=K
70    0%=256:GOSUB 900
72    TRE=K
75    NEXT Z
78    GOSUB 1500
80    CURSOR 25,1:PRINT "Pralines";:CURSOR 27,0:PRINT WINS%;" ";
82    WAIT TIME 100:GOTO 140
00    K=INT(RND(16.0))
10    IF K=8.0 GOTO 900
30    FILL 0%-8,90 0%+7,130 K
70    RETURN
000   FILL 0%-32,42 0%+31,170 0
001   FILL 0%-24,74 0%+23,138 8
002   RETURN
500   IF NOE=3 AND TWO=3 AND TRE=3 THEN WINS%=10:RETURN
510   IF NOE=TWO AND NOE=TRE THEN WINS%=3:RETURN
520   IF NOE=TWO THEN WINS%=1:RETURN
530   IF TWO=TRE THEN WINS%=1:RETURN
540   WINS%=0:RETURN
```



```
1     PRINT CHR$(12)
2     GOSUB 400
5     MODE 3
10    A=GETC
12    IF A=32.0 THEN 200
13    IF A=8.0 THEN 220
14    IF A=9.0 THEN 320
15    IF A<16.0 OR A>19.0 THEN 321
100   Y=Y+1.0:IF Y>YMAX THEN Y=YMAX
105   RETURN
110   Y=Y-1.0:IF Y<0.0 THEN Y=0.0
115   RETURN
120   X=X-1.0:IF X<0.0 THEN X=0.0
125   RETURN
130   X=X+1.0:IF X>XMAX THEN X=XMAX
135   RETURN
200   MODE 0:MODE 3:Y=0.0:X=0.0
210   GOTO 5
220   A=GETC:DOT X,Y 15
221   IF A=32.0 GOTO 200
222   IF A=9.0 GOTO 320
223   IF A<16.0 OR A>19.0 THEN 220
224   DOT X,Y 0:A=A-15.0:ON A GOSUB 100,110,120,130
225   GOTO 220
320   A=GETC:DOT X,Y 0
321   IF A=8.0 GOTO 220
322   IF A=32.0 GOTO 200
323   IF A<16.0 OR A>19.0 THEN 320
329   DOT X,Y 15:A=A-15.0:ON A GOSUB 100,110,120,130
330   GOTO 320
400   PRINT :PRINT
412   PRINT "LES DESSINS  S'OBTIENNENT  EN PRESSANT";
413   PRINT " UNE DES FLECHES":PRINT "          ";
430   PRINT "DANS LA DIRECTION QUI VOUS CONVIENT.":PRINT
432   PRINT " POUR EFFACER UN MORCEAU DE DESSIN ";
440   PRINT " REPLACEZ LE CURSEUR":PRINT "     ";
441   PRINT " A CET ENDROIT APRES AVOIR PRESSE";
442   PRINT " SUR CHAR DEL.":PRINT :PRINT "      ";
444   PRINT "POUR REPASSER EN MODE DESSIN";
445   PRINT " PRESSEZ SUR TAB":PRINT
470   PRINT "L'EFFACAGE  DE L'ECRAN  S'OBTIENT ";
480   PRINT " EN  PRESSANT LA BARRE"
481   PRINT "                D'ESPACEMENT"
490   PRINT :PRINT
491   INPUT "PRESSEZ LU ET RETURN APRES AVOIR FINI":Z$
492   IF LEFT$(Z$,1)="L" THEN 499
493   PRINT :GOTO 491
499   PRINT CHR$(12)
500   RETURN
```

GRAFTEXT SUBDEMO

```
*
1     CLEAR 1400
2     REM :DATA FOR GOSUB40040: X / Y / C / VFLAG / A$ / F
3     REM !!!! DELETE LINE 40 >>>>>> 70 !!!!!!!!!!!!!!!!!!!!
5     COLORG 8 1 3 5
10    MODE 5
20    COLORG 8 0 14 1
30    GOSUB 40012:FOR X=0.0 TO XMAX:DOT X,225+20*SIN(X/20.0) 15:NEXT
31    FOR X=200.0 TO 230.0 STEP 3.0:DRAW X,10 X,45 0:NEXT
32    FOR Y=125.0 TO 150.0 STEP 2.0:FILL 260,Y XMAX,Y+1 0:0=0+1.0:NEXT
33    X=10.0:Y=215.0:C=1.0:A$="DAI":VFLAG=0.0:F=2.0:GOSUB 40040
34    X=80.0:Y=215.0:C=6.0:A$="TEXT":GOSUB 40040
35    X=150.0:Y=215.0:C=5.0:A$="IN":GOSUB 40040
36    X=200.0:Y=215.0:C=0.0:F=2.0:A$="GRAFICS":GOSUB 40040
39    X=130.0:Y=190.0:C=2.0:F=1.0:A$="TEL. 02 / 3751114":GOSUB 40040
40    X=10.0:Y=200.0:C=0.0
41    A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ!#?$%&'()*=:-+;<>./1234567890"
50    GOSUB 40040
55    X=10.0:Y=170.0:C=3.0:F=2.0:GOSUB 40040
56    X=XMAX-10.0:Y=50.0:C=13.0:VFLAG=1.0:F=1.0:GOSUB 40040
60    VFLAG=0.0:X=10.0:Y=90.0:C=12.0:F=4.0:A$=LEFT$(A$,25):GOSUB 40040
65    GOTO 65
40012 DIM CAR$(90.0)
40021 FOR Z=32.0 TO 90.0:READ A$
40022 IF A$="STOP" THEN RETURN
40023 READ CAR$(Z):NEXT:RETURN
40040 X1=X:Y1=Y:IF F=0.0 THEN F=1.0
40041 FOR M=0.0 TO LEN(A$)-1.0
40042 T$=MID$(A$,M,1)
40050 GR$=CAR$(ASC(T$))
40060 FOR N=0.0 TO LEN(GR$)-1.0 STEP 4.0
40065 IF VFLAG=1.0 GOTO 40120
40070 IF MID$(GR$,N,1)="/" THEN X=X+(8.0*F):GOTO 40100
40080 ZZ=VAL(MID$(GR$,N,1)):YY=VAL(MID$(GR$,N+1,1))
40082 JC5%=X+ZZ*F:JC6%=Y+VAL(MID$(GR$,N+1,1))*F
40083 JC7%=X+VAL(MID$(GR$,N+2,1))*F:JC8%=Y+VAL(MID$(GR$,N+3,1))*F
40084 DRAW JC5%,JC6% JC7%,JC8% C
40085 IF F<1.5 THEN GOTO 40090
40086 JC9%=X+1+VAL(MID$(GR$,N+2,1))*F
40087 JC10%=Y+1+VAL(MID$(GR$,N+3,1))*F
40088 DRAW X+1+ZZ*F,Y+1+YY*F JC9%,JC10% C
40090 NEXT N
40100 IF X+8.0*F>=XMAX THEN X=X1:Y=Y-10.0*F
40102 NEXT M
40103 RETURN
40120 IF MID$(GR$,N,1)="/" THEN Y=Y-9.0*F:GOTO 40130
40130 JC1%=X+VAL(MID$(GR$,N+1,1))*F:JC2%=Y-VAL(MID$(GR$,N,1))*F
40131 JC3%=X+VAL(MID$(GR$,N+3,1))*F:JC4%=Y-VAL(MID$(GR$,N+2,1))*F
40132 DRAW JC1%,JC2% JC3%,JC4% C
40140 NEXT N
40180 IF Y-9.0*F<=0.0 THEN Y=Y1:X=X-9.0*F
40190 NEXT M
40200 RETURN
50000 DATA BLANCO,/,UITROEP!,31313337/,QOUTES,25274547/,#
50001 DATA 1353155521274147/,$,12424253244415262656313 7/
50010 DATA %,17271626125641514252/,&,121321315331155116273536/,'
50011 DATA 3537/,(,131513311537/
50020 DATA ),3153535555537/,*,125616523137/,+,32361454/,COMMA
50021 DATA 21303233/
```



```
50030 DATA -,1454/,.,31423241/,/,1256/,0,12162141525627471256/
50040 DATA 1,214131372637/,2,11511233444455564727162 7/,3
50041 DATA 12212141525334561757 4453/,4,414713531447/
50050 DATA 5,12212141525415451517 1757/,6,2141121514445253153737 57/
50051 DATA 7,2122223561757/,8,21412444274712131516525355 56/
50060 DATA 9,11313153535624541516274 7/,:,33333535/,;,213232333535/
50061 DATA <,14471441/
50070 DATA =,13531555/,>,21545427/,?,16272747343331313456/,APE,/
50080 DATA A,11155155135315373755/,B,11171747144411415253555 6/,C
50081 DATA 12162747475621414152/,D,1117114152561747/
50090 DATA E,1117115114441757/,F,111714441757/,G,12162757215151535343
50091 DATA H,111714545157/
50100 DATA I,214131372747/,J,122121415257/,K,111713572451/,L,1117115 1
50110 DATA M,11171735353435575751/,N,111751571652/,O,12162747565221 41
50111 DATA 1117144417475556/
50120 DATA Q,12162747565321313351/,R,11171747565514442451/,S
50131 DATA 12212141525532444151627474756/,T,17573137/
50130 DATA U,111721415157/,V,13175357133131 53/,W,11175157113333513334
50131 DATA X,11121716515257561256165 2/
50140 DATA Y,16175657163434563134/,Z,175712561151/
51000 DATA STOP
*
```

TRAFFICTEST

```
1     COLORG 8 1 3 5:MODE 5
2     ENVELOPE 1 15,10:0,10:
10    CLEAR 2000
30    GOSUB 40012
35    X=50.0:Y=230.0:C=14.0:F=1.5
36    A$="DAI TRAFFIC TEST":GOSUB 40040
110   DRAW 50,220 235,220 0
112   DRAW 0,170 280,170 0
115   P=170.0
120   READ A
125   IF A=999.0 THEN GOTO 140
130   READ B,C,D:DRAW A+50,B C+50,D 0:GOTO 120
140   A$="STOP FOR THE RED LIGHT":X=130.0:Y=80.0
141   C=3.0:F=1.0:GOSUB 40040
150   A$="NO REACTION ON GREEN !!":X=130.0:Y=60.0
151   C=5.0:F=1.0:GOSUB 40040
160   WAIT TIME 200:FILL 130,0 XMAX,100 8
200   REM TEST
210   C=INT(RND(2.0)):CO=3.0:IF C=1.0 THEN CO=5.0
215   SOUND 2 1 10 0 FREQ(800.0):WAIT TIME 20:SOUND OFF
220   WAIT TIME RND(50.0)
230   IF CO=3.0 THEN FILL 57,112 73,128 CO
235   IF CO=5 THEN FILL 57,87 73,103 5
237   IF CO=5 THEN GOTO 700
240   S=S+1.0:IF GETC=0.0 GOTO 240
250   FOR X=0.0 TO 250.0-S*2.0 STEP 3.0
251   FILL 300,X 310,X+1 1:SOUND 1 0 5 0 FREQ(31.0+X)
260   NEXT
265   SOUND OFF
270   MG=MG+10.0:NG=125.0+70.0-S/2.5
271   IF MG>280.0 THEN A$=" THE END":F=2.0:X=140.0:GOSUB 40040
272   IF MG>280.0 THEN WAIT TIME 1000:GOTO 1
275   IF NG<125.0 THEN NG=125.0
280   DRAW O,P MG,NG 15
290   O=MG:P=NG
295   S=S*1.5
300   IF S>=100.0 THEN A$=" WAKE UP !!          "
305   IF S>150.0 THEN A$=" YOU ARE SLOW !       "
310   IF S<100.0 THEN A$=" ATTENTION PLEASE !"
320   IF S<90.0 THEN A$=" NOT GOOD!             "
330   IF S<80.0 THEN A$=" MMMM...               "
340   IF S<70.0 THEN A$=" GOOD                  "
350   IF S<60.0 THEN A$=" VERY GOOD!            "
360   IF S<50.0 THEN A$=" EXCELLENT !           "
370   IF S<40.0 THEN A$=" SUPERB !              "
380   IF S<30.0 THEN A$=" MARVELLOUS !          "
390   IF S<20.0 THEN A$=" GENIUS !              "
400   X=150.0:Y=50.0:C=3.0:F=1.0:GOSUB 40040
490   WAIT TIME 50
491   FILL 57,112 73,128 8:FILL 57,87 73,103 8
495   FILL 300,100 XMAX,YMAX 8
496   FILL 100,0 XMAX,100 8
```



```
505   S=0.0
510   GOTO 200
700   FOR X=0.0 TO 200.0:IF GETC<>0.0 THEN GOTO 710
705   NEXT:GOTO 490
710   FOR X=0.0 TO 10.0:SOUND 1 0 10 0 FREQ(1000.0)
711   SOUND 1 0 12 2 FREQ(500.0):WAIT TIME 10:NEXT
715   MG=MG+10.0:IF MG<125.0 THEN NG=125.0
716   DRAW O,P MG,NG 5:O=MG:P=NG
720   SOUND OFF :X=150.0:Y=80.0:C=5.0:F=1.5
721   A$="GREEN !":GOSUB 40040:GOTO 490
1000  GOTO 1000
40012 DIM CAR$(90.0)
40021 FOR Z=32.0 TO 90.0:READ A$
40022 IF A$="STOP" THEN RETURN
40023 READ CAR$(Z):NEXT:RETURN
40040 X1=X:IF F=0.0 THEN F=1.0
40041 FOR M=0.0 TO LEN(A$)-1.0
40042 T$=MID$(A$,M,1)
40050 GR$=CAR$(ASC(T$))
40060 FOR N=0.0 TO LEN(GR$)-1.0 STEP 4.0
40065 IF VFLAG=1.0 GOTO 40120
40070 IF MID$(GR$,N,1)="/" THEN X=X+(8.0*F):GOTO 40100
40080 JC1%=X+VAL(MID$(GR$,N,1))*F:JC2%=Y+VAL(MID$(GR$,N+1,1))*F
40081 JC3%=X+VAL(MID$(GR$,N+2,1))*F:JC4%=Y+VAL(MID$(GR$,N+3,1))*F
40082 DRAW JC1%,JC2% JC3%,JC4% C
40090 NEXT N
40100 IF X+8.0*F>=XMAX THEN X=X1:Y=Y-10.0*F
40102 NEXT M
40103 RETURN
40120 IF MID$(GR$,N,1)="/" THEN Y=Y-9.0*F:GOTO 40180
40130 JC5%=X+VAL(MID$(GR$,N+1,1))*F:JC6%=Y-VAL(MID$(GR$,N,1))*F
40131 JC7%=X+VAL(MID$(GR$,N+3,1))*F:JC8%=Y-VAL(MID$(GR$,N+2,1))*F
40132 DRAW JC5%,JC6% JC7%,JC8% C
40140 NEXT N
40180 IF Y-9.0*F<=0.0 THEN Y=Y1:X=X-9.0*F
40190 NEXT M
40200 RETURN
50000 DATA BLANCO,/,UITROEP!,31313337/,QOUTES,25274547/,#
50001 DATA 1353155521274147/,$,12424253244415262656313 7/
50010 DATA %,17271626125641514252/,&,12132131533115511627 3536/,'
50011 DATA 3537/,(,131513311537/
50020 DATA ),3153535555537/,*,125616523137/,+,32361454/,COMMA,2132323
50030 DATA -,1454/,.,31423241/,/,1256/,0,12162141525627471256/
50040 DATA 1,2141313726377/,2,11511233444455564727162 7/,3
50041 DATA 12212141525334561757 4453/,4,414713531447/
50050 DATA 5,12212141525414515171757/,6,21411215144525315373757/,7
50051 DATA 212223561757/,8,21412444274712131516525355 56/
50060 DATA 9,11313153535624541516274 7/,:,33333535/,;,213232333535/,<
50061 DATA 14471441/
50070 DATA =,13531555/,>,21545427/,?,16272747343331313456/,APE,/
50080 DATA A,11155155135315373755/,B,11171747144411415253555 56/,C
50081 DATA 121627474 5621414152/,D,1117114152561747/
50090 DATA E,1117115114441757/,F,111714441757/,G,12162757215151535343
50091 DATA 111714545157/
50100 DATA I,214131372747/,J,122121415257/,K,111713572451/,L,1117115
50110 DATA M,11171735353435575751/,N,111751571652/,O,12162747565221 41
50111 DATA 1117144417475556/
50120 DATA Q,12162747565321313351/,R,11171747565514442451/,S
50121 DATA 12212141525324441516274756/,T 17573137/
50130 DATA U,111721415157/,V,131753571331 53/,W,11175157113333513334
50131 DATA 11121716515257561256 1652/
50140 DATA Y,16175657163434563134/,Z,175712561151/
```

```
51140 DATA 10,0,10,80,20,0,20,80,25,80,30,85,30,85,30,135,30
51141 DATA 135,25,140,25,140,5,140,5 ,140,0,135,0,135,0,85
51150 DATA 0,85,5,80,999
*
```



```
1     GOTO 20
7     GOTO 64000
8     GOTO 64000
9     GOTO 64000
10    GOTO 64000
20    COLORT 8 0 0 8
21    POKE #131,1
22    PRINT CHR$(12)
23    CURSOR 1,20:PRINT "1  CHANGE BACKGROUND COLOUR"
24    CURSOR 31,20:PRINT "6  ANIMATION  / COLORT "
25    CURSOR 1,18:PRINT "2  FLASHING BACKGROUND"
26    CURSOR 31,18:PRINT "7  ................ "
27    CURSOR 1,16:PRINT "3  SCREEN LINE ADDRESS"
28    CURSOR 31,16:PRINT "8  ................ "
29    CURSOR 1,14:PRINT "4  SCREEN CURSOR ADDRESS"
30    CURSOR 31,14:PRINT "9  ................ "
31    CURSOR 1,12:PRINT "5  ANIMATION, COLOURS 1619"
32    CURSOR 30,12:PRINT "10  ................ "
40    CURSOR 30,2:INPUT "WICH PROGRAM ";P$:PRINT
41    IF P$="1" OR P$="2" OR P$="3" OR P$="4" THEN 46
42    IF P$="5" OR P$="6" THEN 46
43    IF P$="7" OR P$="8" OR P$="9" OR P$="10" THEN 64000
44    CURSOR 1,4:PRINT "WRONG INPUT ONLY THE NUMBER OF THE PROGRAM
45    CURSOR 30,2:PRINT "WICH PROGRAM                ":GOTO 40
46    P=VAL(P$)
47    ON P GOTO 100,1000,2000,3000,4000,10000,7,8,9,10
100   PRINT CHR$(12):PRINT :PRINT :PRINT
108   LIST 110-170
110   E%=#FF
115   COLORT 0 9 9 0
120   B%=#7FEF
125   FOR A%=0 TO 23
130   D%=B%-3
135   FOR C%=0 TO 65
140   POKE D%,E%
145   D%=D%-2:NEXT
146   RJ%=GETC:IF RJ%=32 GOTO 20
155   B%=B%-#86:NEXT
165   E%= INOT E% IAND #FF
170   GOTO 120
1000  PRINT CHR$(12):A5%=0
1010  FOR A%=0 TO 10
1020  POKE #79E4+2*A%,#FF
1025  POKE #79E4+2*A%+#86,#FF
1030  NEXT
1035  CURSOR 23,12:PRINT "WARNING"
1040  FOR B%=20 TO 1 STEP -1
1043  GOSUB 1200
1045  COLORT 0 9 A5% 15-A5%
1046  GOSUB 1100
1050  WAIT TIME B%
1055  COLORT 0 9 15-A5% A5%
1056  GOSUB 1100
1060  WAIT TIME B%
1065  NEXT
1070  GOTO 1040
1100  RJ%=GETC:IF RJ%<>32 THEN RETURN
```

```
1130  PRINT :INPUT "LIST PROGRAM < Y/N > ";RJ$
1140  IF RJ$="Y" THEN PRINT CHR$(12):GOSUB 64500:GOTO 20
1141  IF RJ$="N" THEN PRINT CHR$(12):PRINT :GOTO 20
1145  CURSOR 0,10:PRINT SPC(30):CURSOR 0,11
1150  RETURN
1200  A5%=A5%+1:IF A5%>15 THEN A5%=0
1210  RETURN
2000  GOSUB 2100
2020  FOR A%=0 TO 23
2035  PRINT 23.0-A%;SPC(9-CURX);"# ";HEX$(#7FEA-(#86*A%));
2036  PRINT SPC(22-CURX);"# ";HEX$(#7FED-(#86*A%));SPC(37-CURX);
2040  PRINT "# ";HEX$(#7F6A-(#86*A%));
2041  PRINT SPC(52-CURX);"# "+HEX$(#7F6D-(#86*A%))
2045  IF A%=11 THEN GOSUB 2150:GOSUB 2100
2050  NEXT:PRINT :GOSUB 2150:GOTO 20
2100  PRINT CHR$(12):PRINT
2105  PRINT "        # LOCATION                  # LOCATION"
2110  PRINT "LINE    COLOR CODE   # LOCATION";
2111  PRINT "    COLOR CODE   # LOCATION"
2120  PRINT "NUMBER  BEGIN LINE   BEGIN LINE";
2121  PRINT "    END LINE     END LINE"
2125  PRINT
2130  RETURN
2150  RJ%=GETC:IF RJ%<>32 GOTO 2150
2160  RETURN
3000  PRINT CHR$(12):PRINT :PRINT "CHARACTERS FROM  <-2 TO 61 > "
3002  PRINT "LINES FROM       < 0 TO 23 > ":PRINT
3003  PRINT "INPUT CURSOR EXAMPLE 31,12 FOR CENTER OF SCREEN":PRINT
3004  INPUT "INPUT CURSOR ";B1%,A1%:PRINT :PRINT
3005  IF A1%<0.0 OR B1%>61.0 OR A1%>23.0 THEN PRINT "WRONG INPUT":PRINT :GOTO
3009  B1%=B1%+3
3010  PRINT "POKE # ";HEX$((#7FEA-(#86*(23-A1%)))-((B1%*2)));" TO CHANGE COLO
3020  PRINT "POKE # ";HEX$((#7FED-(#86*(23-A1%)))-((B1%*2)));" TO CHANGE CHAR
3030  PRINT :PRINT
3035  PRINT "FOR OTHERS PRESS RETURN ,FOR OTHER PROGRAMS SPACE BAR"
3040  RJ%=GETC:IF RJ%=32 GOTO 20
3045  IF RJ%=0 GOTO 3040
3050  GOTO 3004
4000  MODE 4
4110  FOR B=0.0 TO 2.0*PI STEP 0.2
4120  A=B-0.2:B%=16:GOSUB 4220
4130  A=B:B%=17:GOSUB 4220
4140  COLORG 0 10 0 10
4150  A=B-0.1:B%=18:GOSUB 4220
4160  A=B+0.1:B%=19:GOSUB 4220
4170  COLORG 0 0 10 10
4180  NEXT
4190  A=B-0.2:B%=16:GOSUB 4220
4200  A=B-0.1:B%=18:GOSUB 4220
4210  GOTO 4110
4220  X%=XMAX/2+30*SIN(A)
4230  Y%=YMAX/2+30*COS(A)
4240  DRAW XMAX/2,YMAX/2 X%,Y% B%
4245  RJ%=GETC:IF RJ%=32.0 THEN MODE 0:GOTO 20
4250  RETURN
10000 MODE 0:COLORT 8 0 0 8
10010 PRINT CHR$(12.0)
10020 A%=#7A28-2:B%=#79A8+2
10030 FOR C%=A% TO B% STEP -2
10040 POKE C%,#FF
```



```
10041 REM POKE C-2,#FF
10042 WAIT TIME 1:POKE C%+2,#0
10050 NEXT:POKE C%,#0
10060 FOR C%=B% TO A% STEP 2
10070 POKE C%,#FF:POKE C%-2,#0
10080 NEXT:POKE C%,#0
10090 JCC%=GETC:IF JCC%>0 GOTO 1
10100 GOTO 10030
64000 P%=P
64005 CURSOR 1,4:PRINT "
64006 PRINT "                    "
64010 CURSOR 1,4:PRINT "NO PROGRAM IN";P%
64020 GOTO 45
64500 PRINT :LIST 1000-1070:GOSUB 2150:RETURN
*
```

B I O R Y T H M

==========

```
90    CLEAR 1000
95    PRINT CHR$(12)
100   DIM X$(31.0):DIM M$(12.0)
110   M$(1.0)="JAN"
111   M$(2.0)="FEB"
112   M$(3.0)="MAR"
113   M$(4.0)="APR"
114   M$(5.0)="MAI"
115   M$(6.0)="JUN"
116   M$(7.0)="JUL"
117   M$(8.0)="AUG"
118   M$(9.0)="SEP"
119   M$(11.0)="NOV"
120   M$(12.0)="DEC"
121   M$(10.0)="OCT"
200   P9=6.28318
210   P1=23.0:P2=28.0:P3=33.0
220   D1=P9/P1:D2=P9/P2:D3=P9/P3
230   DATA 31,28,31,30,31,30,31,31,30,31,30,31
300   INPUT "YOUR NAME PLEASE ";N$
311   PRINT
312   PRINT "BIORYTHM OF YEAR OR MONTH ";
313   INPUT X$
320   IF X$<>"YEAR" AND X$<>"MONTH" THEN GOTO 311
330   N1=0.0
340   GOSUB 8000
360   IF B1>2.0 THEN GOTO 400
370   IF B1=2.0 THEN IF B2=29.0 THEN GOTO 400
380   R=(B3-1900.0)/4.0
381   IF INT(R)<>R THEN GOTO 400
390   N1=1.0
400   GOSUB 8500
420   FOR J=1.0 TO B1
430   READ X
440   NEXT J
450   N1=N1+X-B2
460   IF B1=12.0 THEN GOTO 510
470   FOR J=B1+1.0 TO 12.0
480   READ X
490   N1=N1+X
500   NEXT J
510   IF C3-B3<2.0 THEN GOTO 560
520   FOR J=B3-1899.0 TO C3-1901.0
530   IF INT(J/4.0)=J/4.0 THEN N1=N1+1.0
540   N1=N1+365.0
550   NEXT J
560   RESTORE
570   IF C1=1.0 THEN GOTO 620
580   FOR J=1.0 TO C1-1.0
590   READ X
600   N1=N1+X
610   NEXT J
620   T=(C3-1900.0)/4.0
621   IF INT(T)<>T THEN GOTO 640
630   IF C1>2.0 THEN N1=N1+1.0
640   I1=N1:I2=N1:I3=N1
```



```
650   READ X
655   PRINT CHR$(12)
660   PRINT "  BIORYTHMIC CHART  ";N$
665   PRINT :PRINT
667   B2%=B2:B1%=B1:B3%=B3
670   PRINT "DATE OF BIRTH";B2%;" ";B1%;" ";B3%
680   PRINT :PRINT :PRINT
690   PRINT "I=INTELLIGENCE"
700   PRINT "P=PHYSICAL"
710   PRINT "E=EMOTIONNAL"
720   L=0.0
730   GOSUB 2000
740   D=0.0
745   L=L+1.0
750   FOR I=1.0 TO 31.0
760   X$(I)=" "
770   NEXT I
780   X$(16.0)=":"
800   V1=INT(15.0*SIN((L+I1)*D1)+16.5)
810   V2=INT(15.0*SIN((L+I2)*D2)+16.5)
820   V3=INT(15.0*SIN((L+I3)*D3)+16.5)
830   X$(V1)="P"
840   X$(V2)="E"
850   X$(V3)="I"
860   IF V1=V2 THEN X$(V1)="*"
870   IF V2=V3 THEN X$(V3)="*"
880   IF V1=V3 THEN X$(V1)="*"
890   D=D+1.0
900   IF D<X+1.0 THEN GOTO 1020
910   S1=S1+1.0
920   IF S1=12.0 THEN GOTO 1500
930   C1=C1+1.0
940   IF C1>12.0 THEN GOTO 980
950   READ X
955   IF X9=1.0 THEN GOTO 1500
960   GOSUB 3000
970   GOTO 1020
980   RESTORE
990   C1=1.0
1000  C3=C3+1.0
1010  GOTO 950
1020  D%=D
1021  IF D<10.0 THEN 1023
1022  PRINT M$(C1);" ";D%;"    ";:GOTO 1025
1023  PRINT M$(C1);"  ";D%;"    ";
1025  V$=" "
1030  FOR J=1.0 TO 31.0
1050  V$=V$+X$(J)
1055  NEXT J
```

```
1056  PRINT V$
1060  GOTO 745
1500  STOP
2000  IF X$="MONTH" THEN X9=1.0
2020  PRINT :PRINT " BIORYTHMIC CHART OF ";N$;:C3%=C3
2022  PRINT " FOR ";M$(C1);"  ";C3%
2030  PRINT
2040  PRINT "                    ";"(-)";
2045  PRINT "                    ";"(+)"
2050  PRINT
2060  D=1.0
2070  RETURN
3000  IF X$="MONTH" THEN X9=1.0
3002  PRINT
3004  D=1.0
3010  RETURN
8000  PRINT :PRINT "MONTH,DAY,YEAR OF BIRTH"
8002  PRINT "EXAMPLE   BIRTH ON 30 MAY 1942"
8003  PRINT "PRESS 5  RETURN 3  RETURN 1942"
8015  INPUT B1,B2,B3
8020  RETURN
8500  PRINT
8501  PRINT " GIVE MONTH OND YEAR FOR THE BIORYTHM"
8502  PRINT "EX FOR AND STARTING ON JANUARY   1980"
8503  PRINT "PRESS 1 RETURN 1980 RETURN"
8508  INPUT C1,C3
8510  IF B3>=C3 THEN GOTO 90
8520  RETURN
*
```



```
1     MODE 3A:BST=0.0:CNT=0.0
2     CURSOR 0,3:PRINT "       LAST PLAY";
3     CURSOR 40,3:PRINT "BEST RESULT";
4     GOSUB 5000
10    REM CLEAR 1000
15    ENVELOPE 0 3,10:3,10:3,10:0
20    DIM A(4,0):DIM B(4,0)
70    A(1,0)=40.0:B(1,0)=40.0:A(2,0)=70.0
75    B(2,0)=70.0:A(3,0)=100.0:B(3,0)=40.0
77    A(4,0)=70.0:B(4,0)=10.0
40    DIM TUNE(100,0)
70    DIM NOTE(4,0)
90    NOTE(4,0)=262.0:NOTE(1,0)=330.0:NOTE(3,0)=392.0:NOTE(2,0)=523
100   DIM COLOR(4,0)
110   COLOR(1,0)=1.0:COLOR(2,0)=5.0:COLOR(3,0)=7.0:COLOR(4,0)=11.0
450   CNT=0.0
460   CNT=CNT+1.0
490   TUNE(CNT)=INT(RND(4.0))+1.0
500   WAIT TIME 30
520   FOR I=1.0 TO CNT
530   PLAY=TUNE(I)
540   GOSUB 2000
560   NEXT I
590   I=0.0
600   I=I+1.0
610   IF I<=CNT THEN 635
620   GOTO 490
630   GOSUB 5000
640   GOSUB 2000
645   IF BST<CNT THEN BST=CNT
650   IF PLAY=TUNE(I) THEN 600
670   GOSUB 5000
760   CURSOR 22,2:PRINT "PLAY BROKEN";:WAIT TIME 75
761   CURSOR 22,2:PRINT "                ";:CURSOR 44,2
770   IF BST>CNT THEN GOSUB 5010
771   GOTO 10
2000  SOUND 0 0 10 0 FREQ(NOTE(PLAY))
2020  SOUND 2 0 10 2 FREQ(NOTE(PLAY)*4.0)
2040  FILL A(PLAY),B(PLAY) A(PLAY)+20.0,B(PLAY)+20.0 COLOR(PLAY)
3000  WAIT TIME 20
3050  SOUND OFF
4040  FILL A(PLAY),B(PLAY) A(PLAY)+20.0,B(PLAY)+20.0 0
4100  RETURN
5000  CURSOR 10,2:CNT%=CNT:PRINT CNT%;:PRINT "   ";
5010  CURSOR 44,2:BST%=BST:PRINT BST%;:PRINT "
5015  CURSOR 44,2
5020  RETURN
5999  WAIT TIME 5:G=GETC:IF G=0.0 GOTO 6000
6050  IF G=18.0 THEN PLAY=1.0
6060  IF G=16.0 THEN PLAY=2.0
6070  IF G=19.0 THEN PLAY=3.0
6080  IF G=17.0 THEN PLAY=4.0
6120  RETURN
```

PADDLE SOUND

```
1     REM MAKE SOUND WITH BOTH PADDLES
5     ENVELOPE 0 16
10    P=PDL(0):Q=PDL(2):R=PDL(3)
20    IF P>3.0 OR Q>31.0 THEN SOUND 1 0 R*3/52 0 FREQ(P*12.0+Q)
40    S=PDL(1):T=PDL(4):U=PDL(5)
50    IF S>3.0 OR T>31 THEN SOUND 2 0 U*3/52 0 FREQ(S*12.0+T)
90    GOTO 10
```

*

RANDOM POS TEST

```
1     MODE 0
2     COLOUR 7 0 15 4
4     INPUT "TYPE H OR S . FOR HARDWARE OR SOFTWARE":RNT$
6     W%=1
7     MODE 4
10    DIM A%(XMAX)
15    IF RNT$="S" THEN K=RND(XMAX+1.0):GOTO 21
16    IF RNT$="H" THEN K=RND(0.0)*(XMAX+1.0):GOTO 21
20    GOTO 4
21    R=R+K
22    S%=S%+1
30    A%(K)=A%(K)+1.0
40    Q%=A%(K)
50    P%=Q%/W%
60    IF P%*W%<>Q% THEN 20
69    IF P%>=YMAX%+1 THEN DOT XMAX,0 14:GOTO 69
70    DOT K,P% 15
75    DOT T%,0 7
80    T%=(R/S%-((XMAX+1)*0.495))*100
91    IF T%<0 THEN T%=0
92    IF T%>XMAX% THEN T%=XMAX
93    DOT T%,0 0
999   GOTO 15
```


LANDSCAPE V2

```
5     ENVELOPE 0 5,10:2,5:4,15:0
6     ENVELOPE 1 10,5:15,2:5,3:0
10    MODE 5:FLAG%=0
20    FILL 0,0 XMAX,50 5
30    FILL 0,50 XMAX,YMAX 12
50    DRAW 0,0 150,50 0
60    DRAW 150,50 XMAX,0 0
70    FOR X=0.0 TO 2.0*PI STEP 0.1
80    DRAW 250,150 250+30*COS(X),150+30*SIN(X) 14
90    NEXT
95    GOSUB 1000
165   NOISE 1 15
166   WAIT TIME 3
170   FILL A,50 A+10,60 0
180   FILL A,50 A+1,60 12
185   NOISE 1 15
190   FILL A+10,50 A+11,60 0
195   IF A>50.0 GOTO 210
200   A=A+1.0:GOTO 165
210   FOR X=0.0 TO PI STEP 5E-2
220   DOT 150+50*COS(X),50+50*SIN(X) 0
225   SOUND 1 0 10 0 FREQ(X*100.0+31.0)
230   NEXT
240   A=150.0:B=150.0:C=50.0
250   FILL A,50 B,C 11
260   A=A-1.0:B=B+1.0:C=C+1.0
270   IF A<120.0 GOTO 300
280   GOTO 250
290   SOUND 1 0 15 2 FREQ(2000.0)
310   WAIT TIME 5
320   SOUND 1 0 10 2 FREQ(31.0)
325   NOISE 1 15
330   WAIT TIME 1
340   SOUND 1 0 15 2 FREQ(330.0)
350   SOUND 0 0 15 2 FREQ(440.0)
355   SOUND 2 0 15 2 FREQ(523.0)
360   WAIT TIME 100
370   SOUND 0 0 15 2 FREQ(370.0)
380   WAIT TIME 100
390   SOUND 0 0 15 2 FREQ(415.0)
400   SOUND 2 0 15 2 FREQ(494.0)
450   WAIT TIME 50
500   SOUND 1 0 15 2 FREQ(1318.0)
515   WAIT TIME 100
516   SOUND OFF
520   SOUND 1 0 10 0 FREQ(247.0)
530   WAIT TIME 13
```

```
540   SOUND 1,0,10,0,FREQ(277.0)
550   WAIT TIME 20
560   SOUND 1,0,10,0,FREQ(247.0)
570   WAIT TIME 13
580   SOUND 1,0,10,0,FREQ(208.0)
585   SOUND 1,0,5,0,FREQ(165.0)
590   WAIT TIME 20:SOUND OFF
610   FOR M=0.0 TO 200.0
620   DOT RND(XMAX),(50+RND(YMAX-50.0)) 15
621   NOISE 0,10
625   SOUND 1,0,1,0,FREQ(RND(1000.0)+31.0):WAIT TIME 1:SOUND OFF
626   NOISE OFF
630   NEXT
650   FLAG9%=1
1000  FOR M=0.0 TO 100.0
1100  DRAW 50+A,100 55+A,95,0
1110  DRAW 55+A,95 60+A,100,0
1120  DRAW 50+A,100 55+A,95,12
1130  DRAW 55+A,95 60+A,100,12
1140  DRAW 50+A,95 60+A,95,0
1150  DRAW 50+A,95 60+A,95,12:A=RND(50.0)
1155  SOUND 1,0,3,3,FREQ(3000.0+RND(1000.0))
1156  WAIT TIME 1:SOUND OFF
1159  NEXT M
1170  IF FLAG9%=1 GOTO 1000
1200  RETURN
*
```

*

POLYGONS

```
1     CLEAR 5000
5     INPUT "How many sides ";N
5     PRINT :INPUT "Radius  (between 4 and 120) ";R
10    MODE 5
50    DIM B(N),C(N)
90    P1=2.0*PI/N
100   FOR I=1.0 TO N
110   B(I)=R+10.0+R*COS((I-1.0)*P1)
120   C(I)=R+10.0+R*SIN((I-1.0)*P1)
130   NEXT I
140   FOR I=1.0 TO N
150   FOR J=1.0 TO N
160   DRAW B(I),C(I) B(J),C(J) 15
170   NEXT J:NEXT I
180   WAIT TIME 100:GOTO 5
*
```


MUSIC U2

```
5     DIM F(20.0)
6     ENVELOPE 0,15,3,7,5,13,10,0
10    FOR N=1.0 TO 17.0:READ F(N):NEXT
15    FOR JCC%=1 TO 27
20    READ N,L
30    A=F(N):GOSUB 100:WAIT TIME L
35    NEXT
41    RESTORE:GOTO 10
100   SOUND 0,0,15,0,FREQ(A)
200   SOUND 1,0,15,0,FREQ(A*2.0)
300   SOUND 2,0,10,0,FREQ(A*4.0)
301   RETURN
1000  DATA 262,277,294,311,330,349,370,392,415,440,466
1005  DATA 494,523,554,587,622,659
1010  DATA 1,5,5,5,5,5,13,10,12,5,13,5,15,5,17,10,13,5
1020  DATA 8,5,5,5,1,10,17,10,13,10,9,10,5,10,1,10,1,1
1030  DATA 4,1,10,1,14,1,1,2,3,4,5,6,7,8,9,10,5,13,8
*
```

VIENNA U2

```
2     ENVELOPE 0,1,5,2,5,3,5,0
3     ENVELOPE 1,5,3,3,3,1,3,1
5     DIM F(20.0)
10    FOR N=1.0 TO 17.0:READ F(N):NEXT
15    DATA 262,277,294,311,330,349,370,392
16    DATA 415,440,466,494,523,554,587,622,659
17    FOR JCC%=1 TO 18
20    READ O,E,V,M,N,L
40    SOUND O,E,V,M,FREQ(F(N)):WAIT TIME L
45    NEXT
50    RESTORE:GOTO 10
100   DATA 0,0, 5,0, 7,0,1,0, 5,0, 4,50
110   DATA 0,0, 7,2, 8,0,1,0, 7,2, 5,20
120   DATA 0,0,10,2,17,0,1,0,10,2,13,80
130   DATA 0,0, 5,0,12,0,1,0, 5,0, 9,20
140   DATA 0,0, 7,0,13,0,1,0, 7,0,10,10
150   DATA 0,0,10,0,13,0,1,0 ,7,0,10,80
160   DATA 0,0,10,0,12,0,1,0,10,0, 9,20
170   DATA 0,0,12,0,13,0,1,0,12,0,10,10
180   DATA 0,0,15,0, 8,0,1,1,15,2, 5,30
```

```
* * * * * * M U S I C   T U T O R * * * * * *

          ====================


THIS PROGRAM GENERATES MUSIC  AND DISPLAYS THE NOTES.
IF YOU ANSWER YES BY TYPING  Y TO THE FIRST QUESTION,
THE ONLY KEYS YOU CAN PRESS ARE THE A TO F <DO TO SI>
AND IF YOU ANSWER NO BY TYPING N  ALL ALPHABETIC KEYS
ARE GIVING A NOTE.     YOU CAN ALSO DISPLAY THE NOTES
LARGE OR SMALL SCALE BY TYPING L OR S TO THE QUESTION
BUT YOU NEED A 48K RAM FOR THE SMALL SCALE.

THE NUMERIC KEYS HAVE THE FOLLOWING FUNCTIONS:


1= NORMAL NOTES
2= TREMOLO
3= GLISSANDO
4= GLISSANDO+TREMOLO
5= SHORT NOTES
7= START RECORDING UP TO 2000 NOTES
8= ENDS RECORDING AND REPLAYS EACH TIME YOU PRESS IT
9= SCROLLS PAGE
10=CLEARS  PAGE
SHIFT+ALPHA KEY=INVERT NOTES
TAB KEY RESTART THE PROGRAM
```

```
1     CLEAR 10000:LIMIT%=10:DIM ARRAY%(LIMIT%,200,0)
2     PAGE%=0:POINTER%=0:RECORD%=0:PLAYBACK%=0:TUTOR%=0:ACCENT%=0
3     PRINT CHR$(12):PRINT :PRINT :PRINT "TUTOR MODE  YES OR NO  < Y / N >"
4     ANS%=GET%:IF ANS%=0 GOTO 4
5     IF ANS%=ASC("Y") THEN TUTOR%=1:GOTO 7
6     IF ANS%<>ASC("N") GOTO 1
7     PRINT :PRINT "SIZE - LARGE OR SMALL. < L / S >"
8     ANS%=GET%:IF ANS%=0 GOTO 8
9     IF ANS%=ASC("L") THEN MODE 3:GOTO 15
10    IF ANS%=ASC("S") THEN MODE 5:GOTO 15
11    PRINT "ANSWER ONLY WITH ''S'' OR ''L''":GOTO 7
15    ENVELOPE 0 15,100:0.75:3.50:0:ENVELOPE 1 15,3:10,2:0:STYLE%=0
17    RESTORE:DIM NOTE(21,0,2,0),COMP%(21,0,1,0),SPOT%(21,0)
18    FOR I%=1 TO 13:FOR J%=0 TO 1:READ COMP%(I%,J%):NEXT J%
19    NOTE(I%,0,0)=FREQ(267.0*(2.0^(I%/12.0)))
21    NOTE(I%,1,0)=2.0*NOTE(I%,0,0):NOTE(I%,2,0)=NOTE(I%,0,0)/2.0:NEXT I%
22    FOR I%=14 TO 21:FOR J%=0 TO 1:READ COMP%(I%,J%):NEXT J%:FOR J%=0 TO 2
23    READ CHORD%:NOTE(I%,J%)=NOTE(CHORD%,0,0):NEXT J%:NEXT I%
24    FOR I%=1 TO 21:READ SPOT%(I%):NEXT I%
25    GOSUB 1500
29    FOR TIMER%=1 TO 100-99*ACCENT%
30    GOSUB 10000:IF KEY%=0.0 THEN NEXT TIMER%:SOUND OFF :GOTO 28
31    IF KEY%=53.0 THEN ACCENT%=0:GOTO 30
32    IF KEY%=54 THEN ACCENT%=1:GOTO 30
33    IF KEY%=48 THEN GOSUB 2000:GOTO 30
34    IF (KEY%=57) OR (WHERE=(-1)) THEN OFFSET=OFFSET-75.0:GOSUB 2010:GOTO 3
35    IF KEY%=9.0 THEN SOUND OFF :MODE 0:GOTO 3
36    IF (KEY%>48.0) AND (KEY%<53.0) THEN STYLE%=KEY%-49:GOTO 30
37    OCTAVE%=1:IF (KEY%>96) OR (KEY%=60) THEN OCTAVE%=2:GOSUB 3000
```



```
38    FOR I%=1+13*TUTOR%*(1-ACCENT%) TO 21
39    IF KEY%<>COMP%(I%,TUTOR%) THEN NEXT J%:GOSUB 3500:GOTO 28
40    FOR I%=0 TO 2
41    SOUND I%,ACCENT%,15-10*SGN(I%),STYLE%,NOTE(J%,I%)/OCTAVE%:NEXT I%
42    IF (SPOT%(J%)=100.0) OR (WHERE=(-1.0)) OR (OFFSET<0.0) GOTO 100
48    GOSUB 4000
50    FILL AA,BB CC,DD EE
55    DRAW FF,GG HH,II JJ
60    WHERE=WHERE+10.0:IF WHERE>XMAX-10.0 THEN WHERE=-1.0
100   GOTO 28
1000  DATA 90,67,83,67,88,68,68,67,67,69,86,70,71,67,66,71,72,67,78,65
1010  DATA 74,67,77,66,44,99,87,67,1,5,8,69,68,3,8,1,82,69,5,1,8,84,79
1015  DATA 6,10,13,89,,71,8,1,5,85,65,10,1,6,73,66,12,3,8,79,99,13,5,8
1020  DATA -10,100,-5,100,0,5,100,10,100,15,100,20,25,-10,-5,0,5,10,15,20,
1500  OFFSET=YMAX-62.0:GOTO 2020
2000  FILL 0,0 XMAX,YMAX 0:GOTO 1500
2010  IF OFFSET<0 GOTO 1500
2020  WHERE=5.0
2030  FILL 0,OFFSET-12 XMAX,OFFSET+62 0
2040  FOR Z%=OFFSET TO OFFSET+40 STEP 10
2050  DRAW 0,Z% XMAX,Z% 12:NEXT Z%:RETURN
3000  KEY%=KEY%-32:IF KEY%=28 THEN KEY%=44
3010  RETURN
3500  TIMER%=TIMER%+1:NEXT TIMER%:SOUND OFF
3510  RETURN
4000  AA=WHERE-2.0:BB=OFFSET+(OCTAVE%-1.0)*35.0+SPOT%(J%)-2.0
4010  CC=WHERE+2.0:DD=OFFSET+(OCTAVE%-1.0)*35.0+SPOT%(J%)+2.0
4020  EE=SPOT%(J%)/5.0+8.0
4030  FF=WHERE+6.0-4.0*OCTAVE%:GG=OFFSET+SPOT%(J%)+(OCTAVE%-1.0)*35.0
4040  HH=WHERE+6.0-4.0*OCTAVE%:II=OFFSET+SPOT%(J%)+20.0:JJ=SPOT%(J%)/5.0+8
4050  RETURN
5000  IF KEY%=56 THEN RECORD%=0:ARRAY%(PAGE%,POINTER%)=128
5010  RETURN
6000  IF POINTER%=200 THEN POINTER%=0:PAGE%=PAGE%+1:GOSUB 7000
6010  RETURN
7000  IF PAGE%>LIMIT% THEN PAGE%=LIMIT%:RECORD%=0:PLAYBACK%=0
7010  RETURN
10000 KEY%=GET%:IF KEY%=55 THEN GOTO 30000
10002 IF (KEY%=56) AND (RECORD%=0) THEN PLAYBACK%=1:POINTER%=0:PAGE%=0
10005 IF RECORD%=1 THEN ARRAY%(PAGE%,POINTER%)=KEY%:GOSUB 5000
10010 IF PLAYBACK%=1 THEN KEY%=ARRAY%(PAGE%,POINTER%)
10015 IF (RECORD%=1.0) OR (PLAYBACK%=1.0) THEN POINTER%=POINTER%+1:GOSUB 6
10020 IF KEY%=128 THEN PLAYBACK%=0
10030 RETURN
30000 RECORD%=1:PLAYBACK%=0:POINTER%=0:PAGE%=0
30010 KEY%=GET%:IF KEY%=0 GOTO 30010
30020 GOTO 10002
*
```

```
5     CLEAR 5000
10    MODE 6
15    DIM A(250.0),B(250.0)
20    COLORG 8 0 15 3
30    FOR X=0.0 TO 2.0*PI STEP 3E-2
40    A(N)=XMAX/2.0+100.0*COS(X):B(N)=YMAX/2.0+100.0*SIN(X*2.0)
45    N=N+1.0
50    NEXT
90    COLORG 8 0 15 3
100   FOR X=0.0 TO 209.0
110   DRAW 150.125 A(X),B(X) 0
115   DRAW 0.0 A(X),B(X) 3
116   DRAW A(X),B(X) XMAX,0 15
120   NEXT
300   FOR X=0.0 TO 50.0
320   COLORG 0 A 0 0
330   WAIT TIME 15
335   COLORG 0 0 A 0
337   WAIT TIME 15
338   COLORG 0 0 0 A
339   WAIT TIME 15
340   A=A+1.0:IF A=16.0 THEN A=1.0
345   NEXT X
400   FOR X=0.0 TO 50.0
410   COLORG RND(15.0) RND(15.0) RND(15.0) RND(15.0)
420   WAIT TIME 20
430   NEXT X
450   GOTO 90
```



```
1     MODE 0:PRINT CHR$(12):PRINT :PRINT
2     PRINT "..............TOWER OF HANOI..................."
3     PRINT :PRINT
4     PRINT "AN EXAMPLE OF ANIMATED GRAPHIC CAPABILITIES OF THE"
5     PRINT :PRINT "           D A I PERSONAL COMPUTER"
6     PRINT :PRINT :PRINT :PRINT "DO YOU WANT INSTRUCTIONS"
7     PRINT :PRINT "ANSWER YES OR NO  ":INPUT A$
8     IF A$="YES" GOTO 10:IF A$="NO" GOTO 20
9     PRINT CHR$(12):PRINT :PRINT "ANSWER ONLY YES OR NO":GOTO 2
10    PRINT CHR$(12):PRINT :PRINT
11    PRINT "          TOWER OF HANOI":PRINT :PRINT :PRINT
12    PRINT "YOU HAVE TO MOVE ALL HORIZONTAL BARS FROM COLUMN 1 TO"
13    PRINT "COLUMN 3 WITHOUT PLACING A LARGER BAR ABOVE A SMALLER"
14    PRINT "BAR.     FOR MOVING THE BAR YOU PRESS ON 1 , 2  OR  3"
15    PRINT "GIVING  THE NUMBER  OF THE COLUMN  FROM WHERE THE BAR"
16    PRINT "HAS  TO LEAVE  FOLLOWED  BY THE NUMBER  OF THE COLUMN"
17    PRINT "WHERE THE BAR HAS TO GO":PRINT :PRINT :PRINT
18    PRINT "PRESS ANY KEY TO START THE GAME"
19    T=GETC:IF T=0.0 GOTO 18
20    CLEAR 2000
21    DIM Z@100.@)
22    PRINT CHR$(12)
23    COLORT 7 0 0 0
24    COLORG 7 4 5 1
25    MODE 2A
30    JC1%=0:Y9=48.0:N=9.0:C1=4.0:C2=5.0:C3=1.0:C0=7.0
33    DRAW 0,0 70,0 C1
36    FOR I=1.0 TO 3.0
38    DRAW I*24-12,0 I*24-12,Y9 C2
40    Z(1.0)=0.0:Z(I*10.0)=10.0:NEXT
50    M=1.0:C=C3
60    FOR I=1.0 TO N
70    Z(1.0)=I:Z(10.0+I)=10.0-I
80    GOSUB 900:NEXT
90    GOTO 110
100   PRINT "INVALID MOVE"
110   JC1%=JC1%+1:PRINT "YOUR MOVE  FROM <1,2 OR 3>  ";
111   P=GETC:WAIT TIME 5:IF P=0.0 GOTO 111
112   M1=P-48.0:M1%=M1:PRINT M1%;:PRINT "  TO  ";
113   P=GETC:WAIT TIME 5:IF P=0.0 GOTO 113
114   M2=P-48.0:M2%=M2:PRINT M2%;:PRINT "   ";:PRINT JC1%;:PRINT " MOVE
120   IF M1<>INT(M1) OR M1<1.0 OR M1>3.0 GOTO 100
130   IF M2<>INT(M2) OR M2<1.0 OR M2>3.0 GOTO 100
140   IF M1=M2 OR Z(M1)=0.0 GOTO 100
150   P1=Z(M1)+10.0*M1
160   P2=Z(M2)+10.0*M2
170   IF Z(P1)>Z(P2) GOTO 100
200   M=M1:C=C0:GOSUB 900
210   Z(M2)=Z(M2)+1.0:Z(P2+1.0)=Z(P1)
220   Z(M1)=Z(M1)-1.0
230   M=M2:C=C3:GOSUB 900
240   G=G+1.0
250   IF Z(3.0)<N GOTO 110
300   PRINT "THAT TOOK YOU ",JC1%,"MOVES"
310   STOP
900   X=M*24.0-12.0
910   Y=5.0*Z(M)
920   X1=Z(Z(M)+10.0*M)+2.0
930   DRAW X-X1,Y X-1,Y C
940   DRAW X+1,Y X+X1,Y C
950   RETURN
```

G R A P H I C O F S I N U S

```
1     COLORT 0 15 0 0:PRINT CHR$(12.0):PRINT :PRINT
2     PRINT "THIS PROGRAM DRAW A SINUS WAVE ON THE SCREEN"
3     PRINT :PRINT :PRINT "IF YOUR MACHINE IS AN  8K RAM  YOU MUST CHANGE 6A
4     PRINT "INTO 2A IN LINE 12 AND INTO 4A FOR A  12 K MACHINE"
5     PRINT "THIS IS ACHIEVED BY TYPING EDIT 30 AND PLACING THE"
6     PRINT "CURSOR ON THE ''6'' OF ''6A''WITH THE CURSOR ARROW"
7     PRINT "KEY AND PRESS CHAR DEL KEY AND ''2'' OR ''4''  KEY.":PRINT
8     PRINT :PRINT 'PRESS ANY KEY TO CONTINUE"
9     P=GETC:IF P=0.0 GOTO 9
12    MODE 5A:PRINT CHR$(12):PRINT " FUNCTION  =  A *SINUS B *(X - C)+ D"
13    PRINT "A=? ";
14    P=GETC:IF P=0.0 GOTO 14
15    WAIT TIME 5:A1=P-48.0:A1%=A1:PRINT A1%,"B= ?";
16    P=GETC:IF P=0.0 GOTO 16
17    WAIT TIME 5:A2=P-48.0:A2%=A2:PRINT A2%,"C= ?";
18    P=GETC:IF P=0.0 GOTO 18
19    WAIT TIME 5:A3=P-48.0:A3%=A3:PRINT A3%,"D= ?";
20    P=GETC:IF P=0.0 GOTO 20
21    WAIT TIME 5:A4=P-48.0:A4%=A4:PRINT A4%,
25    WAIT TIME 20:PRINT CHR$(12)
30    COLORG 0 15 5 10
35    PRINT "GRAFIC OF THE FUNCTION :"
40    PRINT A1;"SIN";A2;"(X-";A3;")+";A4
50    D=XMAX/4.0/PI
60    FOR N=0.0 TO XMAX STEP D
65    DRAW N,0 N,YMAX 5
70    NEXT N
75    A4=YMAX/2.0-A4*D
80    FOR M=0.0 TO A4 STEP D
85    DRAW 0,A4-M XMAX,A4-M 5
90    NEXT M
95    FOR M=0.0 TO YMAX-A4 STEP D
100   DRAW 0,A4+M XMAX,A4+M 5
105   NEXT M
115   DRAW 0,A4 XMAX,A4 10
130   FOR X=0.0 TO XMAX
140   DOT X,SIN(A2*(4.0*PI*X/XMAX-A3))*D*A1+YMAX/2.0 15
150   NEXT X
200   PRINT "PRESS ANY KEY TO CONTINUE"
220   W=GETC:WAIT TIME 10:IF W=0.0 GOTO 220:GOTO 12
250   PRINT :PRINT :PRINT :PRINT :PRINT "G R A P H I C  O F  S I N U S":PRI
260   PRINT "==============================":PRINT :PRINT :PRINT
270   LIST
```



```
5     COLORT 12 0 0 0
10    A%=0:B%=0:C%=0:ANS%=0:R%=0:W%=0:POPER%=0:MODE 0
11    GOSUB 3000:GOSUB 3100:GOSUB 3300
20    CURSOR 12,21:PRINT "A R I T H M A T I C  T E A C H E R ";
22    CURSOR 15,19:PRINT "for add press.............1";
24    CURSOR 15,18:PRINT "for subtract press........2";
26    CURSOR 15,17:PRINT "for take-away-add press....3";
28    CURSOR 15,16:PRINT "for multiply press........4";
30    CURSOR 15,15:PRINT "for divide press..........5";
32    CURSOR 15,14:PRINT "for multiply-divide press...6";
34    CURSOR 20,12:PRINT "SELECT YOUR CHOICE";
36    CURSOR 28,10:PRINT "?"::CURSOR 28,10
50    CR%=GETC
51    CR%=GETC:IF CR%=0 THEN 51
52    IF CR%=49 THEN 100:IF CR%=50 THEN 200:IF CR%=51 THEN 400
54    IF CR%=52 THEN 600:IF CR%=53 THEN 700:IF CR%=54 THEN 800
56    GOTO 50
100   A%=0:B%=0:MODE 0:GOSUB 3300:REM CLEAR TOP OF SCREEN
101   CURSOR 28,21:PRINT "ADD"
102   POPER%=0:E%=0:MODE 0
103   GOSUB 3304
104   XP%=19:YP%=19:CURSOR XP%,YP%:X%=A%:GOSUB 1000
105   XP%=27:CURSOR XP%,YP%:X%=B%:GOSUB 1000
106   XP%=35:CURSOR XP%,YP%:X%=ANS%:GOSUB 1000
107   GOSUB 2500:REM CALCULATE RANDOM NUMBERS
108   C%=A%+B%:XP%=20:YP%=13:CURSOR XP%,YP%+1
110   PRINT A%;"   +  ";B%;"   =  ?";
112   XP%=XP%-1:CURSOR XP%,YP%:X%=A%:GOSUB 1000
114   XP%=XP%+8:CURSOR XP%,YP%:X%=B%:GOSUB 1000
118   CP%=36:GOSUB 2040:GOSUB 2050:REM PRINT R% & W%
120   GOSUB 3000:REM DRAW BASIC FACE
122   IF E%=1 THEN E%=0:GOTO 128
124   GOSUB 3100:REM DRAW REWARD FACE
126   GOTO 130
128   GOSUB 3200:REM DRAW PUNISH FACE
130   CURSOR CP%,14:ANS%=0:DIG%=0
132   GOSUB 1500
134   IF POPER%=1 THEN 10:IF POPER%=2 THEN 102
136   ANS%=CR%-48+ANS%
138   IF ANS%>C% THEN W%=W%+1:GOSUB 2050:GOSUB 3200:E%=1:GOTO 3500
140   IF ANS%<C% AND DIG%>=2.0 THEN W%=W%+1:GOSUB 2050:GOSUB 3200:E%=1:GOTO 3
142   IF ANS%<C% AND DIG%=0.0 THEN PRINT ANS%;:ANS%=ANS%*10:DIG%=DIG%+1:GOTO
143   IF ANS%=C% THEN R%=R%+1:GOSUB 2040:GOTO 146
144   DIG%=DIG%+1:PRINT ANS%;:GOTO 132
146   DIG%=0:CURSOR XP%+9,14:PRINT ANS%;
148   REM X%=ANS%:XP%=XP%+8:CURSOR XP%,YP%:GOSUB 1000
150   WAIT TIME 50:CURSOR 20,14
152   IF E%=1 GOTO 108
154   GOTO 102
200   PRINT "SUBTRACT"
202   GOTO 202
400   A%=0:B%=0:C%=0:MODE 0:GOSUB 3300:REM CLEAR TOP OF SCREEN
401   CURSOR 21,17:PRINT "TAKE-AWAY-ADD";
402   E%=0.0:MODE 0
407   XP%=16:YP%=19:X%=A%:CURSOR XP%,YP%:GOSUB 1000
408   XP%=26:X%=C%:CURSOR XP%,YP%:GOSUB 1000
409   XP%=33:X%=B%:CURSOR XP%,YP%:GOSUB 1000
410   GOSUB 2500:REM CALCULATE RANDOM NUMBERS
```

```
415   C%=A%-B%:XP%=17:YP%=13:CURSOR XP%,YP%+1
420   PRINT A%;"    ?    ?   = ";B%;
425   XP%=XP%-1:CURSOR XP%,YP%:X%=A%:GOSUB 1000
430   XP%=XP%+17:CURSOR XP%,YP%:X%=B%:GOSUB 1000
435   CP%=23:GOSUB 2040:REM PRINT R%
440   GOSUB 2050:REM AND W%
445   GOSUB 3000:REM DRAW BASIC FACE
450   IF E%=1 THEN GOTO 465
455   GOSUB 3100:REM DRAW REWARD FACE
460   GOTO 470
465   E%=0:GOSUB 3200:REM DRAW PUNISH FACE
470   CP%=CP%:CURSOR CP%,14
475   GOSUB 1500
480   IF POPER%=1.0 THEN GOTO 10
485   IF C%=0.0 AND CR%=79.0 THEN PRINT "-"::R%=R%+1:GOSUB 2040:GOTO 525
490   IF C%=0 AND CR%=81 THEN PRINT "+"::R%=R%+1:GOSUB 2040:GOTO 525
495   IF C%>0 AND CR%=79 THEN PRINT "-"::R%=R%+1:GOSUB 2040:GOTO 525
500   IF C%<0.0 AND CR%=81.0 THEN PRINT "+"::R%=R%+1:GOSUB 2040:GOTO 525
505   IF POPER%=2.0 THEN GOTO 400
510   W%=W%+1:E%=1:GOSUB 3200:REM PUNISH FACE
515   CURSOR CP%,14:GOSUB 2050
520   GOTO 475
525   CP%=CP%+5:CURSOR CP%,14
530   GOSUB 1500
535   IF POPER%=1 OR POPER%=2 THEN GOTO 475
540   D%=CR%-48
541   IF D%=ABS(C%) THEN N$=CHR$(CR%):PRINT N$::R%=R%+1:GOSUB 2040:GOTO 56
545   W%=W%+1:GOSUB 3200:REM PUNISH FACE
550   E%=1:GOSUB 2050
555   GOTO 530
560   IF E%=1 THEN MODE 0:GOTO 415
565   C%=VAL(N$):XP%=XP%-7:YP%=YP%:X%=C%:CURSOR XP%,YP%:REM GOSUB 1000
566   WAIT TIME 50
570   CURSOR XP%+7,YP%+1:GOTO 402
600   PRINT "MULTIPLY"
602   GOTO 602
700   PRINT "DIVIDE"
702   GOTO 702
800   PRINT "MULTIPLY-DIVIDE"
802   GOTO 802
1000  REM SUBROUTINE TO PLACE DOMINO DOTS
1001  REM EXPECTS TO HAVE DEFINED BEFORE CALL
1002  REM THE X AND Y CURSOR POSITION OF THE FIRST DOT
1003  REM SPECIFIED BY (XP%) AND (YP%)
1004  REM THE NUMBER OF DOTS TO BE PRINTED
1005  REM SPECIFIED BY (X%)
1009  M%=0
1010  IF X%=0 THEN RETURN
1015  IF X%<0 THEN X%=X%+5:GOTO 1030
1020  IF X%>=5 THEN U%=5:M%=M%+1:GOSUB 1040:CURSOR XP%,YP%-M%:X%=X%-5:GOTO 10
1030  U%=X%:GOSUB 1040:RETURN
1040  FOR P%=1 TO U%:PRINT ".";:NEXT:RETURN
1500  REM ROUTINE TO GET A CHARACTER AND TEST
1501  REM FOR OTHER FUNCTIONS AS TAB AND REPT
1503  REM SETS VARIABLE POPER% TO EQUAL 1
1504  REM WHEN DESIRABLE TO RESELECT A NEW PROGRAM
1510  CR%=GETC
1511  CR%=GETC:IF CR%=0 THEN 1511
1512  IF CR%=19 THEN POPER%=2:R%=0:W%=0:GOSUB 2040:GOSUB 2050:RETURN
1515  IF CR%=16 THEN POPER%=1:RETURN
```



```
1520  RETURN
2000  REM ROUTINES THAT PRINT VALUES OF R% & W%
2001  REM IT RETURNS CURSOR TO POSITION OF CP%
2040  CURSOR 1,3:PRINT R%::CURSOR CP%,14:RETURN
2050  CURSOR 48,3:PRINT W%::CURSOR CP%,14:RETURN
2500  REM CALCULATES TWO RANDOM NUMBERS
2501  REM THEY ARE (A%) AND (B%)
2510  A%=10*RND(1.0):A%=INT(A%)
2520  B%=10.0*RND(1.0):B%=INT(B%)
2530  RETURN
3000  FR%=0:GOSUB 3005:FR%=47:GOSUB 3005
3005  CURSOR FR%+1,12:PRINT "########";
3010  FOR F%=7 TO 11
3020  CURSOR FR%,F%:PRINT "# ^    ^ #";:NEXT
3030  CURSOR FR%+1,6:PRINT "#      #";
3040  CURSOR FR%+2,5:PRINT "######";
3050  CURSOR FR%+2,10:PRINT "o   o";
3060  CURSOR FR%+2,9:PRINT "  *  ";
3061  IF FR%=47.0 THEN CURSOR 49,12:PRINT "^    ^"
3062  CURSOR 16,3:PRINT "PRESS ";CHR$(9);" KEY TO RESET SCORE"
3063  CURSOR 18,1:PRINT "PRESS ";CHR$(94);" KEY TO RESELECT"
3100  FR%=0:GOSUB 3250:FR%=47:GOSUB 3253:RETURN
3200  FR%=0:GOSUB 3253:FR%=47:GOSUB 3250:RETURN
3250  CURSOR FR%+2,8:PRINT "'    '";
3251  CURSOR FR%+2,7:PRINT " ''' ";
3252  RETURN
3253  CURSOR FR%+2,8:PRINT " ''' ";
3254  CURSOR FR%+2,7:PRINT "'    '";
3255  RETURN
3300  CURSOR 0,20:PRINT "                                        ";
3301  PRINT "                                        ";
3302  CURSOR 0,21:PRINT "                                        ";
3303  PRINT "                                        ";
3304  CURSOR 0,22:PRINT "                                        ";
3305  PRINT "                                        ";
3306  CURSOR 0,23:PRINT "                                        ";
3307  PRINT "                                        ";
3308  RETURN
3500  CURSOR 20,14:MODE 0:GOTO 108
*
```

```
A G E N D A

==========



2     CLEAR 15000
5     DIM NAME$(50.0),SURNAME$(50.0),ADRESS$(50.0)
10    PRINT CHR$(12):FOR X1=0.0 TO 59.0
20    PRINT CHR$(1);
30    NEXT X1
40    CURSOR 0,0
50    FOR X2=0.0 TO 59.0
60    PRINT CHR$(1);
70    NEXT X2
90    CURSOR 0,20
100   PRINT "*            This is a  demonstration program
110   PRINT "*            for people who do not know about
120   PRINT "*                       COMPUTER.
130   PRINT "*****************************************************************"
140   GOSUB 10000
160   PRINT CHR$(12)
170   FOR X=0.0 TO 59.0
180   PRINT CHR$(2);
190   NEXT X
195   CURSOR 0,18
200   PRINT "#################################################################
210   PRINT "#
220   PRINT "#  We shall make a list of i.e. 50 persons with                 #"
240   PRINT "#                                                               #"
250   PRINT "#        1) NAME                                                #"
260   PRINT "#        2) SURNAME                                             #"
270   PRINT "#        3) NUMBER                                              #"
280   PRINT "#        4) ADRESS                                              #"
290   PRINT "#                                                               #"
300   PRINT "#################################################################
400   GOSUB 10000
405   PRINT CHR$(12)
410   PRINT "#################################################################"
420   PRINT "# NOTE :- If you type an error press on    !CHAR DEL!            #"
430   PRINT "#        - NEVER press on the reset button                      #"
440   PRINT "#        - Every command to the computer must be                #"
450   PRINT "#          followed by pressing RETURN.                         #"
455   PRINT "#        - When you have typed all the names you wanted         #"
457   PRINT "#          to enter just type HALT and the same if you          #"
459   PRINT "#          want to pass to an other part of the program         #"
460   PRINT "#################################################################
470   GOSUB 10000
500   PRINT CHR$(12)
510   PRINT "================================================================="
520   PRINT "+                    M E N U                                    +"
530   PRINT "+                    -------                                    +"
540   PRINT "+      1) New data base          ->> NEW                        +"
550   PRINT "+      2) Look the data          ->> LOOK                       +"
560   PRINT "+      3) Search ONE of the data ->> SEARCH                     +"
570   PRINT "+      4)                        ->> HALT                       +"
580   PRINT "+                                                               +"
590   PRINT "+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++"
600   PRINT CHR$(13)
```



```
610   DIM OPTIE$(1.0):INPUT "Type now one of those options !";OPTIE$
630   IF OPTIE$="NEW" GOTO 1000
640   IF OPTIE$="LOOK" GOTO 2000
650   IF OPTIE$="SEARCH" GOTO 3000
660   IF OPTIE$="UUL" GOTO 4000
670   IF OPTIE$="HALT" GOTO 5000
680   PRINT
690   PRINT "Please answer only with NEW, LOOK, SEARCH or HALT."
700   GOTO 600
1000  REM ********************** NEW ************************
1010  I%=1
1020  GOSUB 20000
1030  CURSOR 54,20
1040  PRINT I%
1050  CURSOR 8,21
1060  INPUT NAME$(I%)
1070  IF NAME$(I%)="HALT" GOTO 500
1080  CURSOR 12,20
1090  INPUT SURNAME$(I%)
1100  CURSOR 14,19
1110  INPUT ADRESS$(I%)
1120  I%=I%+1
1130  IF I%<=20 GOTO 1020
1140  PRINT "Sorry , but you have filled the data base!!!"
1150  GOSUB 10000
1160  GOTO 500
2000  REM ********************** LOOK************************
2010  I%=1
2020  IF NAME$(I%)="HALT" GOTO 500
2025  GOSUB 20000
2030  CURSOR 54,20
2040  PRINT I%
2050  CURSOR 8,21
2060  PRINT NAME$(I%)
2070  CURSOR 12,20
2080  PRINT SURNAME$(I%)
2090  CURSOR 14,19
2100  PRINT ADRESS$(I%)
2110  GOSUB 10000
2120  I%=I%+1
2130  IF I%<=20.0 GOTO 2020
2140  PRINT CHR$(12):PRINT "You have now looked to the 50 persons  !"
2150  GOSUB 10000
2160  GOTO 500
3000  REM ********************** SEARCH ************************
3005  PRINT CHR$(12)
3010  PRINT " YOU WANT TO SEARCH A PERSON."
3020  PRINT " Which characteristic do you know???"
3030  PRINT "     1)Name       ->>NAME"
3040  PRINT "     2)Surname    ->>SURN"
3050  PRINT "     3)Adress     ->>ADRE"
3060  PRINT "     4)Number     ->>NUMB"
3070  PRINT "     5)None ....  ->>NONE"
3080  PRINT CHR$(13)
3090  DIM KOMMANDO$(1.0):INPUT KOMMANDO$
3100  IF KOMMANDO$="NAME" GOTO 3200
3110  IF KOMMANDO$="SURN" GOTO 3300
3130  IF KOMMANDO$="NUMB" GOTO 3500
3140  IF KOMMANDO$="ADRE" GOTO 3400
3150  IF KOMMANDO$="NONE" GOTO 2010
3160  PRINT :PRINT "Answer  only with NAME,SURN,NUMB,ADRE or NONE!"
```

```
3199  GOTO 3999
3200  REM ---------------------- SEARCH NAME -------------------------
3201  PRINT CHR$(12)
3202  DIM D$(1,0):INPUT "Do you know the name YES or NO ":D$
3203  IF D$="NO" GOTO 3210
3204  IF D$="YES" GOTO 7000
3205  PRINT :PRINT " Answer only with YES or NO .":PRINT :GOTO 3202
3210  PRINT :PRINT " Here follow the list of the names : "
3220  I%=1
3225  IF NAME$(I%)<>"HALT" THEN 3230
3226  GOTO 3260
3230  PRINT I%;"  ";NAME$(I%)
3240  I%=I%+1
3250  IF I%<=20 GOTO 3225
3260  INPUT "Wich number do you want to see";I%
3270  GOTO 3540
3300  REM --------------------- SEARCH SURNAME--------------------
3301  PRINT CHR$(12)
3302  DIM F$(1,0):INPUT " do you know the surname  type YES or NO";F$
3303  IF F$="NO" GOTO 3320
3304  IF F$="YES" GOTO 7100
3305  PRINT :PRINT " Answer please only wit YES or NO !!!":PRINT :GOTO 3302
3320  PRINT " Here follows the list of the surnames : "
3330  I%=1
3340  IF NAME$(I%)<>"HALT" THEN 3360
3345  GOTO 3385
3360  PRINT I%;"  ";SURNAME$(I%)
3370  I%=I%+1
3380  IF I%<=20 GOTO 3340
3385  INPUT "Wich number do you want to see  ";I%
3390  GOTO 3540
3400  REM ------------------ SEARCH ADRESS---------------------
3401  PRINT CHR$(12)
3402  DIM G$(1,0):INPUT " Do you know the adress , type YES or NO";G$
3403  IF G$="NO" GOTO 3420
3404  IF G$="YES" GOTO 7200
3405  PRINT :PRINT " Answer only with YES or NO  ":PRINT :GOTO 3402
3420  PRINT " Hereunder the list of all the adresses : "
3430  I%=1
3440  IF NAME$(I%)<>"HALT" THEN 3460
3445  GOTO 3490
3460  PRINT I%;"  ";ADRESS$(I%)
3470  I%=I%+1
3480  IF I%<=20 GOTO 3440
3490  INPUT " Wich number do you want to see ";I%
3495  GOTO 3540
3500  REM -----------------SEAR NUMBER---------------------------
3510  PRINT CHR$(12)
3520  INPUT " Wich number do you want to see";I%
3540  GOSUB 20000
3545  GOSUB 30000
3570  GOSUB 10000
3580  GOTO 500
4000  REM ********************** FILL **************************
5000  REM ********************** HALT ***************************
7000  REM --------------------- NAME KNOWN-------------------
7010  I%=1:PRINT
7014  DIM GEKEND$(1,0):INPUT "Wich name do you want to see ";GEKEND$
7020  IF NAME$(I%)=GEKEND$ GOTO 7050
7030  I%=I%+1
7040  IF I%<=20 GOTO 7020
```



```
7045  GOTO 500
7050  GOSUB 20000
7060  GOSUB 30000
7070  GOSUB 10000
7080  GOTO 7030
7100  REM ------------------- SURNAME KNOWN-------------
7110  I%=1:PRINT
7114  DIM GEKEND$(1,0):INPUT " Wich surname do you want to see ":GEKEN
7120  IF SURNAME$(I%)=GEKEND$ GOTO 7150
7130  I%=I%+1
7140  IF I%<=20 GOTO 7120
7145  GOTO 500
7150  GOSUB 20000
7160  GOSUB 30000
7170  GOSUB 10000
7180  GOTO 7130
7200  REM ------------------    ADRESS KNOWN-------------
7210  I%=1:PRINT
7214  DIM GEKEND$(1,0):INPUT " Wich adress do you want to see ":GEKEND
7220  IF ADRESS$(I%)=GEKEND$ GOTO 7250
7230  I%=I%+1
7240  IF I%<=20 GOTO 7220
7245  GOTO 500
7250  GOSUB 20000
7260  GOSUB 30000
7270  GOSUB 10000
7280  GOTO 7230
9999  REM ********************** RETURNSUBR ****************
10000 CURSOR 5,3
10010 PRINT "                          ----------"
10020 CURSOR 5,2
10030 PRINT " ***    NOW PRESS ON  ! RETURN !     ***"
10040 CURSOR 5,1
10050 PRINT "                          ----------"
10060 DIM TERUG$(1,0):INPUT TERUG$
10070 RETURN
19999 REM ********************** LABELSUBR ****************
20000 PRINT CHR$(12)
20010 PRINT "*************************************************************
20020 PRINT "* NAME :                                                    *
20030 PRINT "* SURNAME :                                                 *Nr.*
20040 PRINT "* ADRESS :                                                  *
20050 PRINT "*************************************************************
20060 RETURN
30000 REM ****************** PRINT SUBR ********************
30045 CURSOR 54,20:PRINT I%
30050 CURSOR 7,21:PRINT NAME$(I%)
30055 CURSOR 12,20:PRINT SURNAME$(I%)
30060 CURSOR 14,19:PRINT ADRESS$(I%)
30070 RETURN
*
```