AUTUMN '83

SHARPSOFT

Issue No. 9

C O N T E N T S

Don´t forget
your 1984 Subscription Form
at the back of this Issue

SHARPSOFT USER NOTES

ISSUE NO. 9

Once again we reach the end of another publishing year. We would like to thank all our subscribers for their support during 1983. Judging by your comments these User Notes still provide a valuable contribution to the SHARP micro computer scene.

Issue No. 9 is packed with interesting articles, programs and comments from readers. Hisoft PASCAL seems to have been well received by most SHARP users - quite rightly so in my opinion as it is a superb package.

The PASCAL articles in this issue present information on strings, a printer driver for the popular MX80T3 and an improved random number generator. Do keep sending us your PASCAL programming ideas and programs. Programs which demonstrate Hisoft PASCAL's speed, for example arcade games, would be interesting - any takers?

FORTH still generates considerable discussion amongst users interested in this language. Mr. Peter Amey has contributed two important FORTH utilities called SET and RESET whose function is similar to the BASIC SET and RESET statements.

A major item this issue is a follow-up article on the MZ-80K graphics by Mr. John Simpson. John has also agreed to write a further article for inclusion in Issue 10 - next year. Please be reminded that all information and charts contained in these articles is copyright by Mr. J. Simpson.

If you are a machine code programmer then the article on machine language monitors by Mr. Peter Andersson will certainly contain information that interests you. Peter is the author of the popular MZ-80K/A PA-MON package.

Due to the number of important articles in Issue 9 we decided to leave out our CP/M column this issue. Issue 10 will revert to our regular format.

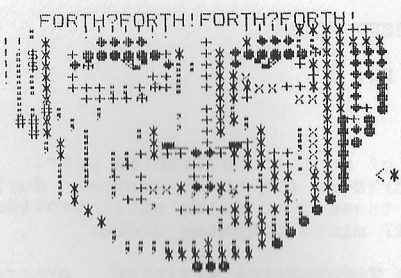
The recent release by SHARP of their MZ-700 range of low cost colour computers is beginning to be widely known amongst SHARP computer users. We have decided to publish a separate set of user notes for this new machine. Adding yet another SHARP system to these User Notes would mean that we would have to reduce the length of articles on the MZ-80K/A/B which we felt was basically unfair to all our existing Members.

With Issue 9 you will find a subscription form for the 1984 User Notes. It helps us tremendously if you could return the form fully completed with your subscription as soon as possible.

Issue 10 of the User Notes will be published towards the end of January or early February 1984.

Best wishes to all our readers
for Christmas and the New Year

MIKE BRINSON
Editor



FORTH

SET and RESET Utilities
by
Peter Amey

On the subject of FORTH I have found recent correspondence and articles in the notes interesting. I have implemented fig-FORTH on my MZ-80K under CP/M and my version is software compatible with yours. If you think there may be sufficient interest I am willing to make it available to members at a very competitive price. I have made a number of small enhancements to the fig model which may interest you. Firstly I redefined EMIT to suppress the MSB of any Byte output. This does away with disconcerting graphics during error messages, VLIST etc. So that graphics can be printed, if desired, I have defined GMIT which is identical to the old EMIT. Secondly, I redefined CR so that it zeros OUT each time it is executed. This makes text formatting much easier. For example a tab function can be provided by:

```
: TAB (  n ---  ) OUT @ - SPACES ;
```

I hope to have an FDOS version of FORTH working fairly soon and also have plans for a version using native disk mode in SP 6015 BASIC format.

Here are two FORTH utilities you may like to try: The DUMP routine expects an address on the stack and provides a hex and ASCII dump until SHIFT-BREAK is pressed. It leaves the current address on the stack so that a dump can be continued easily. The commands WIDE and NARROW format the output for screen or printer.

The SET and RESET commands work as for BASIC except for the reverse polish notation.

The format is: x y SET or x y RESET

Trying to Print off the screen is trapped.

```
(HEX DUMP WITH ASCII addr ---)
HEX
S VARIABLE DWIDTH
: 2H. S->D <# # # #> TYPE SPACE ;
: ASC. DUP 1F > IF EMIT ELSE DROP 2E EMIT THEM :
: HEXLINE DWIDTH @   0 DO DUP I + C@ 2H. LOOP ;
: ASCLINE DWIDTH @   0 DO DUP I + C@ ASC. LOOP;
: ADDR.  DIP U. SPACE ;
: 1LINE ADDR. HEXLINE  SPACE ASCLINE CR ;
: DUMP HEX CR BEGIN 1LINE DWIDTH @ + ?TERMINAL UNTIL ;
: WIDE HEX 10 DWIDTH ! ;
: NARROW HEX 8 DWIDTH ! ;
;S
```

```
(SET-RESET GRAPHICS)
HEX D000 CONSTANT VIDRAM 0 VARIABLE ADDR
: AD@R? ( x y --) ) 2 / 28  * SWAP 2 / + VIDRAM + ADDR ! ;
: CHAR? ( xrem yrem - c ) 3 * 1 + SWAP IF 2 * THEN F0 OR ;
: SETIT ( c --- ) ADDR @ C@ DUP F0 < IF DROP ELSE OR THEN
       ADDR @ C! ;
: SETUP 2DUP ADDR? 2 MOD SWAP 2 MOD SWAP CHAR? ;
: RANGE 31 MIN SWAP 4F MIN SWAP ;
: SET (x y --- )
     RANGE     SETUP  SETIT ;
: RESETIT ( c --- ) ADDR @ C@ DUP F0 > IF FF ROT - F0 OR AND
 ADDR @ C!  ELSE 2DROP THEN ;
: RESET ( x y --- )
    RANGE      SETUP RESETIT    ;
DECIMAL
;S
```

****

FORTH PUBLIC DOMAIN SOFTWARE GROWS

thanks to

EDMUND RAMM

My latest version of Z80 fig-FORTH for the MZ-80K contains all the fig-Model definitions plus 1-, 2-, 2* and 2/. Disk I/O does access all 140 screens on an MZ-80K format disc, ID. suppresses bit 7 of the last characters in a name field, affording a more readable VLIST, and the DEL key now generates a destructive backspace.

You will find execution speed much improved compared to the original 8080 version. The BYTE prime numbers test runs in just 7.7 seconds and the PCW Benchmarks are as follows (4MHz clock and including magnifier):

magnifier	1.0
do-loop	7.8
literal	10.7
literal-store	15.6
variable	10.0
variable-fetch	12.7
constant	10.6
dup	13.3
increment	13.2
test>	17.8
test<	17.8
while-loop	18.9
until-loop	17.1
dictionary-search	7.8
arithmetic	5.5

In true FORTH spirit I have put this work in the public domain and will give it to everyone who submits media and return postage. Sharpsoft are free to do the same, of course.

Sharp MZ-80A and MZ-80B versions are in progress and will be available soon.

SAVING GRAPHIC DISPLAYS TO DISK ON THE MZ-80B

by

E.H. Niblett

Before attempting to discuss some simple methods of saving to disc it is necessary to stress the desirability, for any serious graphic work, of setting aside an 8k Back-up storage area to which the picture can be progressively posted as satisfactory stages are reached. This means that in the event of an error it is not necessary tediously to erase,piece by piece,the offending area. The whole graphic display can be cleared and then replaced from store with the last acceptable stage. If two of the "F" keys are assigned to these posting and recall operations transfer is simple and instantaneous.

A further virtue of the back up store is that it permits numerous inverting,inserting,shifting and similar routines.

The schemes described are accordingly based on the existence of such a storage area and it is from and to this that the saving and reloading is carried out. We use the area from $A000 to $BF3F and the subroutines listed are tailored to these addresses.

(1) Having made the transfer to the back up area,the simplest method of saving is to go first to tape through the MON "S" routine specifying the addresses shown (with a jump address of $1280 if part of a Basic program). Once on tape it can be transferred to disc through the normal "Filing CMT" program of the master disc as an Object File. Although this method is time consuming,it does have the advantage that no special programs are necessary and that several pictures can be recorded sequentially and subsequently transferred to disc at a single session.

Recall from disc takes only three seconds and the routines are shown below. The Basic section switches the graphic RAM for access and the machine code part is a simple block transfer instruction. The basic line numbers of course,have no particular significance- this is just how they occur in a working graphic program. Similarly the location of the machine code is optional providing the Basic USR address is altered to suit.

(2) Since the picture ia already in memory it does perhaps seem a bit pointless to take it off on to tape,only to put it back again for the "Filing CMT" routine. In practice this operation can be circumvented by another short program to shift the data to the required operating area and to provide the necessary header buffer information and the title,in lieu of the tape. This program includes a 3 byte modification to the monitor at the CMT initiation address. Resumption of the "Filing" program occurs at the point to

which the tape would have returned control.
The sequence of operations then becomes - Insert title as A$ - WOPEN USR routine to transfer title to machine code segment - RUN "Filing CMT" from master disc - Enter Drive No. Machine code routine then provides all necessary information and contents of $A000 to BF3F are saved to disc as an Object File.

(3) The third method has certain refinements to improve handling. It stores a modification of the "Filing" routine on the graphic program disc where it can be renamed "GRAPHSAVE", for instance and chains it to the main program for automatic loading at the appropriate point. No modification is required to the Monitor sequence - the jump can be catered for on the Filing section .At the same time the opportunity can be taken,for the sake of completeness,to alter the display headings to reflect the fact that transfer is taking place from Back up Store to disc rather than from cassette.

To get at the "Filing" program for modification,first RUN it from the master disc in the usual way. Do not enter any drive number when invited but push the Monitor reset button at the rear of the MZ80B to return control to the monitor. Then it is possible to display the program and to modify it with the "M" command.

The program loads just after the interpreter at address $675C. It is then transferred to its executing address of $1220. Modifications must be carried out on the higher address block as indicated. The amended program can then be saved to tape by the "S" command,not forgetting the jump address to the new short machine code routine which will relocate it at its proper executing address. The program can then be transferred to disc under its new title by the "Filing CMT" program operating in its normal mode.
The address block to be saved is $675C to $7DB6 with Jump address $CDA8 (if the locations of the example routines are adopted).

The vital amendments to the "Filing" program before renaming are:-
$67CA C3 $67CB 40 67CC [illegible]
The screen message display addresses which can be altered are from $6877 to $68A5 . The actual replacement wording is a matter of personal choice.

To complete a working program from the above routines will require a few housekeeping tasks. The machine code sequences must be given a file name and saved to disc (unless they are going to form part of an existing routine whose loading will already be catered for). A limit statement will be required and the Def Keys defined,as shown in the typical opening line numbers. The Basic sequences can be inserted in other programs as required.

When downloading previously recorded displays ,the data will be returned to the Back up store. A LIMIT $9FFF statement may be required before loading and it is good practice to keep this and the recall sequence as an opening program on any disc carrying a number of graphic displays.

One disc will store many pictures and any one can be called down as required,modified and resaved. Moreover it is not necessary to transfer the whole of a stored picture from back-up to display - sections may be picked off and set in any part of the image being constructed ,but the routines for this are beyond the scope of this particular article.

A few final words of warning :- POSTING A BLANK SCREEN TO THE BACK-UP AREA CLEARS IT ENTIRELY.

Also remember that the "Filing CMT" or substitute program clears part of the interpreter so that rebooting is necessary after saving to disc. We have never found this to be such a disadvantage as to warrant relocating the whole program. However if drawings are being made with the help of a newly constructed program,ensure it is separately saved before starting the graphic saving routine.

E.H.N.
Feb,83

```
1 REM              PREPARATION SEQUENCE
10 DEF KEY(5)=GOTO 4000↵
20 DEF KEY(6)=GOTO 5000↵
30 DEF KEY(9)=GOTO 12000↵
40 CONSOLE C80
50 LIMIT $9FFF
60 LOAD "MC/1"



4000 REM                  COPY TO BACK UP STORE
4005 INP@232,X
4010 X1=X+128
4015 OUT@232,X1
4020 USR($CB20)
4025 OUT@232,X
4042 GRAPH 00
4045 PRINT "[illegible]"
4050 CURSOR16,4:PRINT" SAVE ON TAPE THROUGH MON S $A000 TO $BF3F"
4065 CURSOR 28,6 :PRINT "J ADR $1280"



5000 REM                  RECALL FROM STORE
5010 INP@232,X
5020 X1=X+128
5030 OUT@232,X1
5040 USR($CB30)
5080 OUT@232,X
5090 GRAPH I1,01



11990 REM                  SAVE TO DISC
12000 USR($CD32) :PRINT CHR$(06)
12008 CURSOR 27,4:PRINT"TO SAVE DIRECT TO DISC"
12010 CURSOR 27,6:INPUT"ENTER TITLE ?  "  ;A$
12012 WOPEN#1,USR($CD00)
12014 PRINT#1,A$
12016 CLOSE#1
12040 CURSOR16,14:PRINT" REBOOT AND RELOAD TO RETURN TO PROGRAM"



12042 REM            THIS SECTION FOR METHOD (2)
12045 USR($CD90)
12048 CHAIN FD1,"Filing CMT"



12042 REM              OR THIS LINE FOR METHOD (3)
12045 CHAIN "GRAPHSAVE"
```

SUB ROUTINE MIC/1

```
CB20  21 00 E0 11 00 A0 01 3F 1F ED B0 C9 00 00 00 00 Transfers to Back-up
CB30  21 00 A0 11 00 E0 01 3F 1F ED B0 C9 00 00 00 00 Transfers from Back-up


CD00  EB 11 21 CD ED B0 C9 00 00 00 00 00 00 00 00 00 Puts title at CD21
CD10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
CD20  01 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 0D 01=Object File code
CD30  0D 0D 21 21 CD 06 10 3E 0D 77 23 10 FC C9 00 00 Fill title with 0D's
CD40  21 20 CD 11 C0 10 01 10 00 ED B0 21 40 1F ED 63 Transfer title,size
CD50  D2 10 21 00 A0 ED 63 D4 10 21 B1 00 ED 63 D6 10 and addresses to buffer
CD60  21 B9 47 06 C0 3E 00 77 23 10 FC 21 00 A0 11 79 Clear beyond picture &
CD70  28 01 40 1F ED B0 2A D4 10 22 B6 13 33 33 C3 B2 shift to working area &
CD80  12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 return to program
CD90  21 8E 02 36 C3 21 8F 02 36 40 21 90 02 36 CD C9 Alter Mon tape sequence
CDA0  00 00 00 00 00 00 00 00 01 5A 16 21 5C 67 11 20 Relocate Graphsave
CDB0  12 ED B0 C3 20 12 00 00 00 00 00 00 00 00 00 00 after loading
```

SAVE $CB20 TO $CDBF JUMP ADDR. $1280
COPY/P1

STRING handling in PASCAL

by

Mike Brinson

One of the features which makes BASIC an attractive and useful programming language is its good string handling facilities. In standard PASCAL text is represented by the data type CHAR. Additionally the language does not include any string handling functions or procedures. New PASCAL programmers often consider the omission of RIGHT$, LEFT$ etc. a fundamental mistake in the design of the PASCAL language. PASCAL is howerer, like FORTH, an extensible language allowing programmers to write their own string routines. The following notes should help you develop your own routines for manipulating PASCAL strings.

The most important fact which must be understood by the PASCAL programmer when writing string routines is the structure which describes a string. The basic string structure is formed from two parts - firstly the string's LENGTH and secondly the string of characters which compose the string text. These components each have different DATA TYPES which gives us a clue as to how we may represent a string in PASCAL. PASCAL is a strongly TYPED language which encourages the programmer to define a new data type to represent the string. This new data type we will call STRING and define it to be a RECORD with components LENGTH and an ARRAY of CHARS, for example

```
CONST   SLENGTH = 80; (* Maximum string length *)

TYPE    TXT = ARRAY [1..SLENGTH] OF CHAR;

        STRING = RECORD

                   LENGTH : INTEGER
                   T      : TXT
                   END;

VAR     TSTRING,TEMPSTRING : STRING;
```

Unfortunately in standard PASCAL functions are not allowed to return structured data types. Our new data type STRING is an example of a structured data type. Hence we are forced to develop string handling procedures rather than the more familiar functions which are used in BASIC.

The following demonstration program includes most of the important string handling features found in BASIC. This program is written in HISOFT PASCAL and will run on the MZ-80K, A and B computers. The program includes:

1. PROCEDURE SREAD (VAR X : STRING);

   Reads a string from the terminal. The string is returned as string X. Entry is terminated when the CR key is pressed. The procedure echos each character to the VDU as it is typed. It also recognises backspace so that you can correct errors at entry.

2. PROCEDURE SWRITELN (X : STRING);

Outputs string X to the VDU then moves the cursor to the start of the next line.

3. FUNCTION LEN (X : STRING) :INTEGER;

Returns the length of the string.

4. PROCEDURE CAT (VAR Y : STRING;
X : STRING);

Returns as string Y the concatonation of strings Y and X. Note the final length of string Y must not be greater than SLENGTH.

6. PROCEDURE SMID (VAR Y : STRING;
X : STRING;
FIRST,NUMBER : INTEGER);

Equivalent to the BASIC MID$ function. Returns as string Y the string of characters found in string X beginning with the FIRST characters from the left and continuing for NUMBER characters from that point.

7. PROCEDURE SLEFT (VAR Y : STRING;
X : STRING;
NUMBER : INTEGER);

Equivalent to the BASIC LEFT$ function. Returns as string Y the string consisting of the characters in string X, starting with the leftmost characters up-to and including character NUMBER.

8. PROCEDURE SRIGHT (VAR Y : STRING;
X : STRING;
NUMBER : INTEGER);

Equivalent to the BASIC RIGHT$ function. Returns as string Y the string consisting of the characters in string X, starting with the characters in position NUMBER and ending with the rightmost character.

```
PROGRAM STRINGS;
CONST
        SLENGTH = 80;  (* LENGTH OF STRING *)

TYPE
        TXT    = ARRAY[1..SLENGTH] OF CHAR;
        STRING = RECORD
                        LENGTH : INTEGER; (* LENGTH OF STRING *)
                        T      : TXT      (* TEXT   OF STRING *)
                 END;
VAR
        TSTRING,TEMPSTRING   :  STRING;
```

```
PROCEDURE SREAD( VAR X : STRING);

(* READ A STRING FROM THE TERMINAL *)

VAR
        N  : INTEGER;
        CH,NULL,BACKSP : CHAR;
BEGIN
 NULL   := CHR(0);
 BACKSP := CHR(8);
 WITH X DO
 BEGIN
  FOR N := 1 TO SLENGTH DO T[N] := NULL;
   N := 1;
   IF EOLN THEN READLN;
   REPEAT
    READ(CH);
    T[N] := CH;
    IF CH = BACKSP THEN
     BEGIN
      T[N] := NULL;
      N := N-1
     END;
    IF NOT EOLN THEN N := N+1;
   UNTIL EOLN OR ( N > SLENGTH );
   LENGTH := N
  END
END; (* OF SREAD *)

PROCEDURE SMID( VAR Y : STRING;
                    X : STRING;
                    FIRST,NQMBER : INTEGER);

(* MID$ ROUTINE *)

VAR
        N,M,TEMP : INTEGER;
        NULL     : CHAR;
BEGIN
 NULL := CHR(0);
 FOR N := 1 TO SLENGTH DO Y.T[N] := NULL;
 FOR N := 1 TO NUMBER DO
  BEGIN
   TEMP := FIRST+N-1;
   IF TEMP <= SLENGTH THEN Y.T[N] := X.T[TEMP]
  END;
 Y.LENGTH := NUMBER
END;  (* OF SMID *)


FUNCTION LEN( X : STRING ) : INTEGER;
(* RETURNS THE LENGTH OF THE STRING *)
BEGIN
 WITH X DO  LEN := LENGTH
END; (* OF LEN*)
```

```
PROCEDURE SWRITELN( X : STRING );
(* WRITELN PROCEDURE FOR STRINGS *)
VAR
        N : INTEGER;
BEGIN
 WITH X DO
  BEGIN
    FOR N := 1 TO LEN(X) DO WRITE(T[N]);
    WRITELN
  END
END; (* OF SWRITELN *)


PROCEDURE SLEFT( VAR Y : STRING;
                     X : STRING;
                     NUMBER : INTEGER);

(* LEFT$ ROUTINE *)

BEGIN
 IF NUMBER > LEN(X) THEN NUMBER := LEN(X);
 SMID(Y,X,1,NUMBER)
END;   (* OF SLEFT *)


PROCEDURE SRIGHT ( VAR Y : STRING;
                       X : STRING;
                       NUMBER : INTEGER);

(* RIGHT$ ROUTINE *)

BEGIN
 IF LEN(X) > NUMBER
  THEN
       SMID(Y,X,LEN(X)-NUMBER+1,NUMBER)
  ELSE
       SMID(Y,X,1,LEN(X))
END;    (* OF SRIGHT *)


P OCEDURE CAT( VAR Y : STRING;
                   X : STRING);

(* CAT OF STRING Y AND STRING X
         RESULT TO Y,
         FINAL LENGTH OF Y MUST NOT BE
         GREATER THAN STRLENGTH *)

VAR
        N,K : INTEGER;

BEGIN
 K := Y.LENGTH;
 FOR N := 1 TO X.LENGTH DO
     Y.T[N+K] := Y.T[N];
 Y.LENGTH := N+K
END;  (* OF CAT *)
```

```
(* START OF MAIN DEMONSTRATION PROGRAM *)

BEGIN
WRITELN('START OF STRING TEST');
WRITELN('INPUT STRING PLEASE ');
SREAD(TSTRING);
SWRITELN(TSTRING);
WRITELN('THE STRIJG LENGTH IS ',LEN(TSTRING));
SMID(TEMPSTRING,TSTRING,4,3);
SWRITELN(TEMPSTRING);
WRITELN('THE MID$ STRING LENGTH IS ',LEN(TEMPSTRING));
WRITELN('INPUT ANOTHER STRING PLEASE');
SREAD(TSTRING);
SWRITELN(TSTRING);
WRITELN('THE SECOND STRING LENGTH IS ',LEN(TSTRING));
WRITELN('INPUT ANOTHER STRING PLEASE ');
SREAD(TSTRING);
SLEFT(TEMPSTRING,TSTRING,5);
SWRITELN(TEMPSTRING);
WRITELN('THE STRING LENGTH IS ',LEN(TEMPSTRING));
WRITELN('INPUT ANOTHER STRING PLEASE');
SREAD(TSTRING);
SRIGHT(TEMPSTRING,TSTRING,5);
SWRITELN(TEMPSTRING);
WRITELN('THE STRING LENGTH IS ',LEN(TEMPSTRING))
END.
```

****

BASIC Programmer - Alan Stevens
Gets Converted to PASCAL!

I was pleased to see the PASCAL section in Issue 8 of the S.U.N. as I have recently bought Hisoft Pascal for the MZ-80K. A 'died-in-the-wool' BASIC programmer who scoffs at the 'in' crowd's structured programming, I nevertheless find myself increasingly attracted by the structural cleanliness and efficiency of PASCAL! (I cannot say the same of FORTH I'm afraid - a horrible language!!) I am very impressed with Hisoft's implementation of the language itself, my only dislike being the line editor which is a real 'pain-in-the-neck' after BASIC's full screen editor.

There is one deficiency in Hisoft Pascal for those of us who have non-Sharp printers. Hisoft have built-in a printer driver routine which assumes that a Sharp printer is attached. They do realise that other printers might be attached and so provide a Pascal program in the accompanying documentation which purports to modify the driver to be compatible with the Epson MX80 series of printers. Since I have an Epson MX80FT3 I eagerly tried this modification. Unfortunately, I discovered that the modification was such that the computer refused to communicate with the printer whatsoever!! This was rather annoying, the more so because the UNmodified driver did at least let the computer speak to the printer in a limited way (no lower case letters, and the bell rang at the end of each line), Luckily, I was able to overcome the problem and for others who may be suffering the same way I attach the program 'Epsonprint' which, run with the original, unmodified version enables Pascal (HP4T) to talk properly to an Epson MX80 type 3.

I was interested in the BYTE magazine prime number program reprinted in Issue 8 of the S.U.N. While this program may provide a superb benchtest of Pascal facilities, it doesn't do a thing for prime numbers I'm afraid. It gets the wrong answer (try it with 'Size' replaced by a small number)! For those who want to generate prime numbers (as well as count them) I attach the program 'Eratosthenes'.

Hisoft imply that the inbuilt RANDOM function, though fast, is not very random! The attached function 'RND' should produce better 'random' numbers (more slowly!). Note that SEED1, SEED2 and SEED3 must be global not local variables. A negative argument will re-initialise the random series.

While on the subject of random numbers may I present my understanding of how Sharp Basic SP-5025 generates its random numbers. (I delved into Avalon's commented Basic listing after reading John Simpson's article in S.U.N. 8).

Each time the RND function is called a 'seed' number if first identified. If the function is called for the first time, or if it is called with a negative argument, the seed used is an inbuild constant (= 0.20298827). Subsequent calls use the previously generated 'random' number as the seed. This seed is then multiplied by 23. (Why 23? I don't know. Why not? What number would you choose?!) If the resulting number is less than 1 it becomes the random number and the next seed. If it is greater than 1 it undergoes the following process. First, its (binary) exponent is decreased by one. Because it is a binary exponent this is the same as dividing the number by 2. Secondly, the (binary) mantissa is 'multiplied' by 2 by shifting all its bits one place to the left and 'dropping' any non-zero bit that moves to the left of the (imagined) binary point. For example, if the original mantissa were .11001010, a shift to the left gives 1.10010100, which becomes .10010100 on dropping the '1' to the left of the binary point. (SP-5025 uses 4 bytes to represent the mantissa rather than the 1 byte used in my example). This 'division' and 'multiplication' by 2 process is repeated until the resulting number (i.e. combined mantissa and exponent) is less than 1. This is then the random number and subsequent seed.

I do not guarantee that the above explanation is correct, but is the best interpretation I have been able to make of the listed routine. Why this process should provide a good set of random numbers, I've no idea!

Incidentally, one way of avoiding a repetition of the SAME random numbers on switch-on is to include the following line:

```
100 GET A$: R=RND(1): IF A$="" THEN 100
```

This approach can only be adopted of course if a page of instructions have first to be read, so that the interpreter is buzzing round and round line 100 while it waits for a 'continue' key to be pressed.

(Listings continued overleaf)

```
6041    10 PROGRAM Eratosthenes;
6041    20
6041    30 (* Calculates prime numbers using Eratosthenes sieve *)
6041    40
6041    50 CONST  size = 8190; (* or what you like *)
6041    60
6041    70 VAR    i,prime,crossout,rootsize,count : INTEGER;
604A    80        flag : ARRAY[2..size] OF BOOLEAN;
604A    90
604A   100 BEGIN
6053   110   FOR i:=2 TO size DO flag[i]:=TRUE;
6092   120   rootsize:= TRUNC(SQRT(size));
60A1   130   IF rootsize=1 THEN rootsize:=2;
60B9   140   FOR prime:=2 TO rootsize DO
60D7   150   BEGIN
60DA   160     IF flag[prime] THEN
60F9   170     BEGIN
60F9   180       crossout:=prime + prime;
610A   190       WHILE crossout<=size DO
6122   200       BEGIN
6122   210         flag[crossout]:=FALSE;
6140   220         crossout:=crossout + prime
6144   230       END
6151   240     END
6154   250   END;
6158   260    count:=0;
615E   270    FOR i:=2 TO size DO
617B   280    IF flag[i] THEN
619A   290    BEGIN
619A   300      count:=count + 1;
61A1   310      WRITE(count,i);
61B3   320      WRITELN
61B3   330    END
61B6   340 END.
End Address: 61BB



5F2C    10 PROGRAM Epsonprint;
5F2C    20
5F2C    30 (* Makes Pascal HP4T talk to Epson MX80FT3 *)
5F2C    40
5F2C    50 CONST  ret = CHR(201);
5F2C    60        nop = CHR(0);
5F2C    70        call= CHR(205);
5F2C    80        low = CHR(167);
5F2C    90        addr= #129E;
5F2C   100
5F2C   110 VAR    I : INTEGER;
5F35   120
5F35   130 BEGIN
5F3E   140        POKE(#1299,low);
5F46   150        POKE(#12E9,low);
5F4E   160        POKE(addr,ret);
5F56   170        FOR I:=1 TO 8 DO POKE(addr+I,nop);
5F88   180        POKE(#12A7,call);
5F90   190        POKE(#12A8,#1445)
5F98   200 END.
End Address: 5F9E
```

```
6128   10 PROGRAM RND;
6128   20
6128   30 (* based on 'An efficient and portable pseudo-random number
6128   40     by Wickmann and Hill                                    generator' *)
6128   50    Applied Statistics Vol 31 No 2 (1982)
6128   60
6128   70 VAR SEED1,SEED2,SEED3, I : INTEGER;
6131   80
6131   90 FUNCTION RND(N:INTEGER):REAL;
6134  100 VAR  R : REAL;
6134  110 BEGIN
614C  120   IF N<0 THEN BEGIN
6163  130                SEED1:=5271; (* any integers between *)
6169  140                SEED2:=934;  (* 1 and 30000 will do  *)
616F  150                SEED3:=25859;(* for these seeds      *)
6175  160                END;
6175  170   SEED1:=171*(SEED1 MOD 177)-2*(SEED1 DIV 177);
61A6  180   SEED2:=172*(SEED2 MOD 176)-35*(SEED2 DIV 176);
61D7  190   SEED3:=170*(SEED3 MOD 178)-63*(SEED3 DIV 178);
6208  200
6208  210   IF SEED1<0 THEN SEED1:=SEED1+30269;
622B  220   IF SEED2<0 THEN SEED2:=SEED2+30307;
624E  230   IF SEED3<0 THEN SEED3:=SEED3+30323;
6271  240
6271  250   R:=SEED1/30269+SEED2/30307+SEED3/30323;
62BA  260
62BA  270   RND:=R-TRUNC(R)
62D6  280 END;
62F6  290
62F6  300 BEGIN
62FF  310   SEED1:=5271;SEED2:=934;SEED3:=25859;
6311  320   FOR I:=1 TO 8 DO WRITELN(RND(1));
6344  330   WRITELN(RND(-1))
6357  340 END.
End Address: 635C
```

A SIMPLE INTRODUCTION TO GRAPHICS ON THE MZ-80K BY JOHN SIMPSON

PART 2

In the last issue we looked at simple programs to demonstrate the use of the POKE facility and saw how this could be used to make an image move around the screen. However, in that article we only looked at objects following a laid down path or moving at random. It is clear that for games we will need some control over the movement of our objects - in other words we want them to respond to instructions from the keyboard. The MZ-80K has a command "GET" which serves this purpose. In this article we shall initially consider the use of the GET command and then go on to see how the PEEK function can make use of graphics more sophisticated.

The GET command in a program scans the keyboard to see which key is being pressed. However, it does this very quickly - much faster than you could press a key! - so it is necessary to make it scan the keyboard repeatedly until it receives an input. The following short routine does this:

```
100 GETKB$:IFKB$=""THEN100
```

It simply puts the computer into a loop, constantly repeating line 100 until an input from the keyboard is received. This little routine has many uses. It can keep the computer waiting, for example while you read instructions at the beginning of a game. If you end the instructions with "Press any key to start" this routine will not allow the computer to proceed with the rest of the program until you are ready. However, the use in which we are interested now is that of getting instructions from the player on which way he wants an object to move.

In the last article we saw how a flying saucer could be made to move in any direction at random. We can now rewrite that program, using the GET routine, to make the saucer obey our commands, but first we have to decide which commands we will use. The keys most commonly used for directions are those set out below:

```
Q  W  E
 \ ↑ /
A←   →D
 / ↓ \
Z  X  C
```

But it doesn't really matter which keys you choose as long as they are logically related to the direction of movement and are convenient for the user. Some people prefer to use the blue graphics keys for this purpose, particularly if the game is likely to result in rather violent use of the keyboard!

What is most important is getting the right relationship between the old and new positions of the object, and this is expressed as the plus or minus value required to be added to the current screen location as follows:

```
-41  -40  -39
  Q    W    E
   \   ↑   /
-1  A←      →D  +1
   /   ↓   \
  Z    X    C
+39  +40  +41
```

An alternative version is also given for the blue graphics keys at the lower right hand corner of the keyboard. You may like to experiment until you establish the group of keys which you find easiest to use.

```
-41  -40  -39
  ▀    ▌    ▛
   \   ↑   /
-1  ▄←      →▙  +1
   /   ↓   \
  ▄    ▌    ▞
+39  +40  +41
```

Now we're ready to start controlling our flying saucer. Study the following listing and compare it with the flying saucer program in the last article. The only changes we have made are in line 100, where we have made the program react to the instructions from the keyboard. There are, however, some oddities about this program which would make it unsuitable for use in a proper game. RUN the program and then press the direction keys in turn to satisfy yourself that the routine works. Now press any direction key and then press any key on the keyboard which is <u>not</u> a direction key. The saucer keeps moving in the same direction! Can you see why this happens? Study the listing and try to work it out before you go on.

```
10 PRINT"■"
20 SC=53248
30 PS=SC+500
40 POKE PS,199
50 FORJ=1TO75:NEXTJ
60 OP=PS
70 GOSUB100
80 POKE OP,0
90 GOTO40
100 GETKB$:IFKB$=""THEN100
110 IFKB$="Q"THEN X=-41
120 IFKB$="W"THEN X=-40
130 IFKB$="E"THEN X=-39
140 IFKB$="A"THEN X=-1
150 IFKB$="D"THEN X=1
160 IFKB$="Z"THEN X=39
170 IFKB$="X"THEN X=40
180 IFKB$="C"THEN X=41
190 PS=PS+X
200 IFPS<SCTHENPS=PS+40
210 IFPS>SC+999THENPS=PS-40
220 RETURN
```

Line 100 accepts an input from the keyboard. If that input is one of the direction keys lines 110-180 assign a value to X which is then added to the screen position in line 190. However, if the input received is not one of the direction keys the previous value of X will be added to the screen position when the program reaches line 190, and so on.

In order to overcome this it is necessary to reset X to zero each time the subroutine is called, so change line 70 as follows:

```
70 X=0:GOSUB100
```

Now the program will react only to the direction keys you have determined.

Try making the saucer move continually to the right. Eventually it goes off the edge of the screen and reappears (one line lower) on the left. Similarly, making it move to the left will send it off the edge of the screen and make it reappear one line higher on the right. Can you see why this happens? (Look at the screen chart - diagram 1 - in the last issue and work out the position of the saucer when you continually add or subtract 1.)

You can produce a similar effect if you move the saucer diagonally off the screen.

The effect is odd but how much does it matter to your game? You will see some of the older arcade games where this happens, and it can add an extra dimension to play, allowing you to develop unorthodox tactics. You may find this acceptable - and it certainly keeps the programming easy! - but if you don't, how can we overcome this effect?

One solution would be to incorporate a whole load of IF ... THEN ... statements to check for all the positions down the edges, but that seems like too much hard work, and is not very efficient. Another way is to use the PEEK command and we shall be looking at that later in the article.

But for the moment we'll look at another solution which uses Cartesian co-ordinates. (Don't worry, it's not as complicated as it sounds!) Remember when you were at school and they made you draw graphs, plotted along X:Y axes? This technique can be adapted for POKEing to the screen, although if you remember your graphs, it will

seem topsy-turvey. The X:Y axes on your graphs normally originated from the bottom left corner of your paper, but on the computer, because of the way the screen addresses have been numbered, we have to start from the top left corner of the screen.

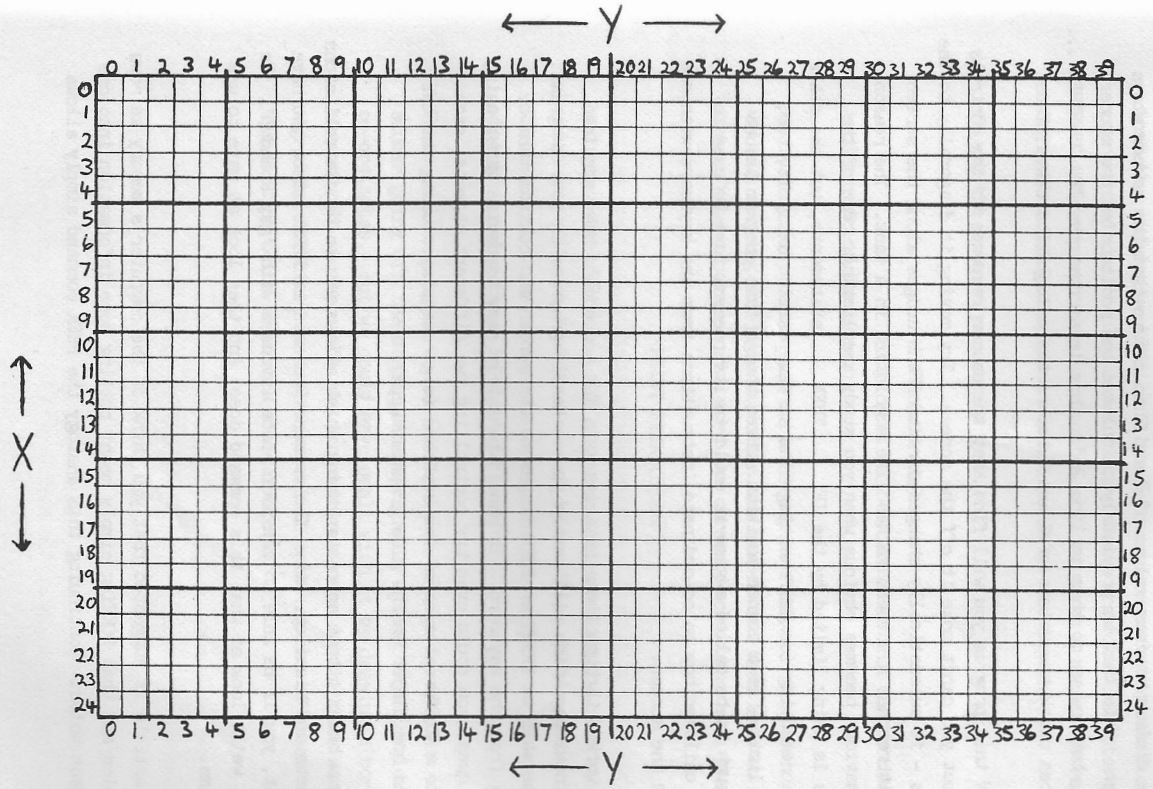
Look at Diagram 3. This is an amended version of the screen address chart which appeared in the previous article. Count down the lines from 0 to 12. Now count along the next line from 0 to 20. You will find yourself roughly at the centre of the screen diagram. We shall refer to the number of complete lines we have moved down as X, and the number of spaces along the next line as Y. Now we have to translate these co-ordinates into a form which the computer can understand as a POKE instruction.

We still have to think in terms of adding the appropriate value to the screen corner address (SC=53248) and this is how we calculate the value. We want to move down 12 complete lines and then along 20 spaces. Since X represents complete lines of 40 spaces, we have to multiply X by 40 before adding it to the screen corner address and then we simply add Y to the total.

You will see how a position near the centre of the screen is worked out in line 30 of the following program. (We don't need to put X*40 in brackets, because the computer always works out multiplications before additions - see page 33 of your Sharp manual.)

```
10 PRINT"■"
20 SC=53248
30 X=12:Y=20:PS=SC+X*40+Y
40 POKE PS,199
50 FORJ=1TO75:NEXTJ
60 OP=PS
70 GOSUB100
80 POKE OP,0
90 GOTO40
100 GETKB$:IFKB$=""THEN100
110 IFKB$="Q"THEN X=X-1:Y=Y-1
120 IFKB$="W"THEN X=X-1
130 IFKB$="E"THEN X=X-1:Y=Y+1
140 IFKB$="A"THEN Y=Y-1
150 IFKB$="D"THEN Y=Y+1
160 IFKB$="Z"THEN X=X+1:Y=Y-1
170 IFKB$="X"THEN X=X+1
180 IFKB$="C"THEN X=X+1:Y=Y+1
200 IFX<0THENX=0
205 IFY<0THENY=0
210 IFX>24THENX=24
215 IFY>39THENY=39
218 PS=SC+X*40+Y
220 RETURN
```

DIAGRAM 3



SCREEN POSITIONS ARE HELD IN RAM FROM 53248 ONWARDS.

THUS TO PLACE CHARACTER C AT POSITION P — POKE(53248+X*40+Y),C

(X = VERTICAL CO-ORDINATE WHERE $0 \leq X \leq 24$)

(Y = HORIZONTAL CO-ORDINATE WHERE $0 \leq Y \leq 39$)

(C = CHARACTER CODE IN MZ-80K DISPLAY CODE WHERE $0 \leq C \leq 255$)

Study this program and compare it with the previous version to see how the commands from the keyboard are related to changes in the X and Y values which will bring about movement in the desired direction. (Lines 110-180). Then see how we can simply use the IF ... THEN ... statements in lines 205-215 to limit the minimum/maximum values of X and Y to ensure that the saucer stays on the screen. Note that we previously calculated the position of the saucer in line 190, but now we do this in line 218. This is so that the IF ... THEN ... statements can check the values of X and Y before the calculation takes place.

When you RUN this program you will find that the saucer responds to the controls as before, but you can't move it off the screen. Try moving it diagonally to the screen edges - it automatically changes direction and nudges along the screen edge until it reaches a corner, rather like a goldfish in a tank. The reason for this behaviour becomes obvious when you study the listing. One of the co-ordinates is being limited by the IF ... THEN ... statements, but the other can go on incrementing (or decrementing) the screen position until it, too, reaches its limit - ie a corner position. Even though this program is not exciting enough to be called a game it could in its present form be used to teach young children how to co-ordinate instructions from the keyboard with the movements on the screen.

If you have worked through these articles this far you should have acquired some basic knowledge which will enable you to handle graphics in your programs. You should be able to PRINT or POKE images to the screen and control them by instructions from the keyboard. It should be fairly easy to devise games using these techniques; you could even try controlling two different objects, each responding to a different area of the keyboard (e.g. one player using the QWE etc. keys and one player on the blue graphics keys).

But so far you haven't come across anything which allows you to handle collisions between objects and missiles, etc. The command you need is PEEK. Once you have mastered that, you'll be able to zap your space invaders with 'gay abandon'. So first of all we'll look at what this command does, and then look at ways to use it in programs.

In the last article it suggested that you think of the computer's memory as rows of pigeon holes and that POKEing simply meant putting some information into one of these pigeon holes. Continuing this analogy the PEEK command simply allows

you to look (or PEEK) at the information held in any particular one of the pigeon holes which make up the computer's memory. Once again we're only looking at those pigeon holes or memory locations numbered from 53248 to 54247 which hold the information displayed on the screen. You will remember that the numbers we POKEd into these locations were between 0 and 255, and represented the display codes for the Sharp graphics characters. So if we PEEK into a particular location we will be given a number corresponding to the symbol being displayed at that point on the screen.

To use the command you simply type:

PEEK (address)

e.g. PEEK (53248) looks at what is in the top right hand corner of the screen.

We can try out the command quite simply as follows. In the direct mode (ie without running any program) clear the screen and then type an asterisk in the top left hand corner. Now move the cursor down to the next line and type:

PRINT PEEK (53248)

and press the 'CR' key. The computer will immediately come back with '107' which you will see from your manual is the Sharp display code for an asterisk. In practical terms this means we can use the command to look ahead, and see what our moving image is about to run into.

In an earlier program we had difficulties because we couldn't stop our flying saucer going off the screen, and we finally solved that problem by use of XY co-ordinates. But we could have dealt with it by drawing a border round the screen. If we then made the program look ahead to the next square the saucer was about to enter, we could prevent it entering one of these border squares. The listing below shows how this is done.

```
1 PRINT"E"
5 SC=53248
12 FORI=0TO39:POKE SC+I,208:NEXTI
13 FORI=40TO920STEP40:POKESC+I,208:NEXTI
14 FORI=79TO959STEP40:POKESC+I,208:NEXTI
15 FORI=960TO999:POKESC+I,208:NEXTI
30 PS=SC+500
40 POKE PS,199
50 FORJ=1TO75:NEXTJ
60 OP=PS
70 X=0:GOSUB100
80 POKE OP,0
90 GOTO40
100 GETKB$:IFKB$=""THEN100
110 IFKB$="Q"THENX=-41
120 IFKB$="W"THENX=-40
130 IFKB$="E"THENX=-39
140 IFKB$="A"THENX=-1
150 IFKB$="D"THENX=1
160 IFKB$="Z"THENX=39
170 IFKB$="X"THENX=40
180 IFKB$="C"THENX=41
190 Q=PS+X
200 IFPEEK(Q)<>0THENRETURN
210 PS=Q
220 RETURN
```

You will see that this is our first flying saucer program but with some important modifications. We have inserted lines 12 to 15 which draw the border round the screen using character code 208 (the chequered square) although any other suitable character could have been used. Note the use of the FOR ...NEXT loops (STEPped where necessary) in these lines. The border could have been simply PRINTed into place - but we are trying to practise the POKE command! Because this procedure needs the screen corner (variable SC) as a starting point we have changed the position of the first few lines. This allows us to define SC in line 5, instead of line 20 where it was previously.

The most interesting changes are found in lines 190 to 210. Having established from the keyboard input the desired new direction, and thus the value of X, we calculate the position the saucer is trying to reach in line 190 and hold this in the variable Q. Then in line 200 we use the PEEK command to look into the computer's memory for that screen location. If it is <u>not</u> empty the computer simply retains the saucer's present location as PS and returns to the main program to seek a further instruction, thus leaving the saucer where it was before. If, however, the calculated new location (Q) <u>is</u> empty, then line 200 is ignored, and in line 210 the saucer position (PS) is set to the same value as Q, so that when the computer returns to main program the saucer will be POKEd into the new location required.

You may feel that line 200 is rather awkward in that it checks to see whether the calculated new location is not empty. We could simply have said:

```
IF PEEK (Q) = 208  THEN RETURN
```

ie just check to see whether the calculated location is in fact a border square. There would be nothing wrong with this solution, and it works perfectly well, although it assumes that the saucer is only going to come up against one sort of obstacle. If there are other types of obstacle on the screen these will involve other graphics characters and so the program would have to be able to cope with their various display codes. The advantage of line 200 is that it only allows the saucer to go into an empty square.

With a few further modifications we can now turn this program into a simple game. All we do is add some randomly scattered stars to the screen and make one of them the home planet. The object is then to guide our little extra-terrestial being in his saucer back home as quickly as possible. To add some excitement to the game we can use the Sharp's built-in clock facility to time each player's attempts.

The modified program is listed below:

```
1 PRINT"E"
5 SC=53248
7 FORK=1TO100
8 R=INT(RND(1)*999)
9 POKESC+R,107:NEXTK
10 POKESC+195,71
12 FORI=0TO39:POKE SC+I,208:NEXTI
13 FORI=40TO920STEP40:POKESC+I,208:NEXTI
14 FORI=79TO959STEP40:POKESC+I,208:NEXTI
15 FORI=960TO999:POKESC+I,208:NEXTI
30 PS=SC+842
35 TI$="000000"
40 POKE PS,199
50 FORJ=1TO75:NEXTJ
60 OP=PS
70 X=0:GOSUB100
80 POKE OP,0
90 GOTO40
100 GETKB$:IFKB$=""THEN100
110 IFKB$="Q"THENX=-41
120 IFKB$="W"THENX=-40
130 IFKB$="E"THENX=-39
140 IFKB$="A"THENX=-1
150 IFKB$="D"THENX=1
160 IFKB$="Z"THENX=39
170 IFKB$="X"THENX=40
```

```
180 IFKB$="C"THENX=41
190 Q=PS+X
195 IFPEEK(Q)=71THEN300
200 IFPEEK(Q)<>0THEN USR(62):RETURN
210 PS=Q
220 RETURN
300 PRINT"■■■■■■■■■■■■■■■■■■■■■■CONGRATULATIONS!"
310 PRINT"■■■■■■■■■■■■■■■■YOU GOT E.T. HOME"
320 PRINT"■■■■■■■■■■AND YOUR TIME WAS ";TI$
330 PRINT"■■■■■■■■■■■■■■ANOTHER TRY?  (Y/N)"
340 GETKB$:IFKB$=""THEN340
350 IFKB$="Y"THEN 1
360 END
```

In lines 7 to 9 we have used the routine given at the end of the last article to scatter some stars randomly about the sky. Line 10 places the home planet (a solid blob) in the top right hand corner of the screen and we have changed the saucer's start position in line 30 to increase the distance you have to travel. Line 35 demonstrates how the clock is set to zero at the start of each game.

Because of the way we designed the original program we don't have to change anything in the body of the game; there is, however, a slight addition (just for fun!) in line 200. When the saucer encounters an obstacle the USR routine emits a squeak before RETURNing to the main program. (You could be more blood-thirsty if you wished by changing this to an explosion and game-end routine!) We also have to add line 195 which checks the calculated new position (Q) to see whether the home planet has been reached; if so, it sends the program to the game-end sequence (line 300 onwards).

The end of game message is contained in lines 300 to 320, but note the way that the time taken is shown, simply by printing TI$. If you know how to use the MID$ instruction you could change this to show minutes and seconds or turn it into a score by using VAL(TI$) subtracted from an appropriate figure.

At this stage let me add a personal plea. When you print messages to the screen (as in lines 300 to 330), take the trouble to lay them out neatly. It only means counting up the characters used and then adding spaces to centre the message and it looks so much better than the sloppy, higgledy-piggledy displays, frequently with poor spelling, which appear in so many programs, even commercially produced ones. It takes so little extra time to get it right, but if you don't bother to do it, your untidy display is a permanent part of your program and gives the user a shoddy impression.

The routine in lines 330 to 360 is worth noting as it can be incorporated in almost all the games you design. It simply uses the GET command which we have discussed earlier to restart the game if the player presses the 'Y' key; if any other key is pressed the program ends. For general use in games you may wish to include what is delightfully called a 'mugtrap' - a feature which ensures that the program responds only to the 'Y' or 'N' keys, and ignores any others which may be pressed accidentally. One way of doing this is to change the end as follows:

```
340 GETKB$:IFKB$=""THEN340
350 IFKB$="Y"THEN 1
360 IFKB$="N"THEN END
370 GOTO340
```

Of course this game could be substantially improved by starting with a title frame and going on to offer instructions to the player. You might like to do that for yourself, perhaps using some of the techniques covered in these articles but now we're going on to look at how the PEEK command gets over the problem of a moving object wiping out anything which is in the background.

You will understand by now that you can only have one graphics symbol in any given screen location at any one time. So if we put some 'scenery' on the screen (stars, trees, etc) and we make a moving object go across them the code for the object has to be POKEd into the location which previously held the scenery code. When the object moves on, we normally POKE zero into the location it has just left (otherwise we'd end up with a trail of objects across the screen) and so the space which held the item of scenery is now blank - the tree, etc. will seem to have disappeared.

Let's take a practical example to look at this effect. As a change from flying saucers and stars we'll consider a man walking on a path through a wood.

```
10 PRINT"E"
20 SC=53248
30 FORK=1TO100
40 R=INT(RND(1)*999)
50 POKESC+R,70:NEXTK
60 FORK=521TO558:POKESC+K,212:NEXTK
70 FORI=0TO39:POKE SC+I,208:NEXTI
80 FORI=40TO920STEP40:POKESC+I,208:NEXTI
90 FORI=79TO959STEP40:POKESC+I,208:NEXTI
100 FORI=960TO999:POKESC+I,208:NEXTI
```

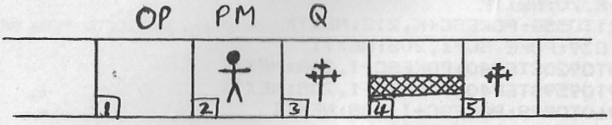
```
110 PM=SC+521
120 POKE PM,202
130 FORJ=1TO75:NEXTJ
140 OP=PM
150 X=0:GOSUB180
160 POKE OP,0
170 GOTO120
180 GETKB$:IFKB$=""THEN180
190 IFKB$="Q"THENX=-41
200 IFKB$="W"THENX=-40
210 IFKB$="E"THENX=-39
220 IFKB$="A"THENX=-1
230 IFKB$="D"THENX=1
240 IFKB$="Z"THENX=39
250 IFKB$="X"THENX=40
260 IFKB$="C"THENX=41
270 Q=PM+X
280 IFPEEK(Q)=208THENRETURN
290 PM=Q
300 RETURN
```

The program should be fairly easy to understand, as it is closely related to our flying saucer programs. Lines 30 to 50 scatter 'trees' (display code 70) about the screen which has a border of shaded squares (display code 208) set up by lines 70 to 100. In between line 60 draws our narrow 'path' (display code 212) across the centre of the screen. The little man (code 202) is placed on the path by lines 110 and 120. His present position is held in the variable PM and again we use OP to denote his old position. The following lines are the routine for keyboard control of movement which should be familiar by now. Again, we use a variable Q to hold the calculated new position of the man and the PEEK function in line 280 checks whether he is about to collide with the border by seeing if location Q contains the border display code.

RUN the program and move the man around.

You will see that path or trees disappear after he has moved over them. Although this effect may be desirable in some games, where the aim is to run over as many objects as possible, in this situation it looks decidedly odd and detracts from our representation of a stroll in the forest.

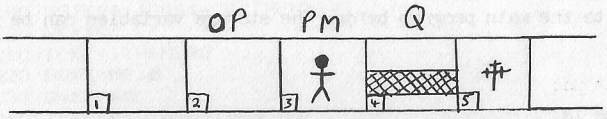
Look at the figure below which represents 5 screen locations:



We are trying to make our little man move to the right. He is presently in the second square, the location of which is held in variable PM, and he came from the first square, whose location is therefore held in variable OP (old position). Moving to the right adds one to his screen location and this will take him into the third square. The computer will have worked this out from the keyboard instructions and will be holding the calculated new position in variable Q. We can see from looking at the screen that there is a tree in the third square which will be wiped out when the man passes over it, so clearly what we need to do is temporarily hold the display code presently in the third square until it can be POKEd back again after the man has passed. We shall put the code initially into a variable W1, and we add the following line to the program.

```
285 W1 = PEEK (Q)
```

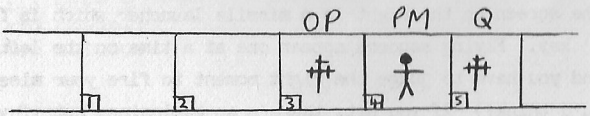
The man can now be moved along, producing the situation below:



But now a problem presents itself. We still want to go on holding the display code for what was in the third square until the man has moved out of it, and we can POKE the scenery back into place, but now we need somewhere to store the code for the object which is in the fourth square while the man moves on to that. The solution is to introduce another variable W2. We can then transfer the information from W1 to W2 so that we can POKE OP, the old position, with W2, while leaving W1 free to receive the next PEEK (Q) information. This is all done by changing line 160 to read:

```
160   POKE OP, W2:W2 = W1
```

In the diagram below the man has reached the fourth square.



W2, which was holding the tree code for the third square, has been used to POKE the tree back. The path code, previously held in W1, is now transferred to W2 as the man goes into the fourth square. W1 is now free to receive the information from the fifth square, which will be provided by PEEK (Q).

There is one more situation we have to consider before the routine is complete. What if the player presses a key other than the direction keys, and the little man stands still for one turn? This produces complications, because the calculated new position, Q, is the same as the present position, PM. So when we PEEK (Q) we are looking into the present location and are bound to see there the code for our little man. If this gets stored in the variables, it will mean that when he eventually moves away, he'll leave behind a mirror image of himself! Not very good!

Fortunately, this can be overcome quite simply by one additional line, which checks whether the calculated and present locations are the same, and if so returns control to the main program before the storage variables can be affected.

The required line is:

```
275       IF Q = PM  THEN RETURN
```

No doubt you will be able to think of many ways of expanding this routine into a full game. For example, you could use the PEEK facility in conjunction with IF ... THEN statements to check for different sorts of terrain (falling into water, quicksands, etc.) or, by using different symbols for each of the borders, you could ensure that movement off an edge led to another appropriate screen.

Finally we go on to the use of PEEK to detect collisions with missiles, and for this we go back to our flying saucer again. The following program brings together many of the routines we have looked at in these articles.

At the bottom of the screen on the right is a missile launcher which is fired by pressing the 'F' key. Flying saucers appear one at a time on the left, at varying heights, and you have to judge the right moment to fire your missile. (Not as easy as you'd think!) If you hit, there's an explosion, but if you

miss, the next saucer appears until you reach the maximum of 10. At the end of the game your score is given, with the chance to try again if you wish.

```
10 SC=53248
20 SS=0
30 FORI=1TO10
40 PRINT"■"
50 FORB=0TO39:POKE SC+B,208:NEXTB
60 FORB=40TO920STEP40:POKESC+B,208:NEXTB
70 FORB=79TO959STEP40:POKESC+B,208:NEXTB
80 FORB=960TO999:POKESC+B,208:NEXTB
90 POKESC+949,216:POKESC+950,208:POKESC+951,215
100 R=INT(RND(1)*10)+1
110 SP=SC+R*40+162
120 MF=0
130 MP=SC+910
140 POKEMP,80
150 FORJ=1TO37
160 POKE SP-1,0
170 POKE SP,199
180 FORK=1TO15
190 GETKB$:IFKB$="F"THENMF=1
200 NEXTK
210 IFMF<1THEN280
220 POKE MP,0
230 Q=MP-40
240 IFPEEK(Q)=208THEN300
250 IFPEEK(Q)=199THEN370
260 MP=Q
270 POKE MP,80
280 SP=SP+1
290 NEXTJ
300 NEXTI
310 PRINT"■■■■■■■■■■■■■■YOU SHOT DOWN";SS;" SAUCERS!"
320 PRINT"■■■■■■■■■■■■■■ANOTHER TRY?  (Y/N)"
330 GETKB$:IFKB$=""THEN330
340 IFKB$="Y"THEN10
350 IFKB$="N"THEN END
360 GOTO330
370 FORX=1TO3
380 POKE SP,107
390 FORY=1TO150
400 POKE4514,Y
410 USR(68)
420 NEXT Y
430 USR(71)
440 POKESP,0
450 FORZ=1TO150:NEXT
460 NEXT X
470 SS=SS+1
480 GOTO300
```

The number of saucers which will appear is set by the FOR ... NEXT loop which starts at line 30. Lines 100 to 110 calculate the random height at which the saucers start, and line 120 sets to zero an indicator (or flag) which shows whether a missile has been fired - in line 190 it is changed to 1 if the 'F' key has been pressed. MP is the missile position and an up arrow (code 80) is used for the missile.

The FOR ... NEXT loop starting at line 150 marks the progress of the saucer across the screen, the saucer position being shown by SP. The FOR ... NEXT loop in line 180 detects whether you have pressed the Fire button, but also acts as a delay loop between each movement of the saucer.

The end of game routine in lines 310 to 360 gives the score. Of course, this has to be reset to zero for a new game, and this is done in line 20. The explosion sequence is from line 370 onwards and line 470 increments the score.

Note particularly lines 230 to 250 in which we use a variable, Q, in which the calculated new position of the missile is held. By PEEK(Q) we are able to look ahead to see whether the missile is about to reach the border (line 240) or hit the saucer (line 250) and direct control to the appropriate part of the program.

The listing as it stands is only a framework which can be developed further into a more playable game. Try modifying it so that the saucers will come on from either side - you'll need to place the missile launcher in the centre of the screen - or alternatively make it possible to move the missile launcher from side to side; vary the number of saucers, or make them unlimited but impose a time limit. Bear in mind, however, that programs written in BASIC must not be too complicated if they are to run smoothly. Since every instruction has to be interpreted into a form which the computer can understand while the program is running, graphics programs which are too complex run too slowly or jerkily to be effective. If your ambition is to write another classic arcade game then you'll have to learn to produce machine code programs!

However, it is hoped that these articles will have given some guidance and assistance to those MZ-80K users who are trying to make some headway with the use of graphics. Although games have been used for all the illustrations, the same techniques can be adopted to produce graphs or diagrams for business or educational use.

Don't worry if you haven't followed all the explanations on the first reading. Even if you don't fully understand the way the individual routines work, you can still incorporate them as complete modules in your own programs - just as a traveller abroad can get by with a phrase book without necessarily being expert in the native grammar!

But do try out the listings on your own computer - you'll find everything becomes much clearer when you can see the examples actually working. Then experiment by making minor changes to the routines and observing the result. The experience will be valuable and you'll be one step nearer writing effective graphics programs of your own.

Finally, I must place on record my thanks to my wife, who devotes evenings and weekends to checking my manuscripts, and my hard-pressed Secretary, who manages to turn my scribbling into a presentable form.

Machine Language Monitors
by Peter Andersson

Author of PA-MON for MZ-80A/K

On the Sharp MZ-80 range of computers the Monitor is used to load BASIC interpreters and other machine language programs. It also contains many standard routines which can be used by other programs.

In the MZ-80K and MZ-80A monitors the only useful commands are the LOAD and GOTO (L and J on the A) commands. Suppose you want to modify or examine a machine language program. How can you do that? Of course, if the program is a BASIC interpreter the PEEK and POKE commands can be used, but most machine language programs are not capable of modifying themselves. One solution is to have a more powerful Monitor which can be used to modify and examine memory. Below a description of what capabilities such a Machine Language Monitor can have is outlined:

To modify the contents of memory one command can, for example, accept hexadecimal numbers as input. When entering machine code programs it is also convenient to have a command which accepts standard Z80 mnemonics as input.

To examine memory a command which displays the contents in memory as, for example, hexadecimal numbers and interpreted as ASCII characters should exist. When a machine language program is examined it is also convenient to have a disassemble command, which translates the contents in memory into Z80 mnemonics.

Commands to load from and save to cassette files should exist.

To find a particular sequence in memory some sort of search command is great.

To make the Monitor even more useful, commands for debugging machine code programs can also be included.

At last, it is convenient if the Monitor is capable of beeing coresident with any machine code program. Therefore a command which can relocate the monitor would be nice.

A Machine Language Monitor as outlined above can be useful in many ways. It can be used to

- Enter machine code programs,
- Modify existing machine code programs,
- Find out how a program works,
- Learn how the computer works,
- Debug machine language programs,

  etc.

PA-MON is a Machine Language Monitor for the MZ-80A and MZ-80K which has all the above capabilities (and more). It consist of 31 different commands and resides in less than 5 kbyte of memory. An example of how a Machine Language Monitor, such as PA-MON, can be used to modify a machine code program now follows:

Suppose that you own a MZ-80A computer and want to be able to use the BASIC SP-5025 interpreter (supplied with the MZ-80K). The MZ-80A and MZ-80K are very similar, so the same machine language program will often run on both of these. However, there are two important hardware

differences, namely the keyboard hardware and the organization of the display memory.

If the SP-5025 is loaded into the MZ-80A it first seems to work properly. However if you try to run a program, the screen behaves rather strange and in most cases the Break key doesn't work. The first problem can be solved if you set the screen in the MZ-80K mode (press CTRL-[), but how to solve the Break key problem? As there are standard routines in both the MZ-80A and K monitor ROM's, which can be used to check the Break key, the Break key should work if this routine is used by the SP-5025. The mentioned routine starts at address 1EH (hexadecimal). To find out if this routine is used by the SP-5025, and if so, why the Break key doesn't work, PA-MON can be used.

After loading, PA-MON resides from address 1200H and upwards, which is the same area occupied by the SP-5025. Therefore the first thing to do is to move PA-MON to high memory. On a 48 kbyte machine PA-MON will fit from address BC00H and upwards. The command to move PA-MON to this address is just

```
*$BC00
```

where the asterix (*) is the prompt used by PA-MON. Note that addresses and other numbers are always entered as hexadecimal numbers. Now the SP-5025 interpreter can be loaded using the Read command:

```
*R
```

PA-MON now tells you that the BASIC is loaded between addresses 1200H and 41FFH. When the loading is completed no auto-run is done, so PA-MON is still in control. Next we want to find the locations where the Break key check routine is called from. The routine is a subroutine, hence it is probably referenced with the instruction

```
CALL 001E
```

The CALL instruction has the code 0CDH followed by the address in low - high byte order. This means that we are searching for sequences of the form

```
CD 1E 00
```

To do this the Search command can be used:

```
*S1200 41FF CD 1E 00
```

This command will search the whole interpreter for the CALL 001E instruction. PA-MON now displays

```
19EC 1AD9
```

indicating that the sequence was found at addresses 19ECH and 1AD9H. Now we can use the Disassemble command to examine the memory area around address 19ECH:

```
*D19E1 19EF
```

This will disassemble the contents of memory between addresses 19E1H and 19EFH. The following is displayed:

```
19E1   LD   HL,E000
19E4   LD   (HL),F8
19E6   INC  HL
19E7   LD   A,(HL)
19E8   INC  A
19E9   JP   Z,19F7
19EC   CALL 001E
19EF   JR   NZ,19F7
```

(Actually a hex dump and character interpretation of the contents of memory are also displayed.)
The code between addresses 19E1H and 19E8H scans the keyboard to see if any of the SHIFT or BREAK keys are pressed, and if so the Break key check routine is called at address 19ECH. This works on a MZ-80K, but as mentioned earlier, the keyboard hardware is different on the MZ-80A and K, so when the SHIFT & BREAK keys are pressed on the MZ-80A, the above routine does not indicate that any of these keys are pressed. Hence the Break key check routine will never be called. (However if one of the left half of the numeric keypad keys is pressed together with the SHIFT & BREAK keys the Break will be recognized.)
This problem can be fixed if the instruction at address 19E4H is altered to

```
LD (HL),F0
```

(The reason for this can be the subject of another article.)
To do this the Z80 mnemonics command can be used:

```
*Z19E4
19E4 LD (HL),F0
19E6
*
```

On the second line 19E4 is displayed by the computer and the desired instruction (LD (HL),F0) is entered by the user. The SHIFT & BREAK keys are used to return to the command level. There is also a command which is used to alter the memory contents using hexadecimal numbers or ASCII characters, which could have been used instead of the Z command.

The code around the second address displayed by the Search command does not have to be altered.

It is also possible to alter the SP-5025 so BASIC programs can be run in MZ-80A screen mode. To do this, alter the instruction at address 1CADH from JP Z,3D17 to JP Z,0015.

When the modifications are finished the new version can be saved to tape using the Write command:

```
*W1200 41FF 1200 BASIC SP-5025
```

This will save the memory area between addresses 1200H and 41FFH. The auto start address becomes 1200H and the filename "BASIC SP-5025".

At last some words about the debug commands in PA-MON:
These consists of commands to alter and display the contents in the Z80 registers, set break-points, and execute machine code programs in different ways.
A Z80 simulator is included, which makes it possible to single-step a machine code program. With the simulator it is also possible to

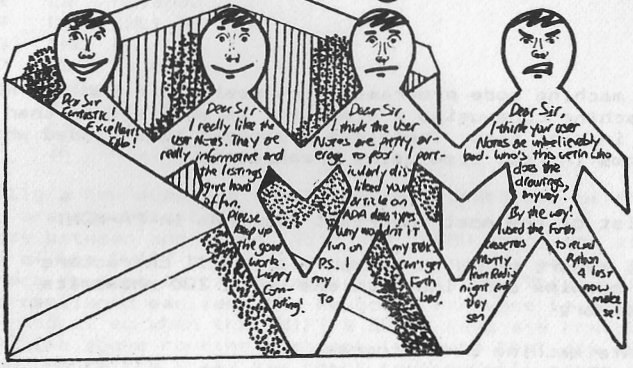
single-step machine code programs which resides in ROM or PROM. Executing machine code using a simulator is much slower than executing the program in real time. Therefore a command is included which executes programs in real time using breakpoints.

Here is a list of the most important commands in PA-MON:

M - Modify memory using hex numbers or ASCII characters
Z - enter machine code instructions using Z80 mnemonics
F - Fill memory
C - Copy
L - reLocate machine code programs
D - Disassemble memory
T - Tabulate memory
W - Write a cassette file
R - Read a cassette file
V - Verify a cassette file
S - Search memory
Y - compare memory blocks
A - Alter register
X - display registers
B - set or show Breakpoints
E - Execute using a Z80 simulator
G - Go, execute in real time using breakpoints
J - Jump to subroutine
$ - relocate PA-MON
# - turn the printer on or off

PA-MON is available from SHARPSOFT
for either the
SHARP MZ-80K
or the
Sharp MZ-80A
at #17.50

LETTERS page



Dear Sirs,

Thank you for your prompt despatch of S.U.N. Issues 1-6 and the fig-FORTH language tapes, which were an unexpected and very welcome surprise.

Your User Notes are marvellous and despite the typographical and paste-up mistakes have been very useful and entertaining.

However, I was very dissappointed to find that the fig-FORTH tape will not load, although the Assembler and Editor tapes do show signs of having dta on them. Please could you replace the enclosed tapes and return them as quickly as possible. My life won't be complete until I can get FORTH running on my MZ-80K!

A suggestion for S.U.N.;

How about a Sharp User Sale/Exchange Section. Much of the Sharp-Marketed interfaces and hardware seem to be very expensive and second hand sales bring this hardware within the reach of mortcage victims such as myself! Oj this subject I would like to see more information about home-brew interface hardware etc., which would extend the 'real-world' capabilities of this excellent machine. I have made some hardware mods myself which I will send to S.U.N. in a future letter.

Congratulations on an excellent user group and magazine, keep-it-up lads!

D. EDWORTHY
E. SUSSEX

Many thanks for your kind words and praise - you may not realise but our low subscriptions make S.U.N. an 'at cost' publication only covering printing, postage and sundry expenses. Any mention of meetings, sales etc. always involves us in a great deal of very time consuming (hence costly) work, by way of telephone calls, letters etc. etc. It would be uneconimcal for Members to purchase 2nd-hand equipment through Sharpsoft as we must charge VAT on top of any asking price, whereas private individuals may sell at an exclusive price.

SHARPSOFT

Dear Editor,

Please find enclosed my subscription for the User Notes, which in my opinion is by far the best publication for the MZ-80K. In my last letter to you with my DISFORTH program I am afraid that there is a bug. I wrongly assumed that all FORTH words threaded down in memory but whilst checking, I have found that certain words thread alh over, therefore the logic in the SEARCH-MC word is wrong and should begin CF @ 2+ CF @ @ = GETPFA.

My only gripe with FORTH is that EMIT is so slow - here is my solution.

```
CREATE EC           CD C, B1 C, OF C, D1 C, 7B C,
                    CD C, B9 C, OB C, 77 C,
                    3E C, C3 C, CD C, DC C, OD C,
                    C3 C, 45 C, 12 C, SMUDGE

: ECHO              DUP 17 > IF EC ELSE NOOP THEN ;
: CONVERT-ECHO      38CF ' ECHO 12 + ! ' ECHO CFA ' EMIT ! ;
: CONVERT-EMIT      28CF ' EMIT ! ;
: CONVERT-EC        ' EC CFA ' EMIT ! ;
```

CONVERT-ECHO converts EMIT to EC all control codes work correctly except the delete key.
CONVERT-EMIT converts back to original EMIT.
CONVERT-EC converts EMIT to EC the control codes are printed in reverse field.

This speeds up EMIT by around 10 times and the screen snow is not much of a problem.

Also in my DISFORTH program there was a new version of ID. this version is okay but it requires VLIST to be redefined plus other words which transmit error codes. Here is my version of REDEFINE perhaps not a sophisticated as the one on that small dedicated FORTHY micro but nonetheless does the same.

```
: INPUT QUERY 1 TEXT PAD NUMBER DROP ; ( INPUT A NUMBER )
0 VARIABLE OLD-CFA
0 VARIABLE NEW-CFA
: REDEFINE

              ." INPUT OLD CFA OF WORD " INPUT OLD-CFA ! CR
              ." INPUT NEW CFA OF WORD " INPUT NEW-CFA ! CR
              HERE 1200 DO I @ OLD-CFA @ = IF
              NEW-CFA @ I ! THEN LOOP ;
```

This leaves both versions of the word in the dictionary but only the last one is used.

Finally I thank you for the definition of PICK in the last Issue of the User Notes, here is my version of ROLL

```
0 VARIABLE ROL
: ROLL        DUP ROL ! PICK ROL @ DUP +
              SP@ 1 + SP@ 2 + SWAP DO
              I 2 - @ I ! - 2 + LOOP DROP ;
```

Please keep up the good work.

R. SHERIDON
CAMBS.

Dear Sir,

Something I have always wanted to do is to change the cursor shape on my MZ-80A. So a friend and I sat down and did it.

As you will probably know the MZ-80A can take CP/M, so the Monitor has to be moved to make room at the begining of memory for CP/M. If you load the A register with (E00CH), the monitor is moved high into memory. Then you have to do a block move to copy the monitor back to its original position in memory. You can now poke into this copied monitor which the MZ-80A will still be using.

The following BASIC program does it all for you.

```
  1 REM Cursor Modifyer
  2 REM by J. Brown
  3 REM & A. Longley

 10 LIMIT ($ BFDF)
 20 POKE $ BFE0, $3A
 30 POKE $ BFE1, $0C
 40 POKE $ BFE2, $E0
 50 POKE $ BFE3, $01
 60 POKE $ BFE4, $FF
 70 POKE $ BFE5, $0F
 80 POKE $ BFE6, $11
 90 POKE $ BFE7, $00
100 POKE $ BFE8, $00
110 POKE $ BFE9, $21
120 POKE $ BFEA, $00
130 POKE $ BFEB, $C0
140 POKE $ BFEC, $ED
150 POKE $ BFED, $B0
160 POKE $ BFEE, $3E
170 POKE $ BFEF, $44
180 POKE $ BFF0, $32
190 POKE $ BFF1, $67
200 POKE $ BFF2, $02
210 POKE $ BFF3, $C3
220 POKE $ BFF4, $50
230 POKE $ BFF5, $12
```

If you now POKE:

$0267 You can change the shape of the normal mode cursor.

$026F You can change the shape of the Graph mode cursor.

$0DA9 You can change the speed of the cursor's movement.

Here is the program in machine code:

```
ADDR        OBJECT          ASSEMBLER

BFE0        3A 0C E0        LD A,(E00CH)
BFE3        01 FF 0F        LD BC, 0FFFH
BFE6        11 00 00        LD DE, 0000H
BFE9        21 00 C0        LD HL, C000H
BFEC        ED B0           LD IR
BFEE        3E 44           LD A. ' '
BFF0        32 67 02        LD(0267H),A
BFF3        C3 50 12        JP   1250H
```

The above code is completely relocatable.

A. LONGLEY
ASHFORD

****

Dear Sir,

I was using my Sharp MZ-80A using a poke statement:

4514,100 (with USR (68)

to create a car-like sound. When I made a mistake, which I did not notice, an• poked 5414,100. I ran the programe as normal and it worked (except the sound was not there). I listed the program to find all the commands like THEN, IF, GOTO, FOR, NEXT had changed to different characters - all totally illegible for BASIC programming.

I then NEWed this programe and wrote a short simple program:

```
1 ?"This is silly"
2 GET X$:IFX$=""THENZ
3 GOTO 1
```

I ran it and it worked normally. But when I listed it, it was changed to:

```
1, Data "THIS IS SILLY"
2  COS (X$:  X$="ATN()
3  LIST 1.
```

I re-rand it and it worked normally! Do you know what the poke statement could be doing? I would be very pleased if you could answer my little mystery. Yours Hopefully,

N. WYLDE
HERTS.

Unfortunately we cannot find out what the strange POKE is doing, other than it is changing the listing of a BASIC program to the screen withut changing the BASIC source program in memory and hence it will still run O.K. I do not know if you have realised but you have stumbled across a simple way of protecting listings of BASIC programs!

SHARPSOFT

Dear Sir,

I have just received Issue No. 8 of the Sharpsoft User Notes and as the owner of an MZ-80A am profoundly disappointed at the content.

Apart from the program on Page 61 which is specifically directed at the MZ-80A one might well conclude that this model was a rare specimen and that only the MZ-80K and MZ-80B were of any account.

Do you intend to redress this imbalance?

K.H. WELLMAN
EXETER

The majority of programs, listings etc. we publish are compatible with the MZ-80A, however, if you look closely at each Issue you will see that the User Notes depend upon contributions from Computer Users, so let's make this a call to **ALL MZ-80A OWNERS - YOUR USER NOTES NEED YOU!**

SHARPSOFT

****

Dear Sirs,

Having read and enjoyed your previous seven issues of the Sharpsoft User Notes, after recently becoming a Member, I feel it is about time I too made a contribution.

Last year while attempting to make a 3D cube rotation program I stumbled across the need for a routine to draw a straight line and to do it fast, not being a master with machine code, I developed the following BASIC routine.

(MR. WELLMAN THIS WILL WORK ON YOUR MZ-80A)

```
10 IF X1=X2 THEN X2=X1 + 1E-3
20 IF Y1=Y2 THEN Y2=Y1 + 1E-3
30 GR=X2-X1:GD=Y2/GR - Y1/GR
40 GY=(ABS(GD)>1) * SGN(Y2-Y1) : GX=(GY=0)
50 S = -SGN(GR) * GX - 1/GD * GY
60 GT = X1 * GD-Y1 : FOR X4 = X1 TO X2 STEP S
70 SET X4,X4 * GD-GT : NEXT : RETURN.
```

The routine draws a line between the points (X1,Y1) and (X2,Y2). The variables X1, Y1, X2 and Y2, must be provided before entering the routine.

In order to maintain the routine's speed no range checking is done on the co-ordinates.

I have made this routine as fast as possible but if anybody has another routine faster, or can speed this up I would be glad to hear about it.

A TIP: POKE 59555,6 : POKE 59555,7

Turns tape recorder motor off if already on!

POKE 59555,22 : POKE 59555,23

Turns tape recorder motor on if tape keys PLAY, PLAY RECORD, FWD, REW are already depressed.

I am surprised that no one else seems to have found these pokes. I hope the above information is useful to fellow Sharp Users. I now await the next edition of the User Notes eagerly.

K. KHAN
BIRMINGHAM

****

Dear Sirs,

Having received all of the back issues ordered, I was pleasantly surprised at the amount of software held within. I commenced typing straight away.

It was at this point that I realised a small loophole in the utilities given (Issue 3). This is to save the program without the utility, one must delete it line by line. This is both time consuming and tedious. Thus I have written a utility to delete lines. It is short enough to be typed in or could be loaded via append and then delete the append program by hand.

```
1 LIMIT53000:FORI=0TO6:READD:POKE53000+I,D:
  NEXT:DATA245,197,229,213,195,91,8
2 INPUT"DELETE FROM:";A:INPUT"TO:";B:INPUT"STEP";S
3 POKE59555,0:FOR C=A TO B STEP 5:A$=STR$(C):PRINT
  "C    ";A$:PRINT:PRINT:PRINT
4 PRINT "GOTO 6"
5 OL=PEEK(4738):OH=PEEK(4739):PRINT"H";:POKE 4738,OL:
  POKE4739,207:STOP
6 POKE4738,OL:POKE4739,OH:NEXT:PRINT "C":POKE59555,1:END
```

NOTE Lines 3, 5 and 6 "C", "C ","H" refers to Clear, Home and cursor down symbols.

The program must be typed without any spaces.

Also Locn. 4634 Hex = BASIC pointer to next available space to list tokenised BASIC line.

Listing 4634 H to 4646 H gives a series of numbers. These refer to BASIC pointers to variable positions etc. Altering 4634 H alters end of program position.

Note Delete works quickest if program to be deleted has equal increments i.e. lines, 10, 20 etc. thus to delete utilities some lines may still have to be hand deleted.

H.G. SOPER
CLEVELAND

Dear Sharpsoft,

Here are a couple of quite useful hints for the MZ-80K.

1. Making a pause in the program and waiting for any key to be pressed seems to be solved everywhere in an uneconomical and complicated way, e.g.

```
650 GET A$:IF A$=""GOTO 650
```

A much simpler:

```
650 USR(2483)
```

will perform the same function!

2. Conversions between "ASCII" and display codes does not have to be done through a large conversion table dimensioned and filled by your BASIC program.

```
10 DEF FNA(X) = PEEK (3269+X):REM display code to ASCII
20 DEF FND X  = -(X>15)*PEEK(3014+X):REM ASCII to display code
```

These functions make use of conversion tables that are stored in the ROM and ready to be accessed not not only by the monitor I/O routines but by PEEKs as well.

3. Two other user-defined functions enable easy access to the current cursor position:

```
DEF FNR(U) = PEEK (4466):REM Current row 0...24
FEF FNC(U) = PEEK (4465):REM Current column 0...39
```

In both cases U is a dummy parameter and may be set to anything, e.g. 0.

4. Instead of outputting a long series of cursor control characters /"arrows"/ one can directly set cursor position on a given column XP and row YP by means of the following subroutine:

```
9000 REM --- XP=column, YP=row
9010 POKE 4465,XP:POKE 4466,YP:RETURN.
```

Could any of the MZ-80K owners supply me an annotated listing of machine code routines for accessing disc files - in the first hand routines for reading and writing a given disk sector on a given track.

A disassembly of ROM disc routines for MZ-80K with comments wold be very welcome too! I am prepared to pay for these pieces of information.

My kind regards and best wishes to the Sharpsoft Staff. I wished your newsletters appeared at least monthly....

M. WIECHOWSKI
SWEDEN

Dear Sharpsoft,

I have your Sharp MZ-80K Computer and am a Raw Amateur at the moment.

I have produced a copy of the BASIC SP-5025 tape, and have added the speed BASIC toolkit.

I have a problem with the ?@ command.

When I type in ?@4,2;"K" it correctly performs the function required, but it will not list. When I give it a line number, say 100, and type in 100 ?@4,2;"K" it will not perform the function, but it will list. (The result is I cannot transfer to a fresh tape).

I have your book Peeking and Pokeing the Sharp MZ-80K and have experimented with the 2 programs on page 16 and the one on page 17. The program at the bottom of page 16 reveals a syntax error in line 50, and the program on page 17 reveals a syntax error in Line 110. I have no success with the program with line nos: 1 2 3 4 5 6 7 8 on page 16.

I have your Sharpsoft Notes 1 to 7 and various other books purchased from you but I am afraid I cannot find any solution.

Is it possible to get a speedy answer?

W.W. BARNES
MIDDX.

****

Dear Sharpsoft,

Thank you for sending me the Sharpsoft User Notes No. 7. How nice it was to find out that I am not the only one who has a SEIKOSHA GP-80D printer. At the moment I am working on a program to do 3-D plots using this printer. I will pass them to you as soon as they are completed.

I have found out that many MZ-80K owners have had problems with some of the keys and also had trouble obtaining them including me! I then came across the MAPLIN ELECTRONIC SUPPLIES catalogue and came to the SWITCH section when I saw an identical key switch as used in the MZ-80K keyboard. If other readers are desperate to replace a 'DEAD' key switch, the order code is FF6IR (KEYBOARD SWITCH) and cost 23p including the dreaded VAT. And they quite rightly call it 'A LOW COST PUSH SWITCH'.

I hope this has calmed some MZ-80K owners' hearts.

I have also enclosed some program listings which I hope will be printed in future issues of the USER NOTES.

A.K. NATHEKAR
BIRMINGHAM

****

```
10 POKE10167,1
20 GOSUB830:GOSUB610
30 DIMM$(4)
40 FORA=0TO4:READM$(A):NEXT
50 PRINT"■"
60 A=53649:B=202:C=0:E=53649:F=53685
70 POKEA,B:POKEA,C:GETB$:D=PEEK(17828)
80 X=INT(RND(1)*5)
90 PRINTM$(X)
100 IFD=ASC(L$)THENPOKEA,C:A=A-1
110 IFD=ASC(K$)THENPOKEA,C:A=A+1
120 IFA<ETHENA=A+1
130 IFA=FTHEN190
140 IFPEEK(A+40)=201THEN430
150 POKEA-40,PEEK(A+40)
160 POKEA,B
170 FORS=1TO2:NEXT
180 POKEA,B:POKEA,C:GOTO70
190 POKEA,B
200 FORJ=1TO255:POKE4513,J:POKE4514,1:USR(68):NEXT:USR(71)
210 POKE59555,0:PRINT"■■■■■■■■■■■■■■■"
220 PRINT"          SEEMS LIKE YOU USED■
230 PRINT"                 THE■
240 PRINT"           GREEN CROSS CODE.■
250 PRINT"
260 PRINT"
270 PRINT"
280 PRINT"
290 PRINT"
300 PRINT"
310 FORY=1TO10
320 POKE59555,0
330 FORJ=1TO100:NEXT:POKE59555,1
340 FORJ=1TO100:NEXTJ,Y
350 FORJ=1TO1500:NEXT
360 PRINT"■"
370 POKE4465,14:POKE4466,10
380 PRINT"ANOTHER GO?"
390 GETF$:IFF$=""THEN390
400 IFF$="Y"THENPRINT"■":USR(62):GOSUB610:GOTO50
410 IFF$="N"THENUSR(62):END
420 GOTO390
430 POKEA,B
440 FORJ=0TO255
450 POKE4513,255:POKE4514,J :USR(68)
460 NEXT:USR(71)
470 PRINT"■"
480 FORJ=1TO10
490 POKE4465,13:POKE4466,10
500 PRINT"YOU CRASHED!!!"
510 FORQQ=1TO100:NEXT
520 POKE4465,13:POKE4466,10
530 PRINT"                "
540 FORQQ=1TO100:NEXT:NEXT
550 GOTO360
560 DATA |   | ■ |   | ■ | ■ |   |   |   |   |*
570 DATA | ■ |   | ■ |   |   | ■ | ■ |   | ■ |*
580 DATA |   |   |   |   |   |   |   |   |   |*
590 DATA | ■ |   |   |   | ■ |   | ■ |   | ■ |*
600 DATA |   |   |   |   |   |   |   |   |   |*
610 PRINT"■"
620 PRINT"■WHAT SKILL LEVEL WOULD YOU LIKE  (1-9)"
```

```
630 PRINT"■ 1-Seb Coe!!
640 PRINT"■ 2-Steve Ovett!
650 PRINT"■ 3-Getting Even Better!
660 PRINT"■ 4-European Runner.
670 PRINT"■ 5-Test Passed.
680 PRINT"■ 6-Getting Better.
690 PRINT"■ 7-Learner.
700 PRINT"■ 8-Just Joined The A.A.A.
710 PRINT"■ 9-Chicken!!■
720 USR(62)
730 INPUTZ:USR(62)
740 IF(Z<1)+(Z>9)THEN610
750 IFZ=1THENPRINT"■":GOSUB970
760 FORJ=1TO2500:NEXT
770 Z=Z*10
780 PRINT"■"
790 INPUT"■■■■WIHCH KEY WILL BE LEFT ";L$:USR(62)
800 INPUT"■WHICH KEY WILL BE RIGHT ";K$:USR(62)
810 RETURN
820 END
830 PRINT"■              AUTOBAHN
840 PRINT"
850 PRINT"What, Not heard of AUTOBAHN?"
860 PRINT"■Well you have now."
870 PRINT"■Now I'l give you the data of■            how to play this game.
880 PRINT"■You (";CHR$(99);") have to cross a very busy■       autobhan,";
890 PRINT"(motorway for those who don't■  posses the German tounge.)"
900 PRINT"■The man only moves left and right,
910 PRINT"■and which keys to control the man depend■on you."
920 PRINT"■               GOOD LUCK!!"
930 PRINT"              Press any key"
940 GET0$:IF0$=""THEN940
950 PRINT"■":RETURN
960 END
970 PRINT"■"
980 POKE4465,0:POKE4466,10
990 PRINT"Pushing your luck a bit far aren't you?"
1000 RETURN
1010 END
```

```
10 PRINT"■":GOSUB 570
20 C$="■■■■":Z$="■■■■■":A$="■"
30 PRINT"■"
40 DIMA(4)
50 A(1)=INT(9*RND(1)+1)
60 A(2)=INT(9*RND(1)+1)
70 IFA(2)=A(1)THEN60
80 A(3)=INT(9*RND(1)+1)
90 IF(A(3)=A(1))+(A(3)=A(2))THEN80
100 A(4)=INT(9*RND(1)+1)
110 IF(A(4)=A(1))+(A(4)=A(2))+(A(4)=A(3))THEN100
120 PRINTTAB(12);"■ ■ ■ ■"
130 FORYU=1TO7
140 INPUT"■■■■■ ";M
150 GOSUB340
160 INPUT"■■■■■ ";N
170 GOSUB380
180 INPUT"■■■■■ ";B
190 GOSUB420
200 INPUT"■■■■■ ";V
210 GOSUB460
220 C$=C$+A$:Z$=Z$+A$
```

```
230 IF(M=A(1))*(N=A(2))*(B=A(3))*(V=A(4))THEN500
240 NEXTYU
250 PRINT"■You failed to break the code."
260 PRINT"■It was:-■"
270 PRINTA(1);A(2);A(3);A(4)
280 PRINT" ▀▀▀▀▀▀▀▀      "
290 PRINT"■Do you want another go?(Y/N)"
300 GETZ$:IFZ$=""THEN300
310 IFZ$="Y"THEN20
320 IFZ$="N"THENEND
330 GOTO300
340 PRINTC$:PRINTTAB(11);M
350 IFM=A(1)THENPRINTZ$;TAB(22);"O"
360 IF(M=A(2))+(M=A(3))+(M=A(4))THENPRINTZ$;TAB(22);"●"
370 RETURN
380 PRINTC$:PRINTTAB(13);N
390 IFN=A(2)THENPRINTZ$;TAB(23);"O"
400 IF(N=A(1))+(N=A(3))+(N=A(4))THENPRINTZ$;TAB(23);"●"
410 RETURN
420 PRINTC$:PRINTTAB(15);B
430 IFB=A(3)THENPRINTZ$;TAB(24);"O"
440 IF(B=A(1))+(B=A(2))+(B=A(4))THENPRINTZ$;TAB(24);"●"
450 RETURN
460 PRINTC$:PRINTTAB(17);V
470 IFV=A(4)THENPRINTZ$;TAB(25);"O"
480 IF(V=A(1))+(V=A(2))+(V=A(3))THENPRINTZ$;TAB(25);"●"
490 RETURN
500 PRINTTAB(12);"■WELL DONE!!"
510 PRINT"■You cracked the code in";VU;" goes."
520 PRINT"■Do you want another go(Y/N)?"
530 GETZ$:IFZ$=""THEN530
540 IFZ$="Y"THEN20
550 IFZ$="N"THENEND
560 GOTO530
570 PRINTTAB(12);"MASTERMIND"
580 PRINTTAB(12);"▀▀▀▀▀▀▀▀▀▀"
590 PRINT"■ This is the game of MASTERMIND."
600 PRINT"■You have to crack the code I have"
610 PRINT"■set."
620 PRINT"■But BEWARE!!!! you have only 7 tries."
630 PRINT"■If you set a number correct I will "
640 PRINT"■print the following:-"
650 PRINT"■●:CORRECT NUMBER BUT IN WRONG PLACE"
660 PRINT"■O:CORRECT NUMBER AND IN RIGHT PLACE"
670 PRINT"■■       PRESS ANY KEY TO START"
680 GETA$:IFA$=""THEN680
690 RETURN
700 REM **********************
710 REM *    MASTERMIND    *
720 REM *        BY        *
730 REM *  ABDUL NATHEKAR  *
740 REM **********************
750 END
```

Dear Sharpsoft,

My new game, I recieved Centipede, is superb it has good graphics and sound, my brother has got the new high-score 245,6384 my brother is age 10. I am saving up to buy Printplot as it sounds good, I like writing programs but I get stick when I get to moving the characters, please could you help me out.

S. JAMES (age 11)
ESSEX

If you have been following J.A. Simpson's articles "A SIMPLE GUIDE TO SHARP GRAPHICS" (S.U.N. Issues 8 and 9) I'm sure they will help and in Issue 10 Mr. Simpson is producing a follow-on entitled "QUICK ON THE DRAW" relating to printing on the screen.

****

Dear Sir/Madam,

Please could you tell me if I have beaten the current record for your ASTEROIDS m/c game. My score is 52,090 & I obtained this on 16/5/83 the game took a period of 31 minutes. If I have beaten the record can you please reply to me, I would be most grateful. Also if it is possible could you please print my name and score in your catalogue.

I have definitely beaten the score in your current catalogue for it only reads a score of 50,490/ I also have a witness to my score.

J. KUTTE (Age 13)
COTTINGHAM

Can anyone confirm that Jonathan's score has been beaten!

EDITOR

****

Dear Sirs,

I recently took delivery of your Sharpsoft User Notes including all the back numbers. I wish I had known about them earlier. Whilst I am not in the same league as most of your contributors I have written 2 programs that might be of interest to some other readers. The first is a Graph Plot routine, the second a calculating matrix 'Calcmate'.

Graph Plot gives a mutliple graph or bar chart facility. It automatically resets axis values if too high a value is entered. It is possible to append this to the 'Wordprocess' program you published. With a suitable menu subroutine this makes a very powerful tool.

Calcmate allows the user to insert data anywhere in a formatted matrix, it facilitates the insertion of column relationship formulae. It also allows columns to be displayed or printed in any order over any block of lines selected, formatting being automatic. Because it is written in BASIC it can be slow if too many columnm relationships are used in one program. Despite this I find it saves me writing specialist programs as I use this as the basis for most calculating and table preparation.

For convenince I have joined them together bit it is easy to see at what point to split them. I have inserted a number of REM statements and these include some examples which of course should be removed.

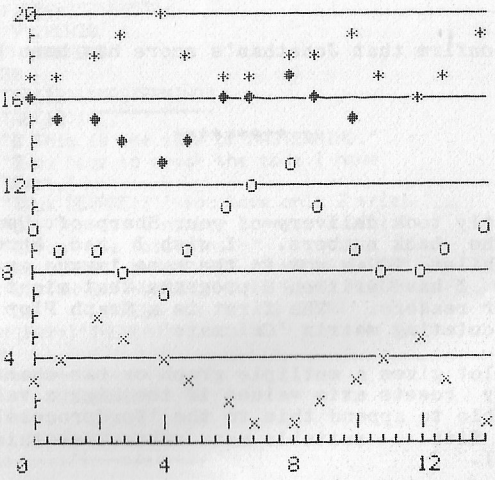
Keying in is rather lengthy but usage is fairly self-evident.

Both programs are written using Speed BASIC but the only change needed to run on standard BASIC is PRINT AT has to be replaced.

One problem I have not solved is how to get SET to print on hardcopy as these are not ASCII characters.

M.G. KEEN
MIDDX.

EXAMPLES OF SCREEN DISPLAYS



TITLE PRINTED HERE

```
 1  TITLE COL        45  ------     5.4
 2  is T             36  ------     4.32
 3  ----------       26  ------     3.12
 4  Entry is by      15  ------     1.8
 5  referance es.   100  ------     12
 6  T3 REQD.TITLE   143  ------     17.16
 7  enters in T3     66  ------     7.92
```

TYPE IN REFERENCE ---- THEN NEW DATA

```
10 DIM A(25)
12 PRINT"■"
14 PRINT"■■■THIS PROGRAM CONTAINS TWO WHICH MAY BE "
15 PRINT"■                  USED"
16 PRINT"■ EITHER SEPERATLY OR IN COMBINATION"
18 PRINT"■■A          CALCMATE"
20 PRINT"■B         GRAPH PLOT "
22 INPUT"■■          A or B     ? ";AS$
24 IF AS$="A"GOTO 580
26 PRINT"■■■"
28 FOR X=1TO 39:PRINT"*";:NEXT X
30 PRINT
32 PRINT"*                                          *"
34 PRINT"*                                          *"
36 PRINT"*              GRAPH PLOTTER               *"
38 PRINT"*                   BY                     *"
40 PRINT"*                M.G.KEEN                  *"
42 PRINT"*                                          *"
44 PRINT"*                                          *"
46 FOR X=1TO 39:PRINT"*";:NEXT X
48 PRINT"■■"
50 PRINT"GRAPH OR CHART TITLE■"
52 INPUT TL$
54 FOR X=1TO 39:PRINT"-";:NEXT X
56 PRINT
58 PRINT "GRAPH OR BAR CHART      (G OR B)   ";
60 INPUT Z$
62 PRINT"■■DO YOU REQUIRE HARD COPY  (Y OR N) ";
64 INPUT PR$
66 IF PR$><"Y" THEN PR$="N"
68 PRINT"■"
70 PRINT"FULL SCALE REQUIRED (1 or more)"
72 PRINT"FOR VERTICAL AXIS                  ";
74 INPUT P
76 IF P<1THENP=1
78 IF Z$="B" THEN A$="N"
80 IF Z$="B" THEN WA=2
82 IF Z$="B" GOTO 98
```

```
84 PRINT"■■■STEP VALUE"
86 PRINT"FOR HORIZONTAL AXIS                    ";
88 INPUT Q
90 PRINT"■■■Do you want interpolation (Y or N)";
92 INPUT I$
94 PRINT"■■■Do you want multiple graph plot    ";
96 INPUT A$
98 IF Z$="B" THEN PRINT"■BAR CHART SELECTED ":PRINT:GOTO 102
100 PRINT"■":PRINT:PRINT"DATA FOR FIRST PLOT " :PRINT
102 IF Z$="B"THEN PRINT "Maximum number of bar plots       18"
104 PRINT:PRINT"Vertical scale chosen                    ";P:PRINT
106 PRINT"Number of samples                      ";
108 INPUT N
110 FOR X=1TO 39:PRINT"-";:NEXT X
112 DIM A(N)
114 FOR X = 1 TO N
116 GOSUB 566
118 INPUT A(X)
120 IF A(X)>P THEN P=A(X)
122 NEXT X
124 IF Z$="B"THEN PH= 1: GOTO 204
126 IF A$ = "N" THEN GOTO 186
128 PRINT"■":PRINT:PRINT"DATA FOR SECOND GRAPH":PRINT
130 GOSUB 558
132 INPUT L
134 DIM B(L)
136 FOR X = 1 TO L
138 GOSUB 566
140 INPUT B(X)
142 IF B(X)>P THEN P=B(X)
144 NEXT X
146 GOSUB 544
148 PRINT"■":PRINT:PRINT"DATA FOR THIRD GRAPH":PRINT
150 GOSUB 558
152 INPUT M
154 DIM C(M)
156 FOR X = 1 TO M
158 GOSUB 566
160 INPUT C(X)
162 IF C(M)>P THEN P = C(M)
164 NEXT X
166 GOSUB 544
168 PRINT"■":PRINT:PRINT"DATA FOR FOURTH GRAPH":PRINT
170 GOSUB 558
172 INPUT O
174 DIM D(O)
176 FOR X = 1 TO O
178 GOSUB 566
180 INPUT D(X)
182 IF D(X)>P THEN P=D(X)
184 NEXT X
186 IF N>19 GOTO 512
188 IF L>19 GOTO 512
190 IF M>19 GOTO 512
192 IF O>19 GOTO 512
194 IF N> 8 GOTO 518
196 IF L> 8 GOTO 518
198 IF M> 8 GOTO 518
200 IF O> 8 GOTO 518
202 WA = 5
```

```
204 PRINT"E";
206 PRINT"  ├──────────────────────────────────────────"
208 PRINT"  ├"
210 PRINT"  ├"
212 PRINT"  ├"
214 PRINT"  ├──────────────────────────────────────────"
216 PRINT"  ├"
218 PRINT"  ├"
220 PRINT"  ├"
222 PRINT"  ├──────────────────────────────────────────"
224 PRINT"  ├"
226 PRINT"  ├"
228 PRINT"  ├"
230 PRINT"  ├──────────────────────────────────────────"
232 PRINT"  ├"
234 PRINT"  ├"
236 PRINT"  ├"
238 PRINT"  ├──────────────────────────────────────────"
240 PRINT"  ├"
242 PRINT"  ├"
244 GOSUB 456
246 IFZ$="B"THEN PRINT"  └──────────────────────────────────────────":GOTO 250
248 PRINT"  └──────────────────────────────────┘"
250 REM VERTICAL SCALE
252 POKE 4466,0:PRINT P
254 POKE 4466,21:PRINTTAB(1);"0"
256 PA =(P/5)*4:IFPA=<10THEN PA=INT(PA*10)/10
258 PA =(P/5)*4:IFPA >10THEN PA=INT(PA)
260 PB =(P/5)*3:IFPB=<10THEN PB=INT(PB*10)/10
262 PB =(P/5)*3:IFPB >10THEN PB=INT(PB)
264 PC =(P/5)*2:IFPC=<10THEN PC=INT(PC*10)/10
266 PC =(P/5)*2:IFPC >10THEN PC=INT(PC)
268 PD =(P/5)  :IFPD=<10THEN PD=INT(PD*10)/10
270 PD =(P/5)  :IFPD >10THEN PD=INT(PD)
272 PA$=STR$(PA):PB$=STR$(PB):PC$=STR$(PC):PD$=STR$(PD)
274 IF P=<1THEN PA$=RIGHT$(PA$,2)
276 IF P=<1THEN PB$=RIGHT$(PB$,2)
278 IF P=<1THEN PC$=RIGHT$(PC$,2)
280 IF P=<1THEN PD$=RIGHT$(PD$,2)
282 POKE 4466,4:PRINT PA$
284 POKE 4466,8:PRINT PB$
286 POKE 4466,12:PRINT PC$
288 POKE 4466,16:PRINT PD$
290 IF Z$="B"GOTO 304
292 REM HORIZONTAL SCALE
294 QA=Q*2:QB=Q*3:QC=Q*4:QD=Q*5
296 QE=Q*6:QF=Q*7:QG=Q*8:QH=Q*9
298 QI=Q*10:QJ=Q*11:QK=Q*12:QL=Q*16:QM=Q*20:QN=Q*24:QO=Q*28:QP=Q*32:QQ=Q*36
300 IF WA<5 THEN GOTO  478
302 IF WA =5 THEN GOTO 502
304 REM CALCULATE AND INSERT VALUES
306 W=2:A(K)=0:JJ=2:IF N>9 THEN NX=4
308 IF N<10 THEN NX=7
310 FOR X = 1 TO N
312 AK=(A(K)+A(X))/2
314 A(K)=A(X)
316 AK  = AK   /(P/20)
318 A(X)= A(X) /(P/20)
320 AK =INT(AK)
```

```
322 A(X)=INT(A(X))
324 AA=(800-(40*A(X)))+W
326 KK=(800-(40*AK))+W-(WA/2)
328 IF Z$="G"THEN  POKE 53248+AA,152
330 IF (I$="Y")*(X>1)THEN POKE 53248+KK,152
332 IF Z$="B" THEN GOSUB 522
334 W=W+WA
336 NEXT X
338 W=2:BK=0
340 IF L<1 THEN 428
342 FOR X = 1 TO L
344 BK=(B(K)+B(X))/2
346 B(K)=B(X)
348 BK = BK/(P/20)
350 B(X)=B(X)/(P/20)
352 BK=INT(BK)
354 B(X)=INT(B(X))
356 BB=(800-(40*B(X)))+W
358 KK=(800-(40*BK))+W-(WA/2)
360 POKE 53248+BB,72
362 IF (I$="Y")*(X>1) THEN POKE 53248+KK, 72
364 W=W+WA
366 NEXT X
368 W=2:CK=0
370 IF M<1 THEN 428
372 FOR X = 1 TO M
374 CK=(C(K)+C(X))/2
376 C(K)=C(X)
378 CK = CK/(P/20)
380 C(X)=C(X)/(P/20)
382 CK=INT(CK)
384 C(X)=INT(C(X))
386 CC=(800-(40*C(X)))+W
388 KK=(800-(40*CK))+W-(WA/2)
390 POKE 53248+CC,68
392 IF (I$="Y")*(X>1) THEN POKE 53248+KK,68
394 W=W+WA
396 NEXT X
398 W=2 :DK=0
400 IF O<1 THEN 428
402 FOR X = 1 TO O
404 DK=(D(K)+D(X))/2
406 D(K)=D(X)
408 DK = DK/(P/20)
410 D(X)= D(X) /(P/20)
412 DK = INT(DK)
414 D(X)=INT(D(X))
416 DD=(800-(40*D(X)))+W
418 KK = (800-(40*DK))+W-(WA/2)
420 POKE 53248+DD,107
422 IF (I$="Y")*(X>1) THEN POKE 53248+KK,107
424 W=W+WA
426 NEXT X
428 PRINT@ 8,23;TL$
430 IF PR$="Y" THEN PRINT/S
432 GET T$:IF T$="" THEN 432
434 PRINT"■":PRINT  "■■■■■■■■ DO YOU REQUIRE FURTHER PLOTS
436 INPUT"■ TYPE   Y or N                                  ";A$
438 PRINT"■■"
440 IF A$="Y" THEN 58
```

```
442 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■THANK YOU"
444 FOR T=1 TO 400:NEXT T
446 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■GET NEW PROGRAM"
448 FOR T=1 TO 400:NEXT T
450 PRINT"■■■■■■■■■■■■■■■■■■■■ OR SWITCH OFF"
452 FOR T=1 TO 400:NEXT T
454 GOTO 442
456 IF Z$="B" THEN 474
458 IF WA < 5 THEN 464
460 IF WA = 5 THEN 470
462 RETURN
464 PRINT"  ├   |   |   |   |   |   |   |   |   |"
466 RETURN
468 FOR X=1 TO N:IF EX(X)>0 THEN A(X)=EX(X):NEXT X
470 PRINT"  ├    |    |    |    |    |    |    |    |"
472 RETURN
474 PRINT"  ├ "
476 RETURN
478 POKE 4466,21:PRINTTAB(((WA*4)+1));QC
480 POKE 4466,21:PRINTTAB(((WA*8)+1));QG
482 POKE 4466,21:PRINTTAB(((WA*12)+1));QK
484 POKE 4466,21:PRINTTAB(((WA*16)+1));QL
486 PRINT@ 8,23;TL$
488 IF WA=2 THEN 306
490 POKE 4466,21:PRINTTAB(((WA*20)+1));QM
492 POKE 4466,21:PRINTTAB(((WA*24)+1));QN
494 POKE 4466,21:PRINTTAB(((WA*28)+1));QO
496 POKE 4466,21:PRINTTAB(((WA*32)+1));QP
498 PRINT@ 8,23;TL$
500 GOTO 306
502 POKE 4466,21:PRINTTAB(((WA* 2)+1));QA
504 POKE 4466,21:PRINTTAB(((WA* 4)+1));QC
506 POKE 4466,21:PRINTTAB(((WA* 6)+1));QE
508 PRINT@ 8,23;TL$
510 GOTO 306
512 WA = 1
514 I$ = "N"
516 GOTO 204
518 WA = 2
520 GOTO 204
522 REM BAR CHART PLOT
524 V=(40-(A(X)*2))+1
526 JJ=JJ+NX:QQ=JJ
528 FOR QQ=QQ TO QQ+3
530 FOR F=V TO 40
532 IF((QQ=JJ)*(F=V))*(PH)9)*(INT(PH/2))<PH/2)THEN PH =PH+1 :GOTO 536
534 IF (QQ=JJ)*(F=V)THEN PRINT @((QQ/2)-1),21 ;PH:PH =PH+1
536 SET QQ,F
538 IF(F=40)*(QQ=JJ+2)THEN RETURN
540 NEXT F,QQ
542 RETURN
544 PRINT:PRINT"Do you require display yet"
546 PRINT"      or to enter another set of data"
548 PRINT"              Type D for display"
550 PRINT"                   M for more data";
552 INPUT B$
554 IF B$ = "D" GOTO  186
556 RETURN
558 PRINT:PRINT"Vertical scale now                    ";P:PRINT
```

```
560 PRINT"Will change to max input fisure if greater   than that shown above"
562 PRINT"■Number of samples                          ";
564 RETURN
566 PRINT:PRINT"ENTER SAMPLE VALUE ";X;"              ";
568 RETURN
570 REM
572 REM
574 REM
576 REM
578 REM
580 REM
582 PRINT"■■■"
584 FOR X=1TO 39:PRINT"*";:NEXT X
586 PRINT
588 PRINT"*                                          *"
590 PRINT"*                                          *"
592 PRINT"*                  CALCMATE                *"
594 PRINT"*                     BY                   *"
596 PRINT"*                  M.G.KEEN                *"
598 PRINT"*                                          *"
600 PRINT"*                                          *"
602 FOR X=1TO 39:PRINT"*";:NEXT X
604 GOTO 642
606 REM ******************************
608 REM ********* FORMULA ************
610 REM ******************************
612 REM
614 PRINT"■"
616 LIST   606- 5095
618 REM  INSERT FORMULA FOR COL RELATIONSHIPS --- USE LINE NOS 5030T05095
620 REM EXAMPLE ON NEXT 4 LINES
622 REM **** CALC B$ ****
624 A1=VAL(A$(T1))
626 A1=(A1/100)*12
628 B$(T1)=STR$(A1)
630 REM
632 REM
634 REM
636 REM
638 REM
640 RETURN
642 PRINT:PRINT:INPUT  "■■■■■■TITLE          ";TL$
644 PRINT:PRINT:INPUT "■■■■■■DATE          " ;DT$
646 PRINT"■■■■DO YOU WISH TO READ OUT DATA FROM STORE "
648 PRINT"■■     OR INPUT NEW DATA FROM START"
650 PRINT"■■■          A  for NEW START"
652 PRINT"■■          B  for STORED DATA "
654 PRINT:PRINT
656 INPUT"                  ";DS$
658 REM
660 REM ****************************
662 REM *** SET RANGE OF MATRIX ***
664 REM ****************************
666 REM
668 DIM Z$(25),N$(3),P(25) :GG=1000
670 DIM A$(25),B$(25),C$(25),D$(25)
672 DIM E$(25),F$(25),G$(25),H$(25)
674 DIM I$(25),J$(25),K$(25),L$(25)
676 DIM T$(25),M$(15),W(25),H(25)
678 IF DS$="B" THEN GOTO 684
680 IF DS$="A" THEN GOTO 706
682 GOTO 646
```

```
684 REM
686 REM ***************************
688 REM *** READ DATA FROM FILE ***
690 REM ***************************
692 REM
694 ROPEN
696 FOR T1=1 TO 25
698 INPUT#T A$(T1),B$(T1),C$(T1),D$(T1),E$(T1),F$(T1),G$(T1)
700 INPUT#T H$(T1),I$(T1),J$(T1),K$(T1),L$(T1),T$(T1)
702 NEXT T1
704 CLOSE
706 REM
708 REM ***********************
710 REM *** COLUMN  SELECT ***
712 REM ***********************
714 PRINT"■■ENTER COLUMN REFERANCE LETTER■■"
716 PRINT"(Title column 'T' comes automaticaly)"
718 INPUT"■■■■■■ First Column?           ";N$(1)
720 INPUT"■■■■■■ Second Column?          ";N$(2)
722 INPUT"■■■■■■ Third Column?           ";N$(3)
724 REM
726 REM ***********************
728 REM *** LINE NO SELECT ***
730 REM ***********************
732 PRINT"■■ENTER LINE NUMBERS (any 7 lines or less)■■■■"
734 INPUT"■■■■■■ First Line?   ";O1
736 INPUT"■■■■■■ Last Line?    ";O2
738 IF O2-O1>7 THEN GOTO 734
740 REM
742 REM ***********************
744 REM *** SCREEN DISPLAY ***
746 REM ***********************
748 REM
750 PRINT"■":PRINT@ 19,1;N$(1)
752 PRINT@ 27,1;N$(2)
754 PRINT@ 36,1;N$(3)
756 FOR T= 1 TO 3
758 S1=4
760 FOR T1=O1 TO O2
762 W$="":Q$=""
764 GOSUB   618
766 IF N$(T)="A" THEN W$= A$(T1)
768 IF N$(T)="B" THEN W$= B$(T1)
770 IF N$(T)="C" THEN W$= C$(T1)
772 IF N$(T)="D" THEN W$= D$(T1)
774 IF N$(T)="E" THEN W$= E$(T1)
776 IF N$(T)="F" THEN W$= F$(T1)
778 IF N$(T)="G" THEN W$= G$(T1)
780 IF N$(T)="H" THEN W$= H$(T1)
782 IF N$(T)="I" THEN W$= I$(T1)
784 IF N$(T)="J" THEN W$= J$(T1)
786 IF N$(T)="K" THEN W$= K$(T1)
788 IF N$(T)="L" THEN W$= L$(T1)
790 Q$= T$(T1)
792 IF Q$="" THEN Q$="----------"
794 IF W$="" THEN W$="------"
796 PRINT @ 8*(T+1)+(6-LEN(W$))/2*(T1-O1)+4;W$
798 PRINT @ 8*(T+1)+(6-LEN(W$))/2*(T1-O1)+4;W$
800 PRINT @ 0,S1;T1
802 PRINT @ 4,S1;Q$:S1=S1+2
804 GOSUB   618
```

```
806 NEXT T1
808 NEXT T
810 REM ******************************
812 REM ****** CHANGE DATA ********
814 REM ******************************
816 PRINT@ 0,19;"──────────────────────────────────────────────────────────────────────"
818 PRINT@0 ,24;"______________________________________________________________________";
820 PRINT @ 4,21;"DO YOU WISH TO CHANGE AN ENTRY ?"
822 FOR T=1 TO 1500:NEXT T
824 PRINT @ 4,21;"TYPE IN REFERENCE ---- THEN NEW DATA"
826 PRINT@ 13,21;""
828 INPUT "                ";R$
830 IF R$="PRINT" THEN GOTO 872
832 IF R$="GRAPH" THEN GOTO 1072
834 IF R$="NEW" THEN GOTO 706
836 IF R$="CALC" THEN GOTO 606
838 IF R$="STORE" THEN GOTO 1040
840 PRINT@14,21;""
842 INPUT "                  ";BB$
844 IF LEFT$(R$,1)="A" THEN A$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
846 IF LEFT$(R$,1)="B" THEN B$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
848 IF LEFT$(R$,1)="C" THEN C$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
850 IF LEFT$(R$,1)="D" THEN D$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
852 IF LEFT$(R$,1)="E" THEN E$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
854 IF LEFT$(R$,1)="F" THEN F$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
856 IF LEFT$(R$,1)="G" THEN G$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
858 IF LEFT$(R$,1)="H" THEN H$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
860 IF LEFT$(R$,1)="I" THEN I$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
862 IF LEFT$(R$,1)="J" THEN J$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
864 IF LEFT$(R$,1)="K" THEN K$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
866 IF LEFT$(R$,1)="L" THEN L$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
868 IF LEFT$(R$,1)="T" THEN T$(VAL(RIGHT$(R$,(LEN(R$)-1))))=BB$
870 GOTO 744
872 REM
874 REM ******************************
876 REM ********* PRINT OUT *********
878 REM ******************************
880 REM
882 PRINT/P "■";"CALCMATE PROGRAM ";
884 PRINT/P"■";"BY  M.G.KEEN"
886 PRINT/P DT$
888 PRINT/P:PRINT/P:PRINT/P"■";TL$
890 PRINT/P"■":PRINT/P:PRINT/P
892 GOSUB 1136
894 REM *** LEAVE OUT PRINT COMMANDS FOR UNUSED COLUMNES ***
896 PRINT/P  X1$
898 PRINT/P  X2$
900 PRINT/P  X3$
902 PRINT/P  X4$
904 PRINT/P  X5$
906 PRINT/P  X6$
908 PRINT/P  X7$
910 PRINT/P  X8$
912 PRINT/P  X9$
914 PRINT/P  X10$
916 PRINT/P  X11$
918 PRINT/P  X12$
920 PRINT/P  X13$
922 PRINT/P  X14$
924 PRINT/P"──────────────────────────────────────────────────────────────────────"
926 PRINT/P"──────────────────────"
928 PRINT/P
```

```
930 PRINT"██████ORDER OF COLUMNS TO BE PRINTED  "
932 PRINT"████████UP TO SIX   (6)"
934 REM ******************************
936 REM ***** COLUMN SELECT B *******
938 REM ******************************
940 M$(1)="T"
942 INPUT "     1         ";M$(2)
944 INPUT "     2         ";M$(3)
946 INPUT "     3         ";M$(4)
948 INPUT "     4         ";M$(5)
950 INPUT "     5         ";M$(6)
952 INPUT "     6         ";M$(7)
954 REM
956 REM ******************************
958 REM ****** LINE NO SELECT *******
960 REM ******************************
962 REM
964 PRINT"█████NUMBER OF THE LINES TO BE PRINTED  "
966 INPUT "████████FIRST LINE        ";O3
968 INPUT "████████LAST  LINE        ";O4
970 REM ***********************
972 REM ****** PRINT OUT *******
974 REM ***********************
976 PRINT/P TAB(24);M$(2);TAB(34);M$(3);TAB(44);M$(4);TAB(54);M$(5);
978 PRINT/P TAB(64);M$(6);TAB(74);M$(7)
980 PRINT/P
982 FOR T1=O3 TO O4
984 ZZ=2
986 FOR T=1 TO 7
988 GOSUB 618
990 IF M$(T)="T" THEN W$= T$(T1)
992 IF M$(T)="A" THEN W$= A$(T1)
994 IF M$(T)="B" THEN W$= B$(T1)
996 IF M$(T)="C" THEN W$= C$(T1)
998 IF M$(T)="D" THEN W$= D$(T1)
1000 IF M$(T)="E" THEN W$= E$(T1)
1002 IF M$(T)="F" THEN W$= F$(T1)
1004 IF M$(T)="G" THEN W$= G$(T1)
1006 IF M$(T)="H" THEN W$= H$(T1)
1008 IF M$(T)="I" THEN W$= I$(T1)
1010 IF M$(T)="J" THEN W$= J$(T1)
1012 IF M$(T)="K" THEN W$= K$(T1)
1014 IF W$="" THEN W$="------"
1016 IF T$(T1)="-----------" THEN W$=T$(T1)
1018 PRINT/P  TAB(ZZ);W$;
1020 IF W$=T$(T1) THEN ZZ=ZZ+8
1022 ZZ=ZZ+10
1024 NEXT T
1026 PRINT/P :ZZ=0
1028 NEXT T1
1030 PRINT/P
1032 PRINT/P"----------------------------------------------------------------------------"
1034 PRINT/P"--------------------"
1036 GOTO 1072
1038 REM
1040 REM ***********************
1042 REM ****** STORE DATA *******
1044 REM ***********************
1046 REM
```

```
1048 PRINT"■■■■■■■■■■SET TAPE READY FOR DATA STORE"
1050 PRINT"■■■■■■■■■■WHEN READY PRESS ANY KEY  "
1052 GET JJ$
1054 IF JJ$="" THEN GOTO  1052
1056 PRINT:PRINT:PRINT
1058 WOPEN "  DATA STORE"
1060 FOR T1=1 TO 25
1062 PRINT#T A$(T1),B$(T1),C$(T1),D$(T1),E$(T1),F$(T1),G$(T1)
1064 PRINT#T H$(T1),I$(T1),J$(T1),K$(T1),L$(T1),T$(T1)
1066 NEXT T1
1068 CLOSE
1070 GOTO 718
1072 REM ********************************
1074 REM     SET UP PLOT VARIABLES
1076 REM ********************************
1078 INPUT"■■■INPUT COLUMN TO BE PLOTTED       ? ";R$
1080 INPUT"■■FROM LINE NUMBER                  ? ";Q1
1082 INPUT"■■TO LINE NUMBER                    ? ";Q2
1084 N=(Q2-Q1)+1
1086 DIM A(N):P=0:PH=1
1088 PRINT
1090 PRINT"BAR CHART          B"
1092 PRINT"■or"
1094 PRINT"■GRAPH              G"
1096 INPUT"■                                ? ";Z$
1098 IF Z$="G" THEN   INPUT"■■HORIZONTAL STEP VALUE          ? ";Q
1100 INPUT"■■MAXIMUM VALUE VERTICAL SCALE   ? ";P
1102 FOR X=1 TO N
1104 IF R$="A" THEN A(X)=VAL(A$(X))
1106 IF R$="B" THEN A(X)=VAL(B$(X))
1108 IF R$="C" THEN A(X)=VAL(C$(X))
1110 IF R$="D" THEN A(X)=VAL(D$(X))
1112 IF R$="E" THEN A(X)=VAL(E$(X))
1114 IF R$="F" THEN A(X)=VAL(F$(X))
1116 IF R$="G" THEN A(X)=VAL(G$(X))
1118 IF R$="H" THEN A(X)=VAL(H$(X))
1120 IF R$="I" THEN A(X)=VAL(I$(X))
1122 IF R$="J" THEN A(X)=VAL(J$(X))
1124 IF R$="K" THEN A(X)=VAL(K$(X))
1126 IF A(X)>P THEN P=A(X)
1128 NEXT X
1130 GOTO 186
1132 REM ********************************************************************
1134 REM *** INSERT YOUR OWN COLUMN    TITLES THESE ARE EXAMPLES ONLY **
1136 REM ********************************************************************
1138 X1$=   "      A        LABOUR BUDGET                          £        "
1140 X2$=   "      B        LABOUR ACTUAL                          £        "
1142 X3$=   "      C        PURCHASES BUDGET                                "
1144 X4$=   "      D        PURCHASES ACTUAL                       £        "
1146 X5$=   "      E        TOTAL BUDGET                           £        "
1148 X6$=   "      F        TOTAL SPENT                            £        "
1150 X7$=   "      G        VARIANCE                               £        "
1152 X8$=   "      H        PERCENT OF BUDGET SPENT                      "
1154 X9$=   "      I  £     BUDGET FOR REST OF YEAR                      "
1156 X10$=   "      J  £     BUDGET FOR NEXT PERIOD                       "
1158 X11$=   "      K        PERIOD NUMBER                                "
1160 X12$=   "      L  £     1982 TOTAL BUDGET                            "
1162 X13$=   "      M  £     1982 ACTUAL SPENT                            "
1164 X14$=   "      N  £     1982 ACTUAL SPENT AVERAGE PER PERIOD         "
1166 RETURN
```

Dear Sir,

Following a small article in P.C.W. recently regarding PEEKing and POKEing the MZ-80K, I typed in the small sample program that was given as a possible screen dump to printer, and found that all I got was a column of characters down the left-hand margin of the printer. Investigation of this program showed a small fault which was quite easy to eliminate, and with a little extra work the screen dump program that I have listed, comprising only two lines was produced.

This runs as fast as the Sharp printer can print it, unlike others that take a second or two to start up. It will only dump characters that have an ASCII code, and not the spcial Sharp graphics, but nevertheless it is most useful. It may of course be tailored to dump only part of the screen by variation of the values X and Y.

The above work made me realise that there was a "look-up" table embedded in the Monitor of the MZ-80K that could be used to convert Display Code to ASCII, and from this I realised that as the Monitor transfers keyboard input to screen, there must also be another table that converts ASCII to display Code.

By writing a short program to PEEK the contents of the Monitor ROM and look for a series of contents that showed a sequence of numbers in ASCII or Display Code order, I was able to discover the location of these tables.

The ASCII value of Display Codes starts at location 3270, and by PEEKing the contents of (3270+Display Code value), the ASCII value for any Display code between 0 and 223, with the exception of 193 to 198 is returned. The Display Code value of an ASCII code starts at location 3046, and by PEEKing the contents of (3014+ASCII value), the Display Code for any ASCII value between 32 and 255 is returned. The difference between 3014 and 3046 is because the lowest ASCII code is 32, whereas the Display Codes start at 0.

This gives a very much faster way of printing a single character string variable, which works in a fraction of the time taken by a CURSOR statement followed by a PRINT statement, and so this gave me an opportunity of further speeding up the operation of my disc version of WP2 wordprocessor. At the same time I removed a few bits of redundant code, tidied up a few REM statements, and incorporated the other modifications that appeared in SUN 8, with the result that the input routine now runs at about 10 cps and that it does not miss a single key stroke, no matter how fast you try to type. A list of the revised input routine is enclosed, and I am prepared to print copies of the whole program for #1.00 per copy, or to SAVE onto a subscribers disc for the same fee.

I am still looking for a way of saving a modified Disc BASIC, as can be done with USR(36):USR(33) in Tape BASIC, so I would welcome any available assistance.

Trusting that you will find these notes as useful as I have found others in SUN.

M. BELLAMY
SUSSEX

```
10 REM.....................DISCWORDPRO.................................
11 REM WRITTEN WITH SP 6015 AND KNIGHT'S DISC COMMANDER...................
580 REM.....................................................INPUT ROUTINE
582 A$="":AA$="":D=12+(S-5*INT(S/5))*2:CURSOR 0,D:PRINT L;"?";SPC(/5)
583 DD=40*D+53250:C=1:S=S+1:FOR A=1 TO 76
585 POKE 57347,(5-C):POKE 4464,C
586 GET AA$:IF AA$="" THEN 586
587 A2=ASC(AA$):IF((A2<96)+(A2>102))*((A2<17)+(A2>20))THEN 596
588 IF A2=99 THEN C=0:GOTO 585
589 IF A2=98 THEN C=1:GOTO 585
590 IF A2=102 THEN A=76:GOTO 600
591 IF A2=19 THEN AA$=" ":A2=32:GOTO 596
592 IF((A2=18)*(A>40))THEN GOSUB 601
593 IF((A2=20)+(A2=96))*(A>1)THEN A$=LEFT$(A$,LEN(A$)-1):A=A-1:GOTO 595
594 IF((A2=17)*(A<CL-40))THEN A$=A$+SPC(40):A=A+40
595 POKE(DD+A),73:POKE(DD+A+1),0:GOTO586
596 IF(LEN(A$)>CL-9)*(AA$=" ")THEN A=76:GOTO 600
597 A$=A$+AA$:POKE(DD+A),(PEEK(3014+A2)):POKE(DD+A+1),73
599 IF((LEFT$(A$,1)="")*(LEN(A$)=2))THEN A=76
600 POKE $208B,0:NEXT A:POKE $208B,$BE:RETURN
601 A=A-40:A$=LEFT$(A$,A):FOR P=(DD+A+40)TO(DD+A)STEP-1:POKE P,0:NEXT:RETURN
3000 REM ..................................................... JTILITY - SCREEN PRINT
3001 FOR Y=0 TO 24:FOR X=0 TO 39:T=53248+40*Y+X
3002 A$=A$+CHR$(PEEK(3270+PEEK(T))):NEXT:PRINT/P A$:A$="":NEXT:END
```

DEFENCE and SKETCH PAD
by Mr. C. Headington

SKETCH PAD (c)

1.DRAW

The 'cursor' is moved using the cursor control keys,most of the characters on the key board can be displayed,when the display is complete press CR.

2.DISPLAY

This will cause the program to display the picture in memory,when it has finished press CR to return to the main menu.

3.ERASE

The drawing in memory will be erased.

4.WRITE

The drawing is written to cassette,you will be prompted to give a name less than 17 chars,upon default the data will be named '*'.

5.READ

The drawing data is read into the computer,wind the cassette to the correct position and enter the name as prompted.After a delay you will return to the main menu and you can display the drawing with option 2.

6.HCOPY

The drawing will be displayed and a hardcopy will be made,you will be prompted to check that the printer is on line beforehand.

7.DATA WRITE

Data will be printed approx. 6 lines at a time on the prompt 'CR' NOW keep pressing the CR key until you have CR'ed 'RUN 1840' and another 6 lines will be written,when all the DATA has been written the lines 5000-5030 will appear,when these are CR'ed you will have the data to draw the picture in lines 5000 onwards,if you wish to use these lines delete the rest of the program and the data may now be saved on tape.

```
10 PRINT"■          *<*****************"
20 PRINT"           *                  *"
30 PRINT"           *   SKETCH PAD     *"
40 PRINT"           *                  *"
50 PRINT"           ******************"
60 PRINT
70 PRINT
80 PRINT
90 PRINT"      DRAW.....................1"
100 PRINT
110 PRINT"     DISPLAY..................2"
120 PRINT
130 PRINT"     ERASE....................3"
140 PRINT
150 PRINT"     WRITE....................4"
160 PRINT
170 PRINT"     READ.....................5"
180 PRINT
190 PRINT"     H/COPY...................6"
200 PRINT
210 PRINT"     DATA WRITE...............7"
220 INPUT"■      INPUT CHOICE.............";I
230 DIM A(255):DIM B(255)
240 DIM C(255):DIM D(255)
250 ON I GOSUB 620,450,1850,820,1110,1900,1390
260 GOTO 10
270 REM STORE
280 DIM A(255)
290 DIM B(255)
300 DIM C(255)
310 DIM D(255)
320 FOR L1=0 TO 250
330 A(L1)=PEEK(L1+53248)
340 NEXT L1
350 FOR L2=251 TO 501
360 B(L2-249)=PEEK(L2+53248)
370 NEXT L2
380 FOR L3=502 TO 752
390 C(L3-499)=PEEK(L3+53248)
400 NEXT L3
410 FOR L4=753 TO 1000
420 D(L4-749)=PEEK(L4+53248)
430 NEXT L4
440 RETURN
450 REM PRINT TO SCREEN
460 PRINT"■■"
470 FOR L5=0 TO 250
480 POKE 53248+L5,A(L5)
490 NEXT L5
500 FOR L6=251 TO 501
510 POKE 53248+L6,B(L6-249)
520 NEXT L6
530 FOR L7=502 TO 752
540 POKE 53248+L7,C(L7-499)
550 NEXT L7
560 FOR L8=753 TO 1000
570 POKE 53248+L8,D(L8-749)
580 NEXT L8
590 IF HCOPY=1 THEN RETURN
600 GET A$:IF A$="*" THEN RETURN
610 GOTO 600
620 REM SKETCH
630 PRINT"■■"
640 LET X=0:LET Y=2
```

```
650 CURSOR X,Y
660 PRINT" "
670 GET C$:IF C$="■" THEN LET X=X+1
680 IF C$="■" THEN LET X=X-1
690 IF C$="■" THEN LET Y=Y-1
700 IF C$="■" THEN LET Y=Y+1
710 IF X>38 THEN LET X=X-1
720 IF X<1 THEN LET X=X+1
730 IF Y>23 THEN LET Y=Y-1
740 IF Y<1 THEN LET Y=Y+1
750 CURSOR X,Y-1
760 GETA$
770 IF A$="¥" THEN GOTO 270
780 PRINT A$
790 CURSORX,Y
800 PRINT "‾"
810 GOTO 650
820 REM SAVE
830 PRINT"■■"
840 PRINT"                    WRITE TO CASSETTE"
850 PRINT"                    *****************"
860 PRINT
870 PRINT
880 PRINT"                      INPUT NAME"
890 PRINT"                      **********"
900 PRINT
910 PRINT"                      16 CHAR'S"
920 PRINT"                   ▓▓▓▓**********▓▓▓▓":PRINT"
930 PRINT"                   ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
940 INPUT"■■                    ";N$
950 PRINT"■■"
960 WOPEN N$
970 FOR C1=0 TO 250
980 PRINT/T A(C1)
990 NEXT C1
1000 FOR C2=251 TO 501
1010 PRINT/T B(C2-249)
1020 NEXT C2
1030 FOR C3=502 TO 752
1040 PRINT/T C(C3-499)
1050 NEXT C3
1060 FOR C4=753 TO 1000
1070 PRINT/T D(C4-749)
1080 NEXT C4
1090 CLOSE
1100 RETURN
1110 PRINT"■■"
1120 PRINT"                    CASSETTE READ  "
1130 PRINT"                    *************"
1140 PRINT
1150 PRINT
1160 PRINT"                      INPUT NAME"
1170 PRINT"                      **********"
1180 PRINT
1190 PRINT"                      16 CHAR'S"
1200 PRINT"                   ▓▓▓▓**********▓▓▓▓":PRINT"
1210 PRINT"                   ▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
1220 INPUT"■■                    ";N$
1230 PRINT"■■"
1240 ROPEN N$
```

```
1250 FOR C1=0 TO 250
1260 INPUT#7 A(C1)
1270 NEXT C1
1280 FOR C2=251 TO 501
1290 INPUT#7 B(C2-249)
1300 NEXT C2
1310 FOR C3=502 TO 752
1320 INPUT#7 C(C3-499)
1330 NEXT C3
1340 FOR C4=753 TO 1000
1350 INPUT#7 D(C4-749)
1360 NEXT C4
1370 CLOSE
1380 RETURN
1390 REM PROG
1400 J=999
1410 U=5040
1420 PRINT"■"
1430 PRINT U;" DATA";
1440 FOR ZZ=0 TO 250
1450 IF A(ZZ)<1 THEN J=J-1
1460 IF A(ZZ)<1 THEN GOTO 1500
1470 PRINT ZZ;",";A(ZZ);",";
1480 XX=XX+1
1490 IF XX>4 THEN GOSUB 1770
1500 NEXT ZZ
1510 FOR ZZ=251 TO 501
1520 IF B(ZZ-249)<1 THEN J=J-1
1530 IF B(ZZ-249)<1 THEN GOTO 1570
1540 PRINT ZZ;",";B(ZZ-249);",";
1550 XX=XX+1
1560 IF XX>4 THEN GOSUB 1770
1570 NEXT ZZ
1580 FOR ZZ=502 TO 752
1590 IF C(ZZ-499)<1 THEN J=J-1
1600 IF C(ZZ-499)<1 THEN GOTO 1640
1610 PRINT ZZ;",";C(ZZ-499);",";
1620 XX=XX+1
1630 IF XX>4 THEN GOSUB 1770
1640 NEXT ZZ
1650 FOR ZZ=753 TO 1000
1660 IF D(ZZ-749)<1 THEN J=J-1
1670 IF D(ZZ-749)<1 THEN GOTO 1710
1680 IF XX>4 THEN GOSUB 1770
1690 PRINT ZZ;",";D(ZZ-749);",";
1700 XX=XX+1
1710 NEXT ZZ
1720 PRINT:PRINT"5000 FOR T=1 TO ";J+2
1730 PRINT"5010 READ A:READ B"
1740 PRINT"5020 POKE 53248+A,B"
1750 PRINT"5030 NEXT T"
1760 PRINT"RUN 10":END
1770 IF H>6 THEN GOSUB 1810
1780 PRINT:U=U+10:PRINT U;" DATA";
1790 H=H+1:XX=0
1800 RETURN
1810 PRINT"■RUN 1840"
1820 PRINT "■                    'CR' NOW"
1830 PRINT"■":END
1840 PRINT"■":H=0:RETURN
1850 REM ERASE
1860 CLR
```

```
1870 DIM A(255):DIM B(255)
1880 DIM C(255):DIM D(255)
1890 GOTO 10
1900 REM HARD COPY
1910 PRINT"                 "
1920 PRINT"          VERY IMPORTANT "
1930 PRINT"                 "
1940 PRINT
1950 PRINT
1960 PRINT
1970 PRINT
1980 PRINT"                 "
1990 PRINT"     CHECK PRINTER ON/OFF SWITCH "
2000 PRINT"                 "
2010 PRINT"   IS ON AND I/O BOX IS ON LINE. "
2020 PRINT"                 "
2030 PRINT"                    PRESS ANY NOW"
2040 PRINT"                 "
2050 GET A$:IF A$="" THEN GOTO 2050
2060 HCOPY=1
2070 GOSUB 450
2080 PRINT/P"  "
2090 COPY/P 1
2100 HCOPY=0
2110 RETURN
```

```
10 PRINT"         "
20 PRINT"        "
30 PRINT"        "
40 PRINT"        "
50 PRINT"        "
60 FOR T=1 TO 5000:NEXT T
70 PRINT"  DEFEND YOUR BASE FROM THE ALIEN ATTACK"
80 PRINT" USE Q AND Z TO MOVE YOUR CRAFT AND L TO"
90 PRINT" FIRE YOU HAVE 3 LIVES.A LIFE IS LOST IF"
100 PRINT" AN ALIEN RAMS YOU OR YOU MISS ONE "
110 FOR T=1 TO 9000:NEXT T
120 PRINT"  ":Q=53248:FOR T=39 TO 999 STEP 40
130 POKE 53248+T,113
140 NEXT T
150 POKEQ+960,118:POKEQ+921,118:POKEQ+882,118:POKEQ+843,118:POKEQ+844,119
160 POKEQ+885,119:POKEQ+926,119:POKEQ+927,60:POKEQ+928,60:POKEQ+929,60
170 POKEQ+930,60:POKEQ+931,118:POKEQ+892,118:POKEQ+893,119:POKEQ+934,119
180 POKEQ+935,60:POKEQ+936,60:POKEQ+937,60:POKEQ+938,118:POKEQ+899,118
190 POKEQ+860,118:POKEQ+861,119:POKEQ+902,119:POKEQ+943,119:POKEQ+944,60
200 POKEQ+945,60:POKEQ+946,60:POKEQ+947,60:POKEQ+948,60:POKEQ+949,60
210 POKEQ+950,60:POKEQ+951,60:POKEQ+952,118:POKEQ+913,118:POKEQ+874,118
220 POKEQ+835,118:POKEQ+836,119:POKEQ+877,119:POKEQ+918,119
230 C=53248+321
240 A=53248+77:LI=3
250 B=A
260 POKE A,81:POKE A-1,52:POKE A-2,71:POKE A-3,81
270 POKE A-40,0:POKE A-2-40,0:POKE A-1-40,0:POKE A-3-40,0
280 POKE A+40,0:POKE A-1+40,0:POKE A-2+40,0:POKE A-3+40,0
290 GET A$
```

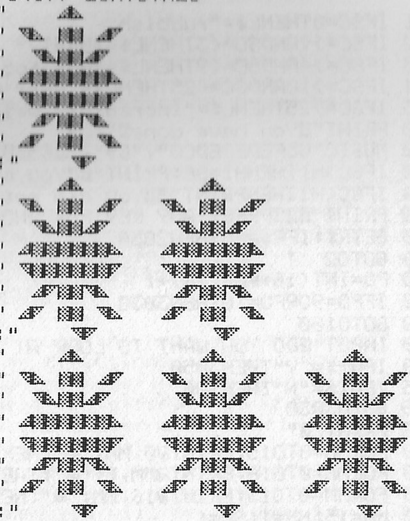
```
300 IF A$="Q" THEN A=A-40
310 IF A$="Z" THEN A=A+40
320 IF A<53248+0 THEN A=B
330 IF A>53248+759THEN A=B
340 IF A$="L" THEN GOSUB 570
350 POKE C,0:POKE C,200
360 IF LI<0 THEN 650
370 D=C
380 IF PEEK(C+1)>0THEN GOSUB720
390 Z=INT(3*RND(1)+1)
400 POKE C-1,0:POKE C-41,0:POKEC+39,0
410 IF Z=1 THEN C=C-39
420 IF Z=2 THEN C=C+41
430 IF Z=3 THEN C=C+1
440 IF C<53248+40THEN C=C+40
450 IF C>53248+719 THEN C=C-39
460 POKE C,0
470 GOTO 250
480 LI=LI-1:REM DEAD
490 POKE C-1,0:POKE C+39,0:POKE C-41,0
500 SC=SC+5
510 MUSIC"A"
520 POKE C,0:POKEZZ+1,0:POKEZZ+2,0
530 POKE ZZ,0:POKE ZZ-1,0:POKEZZ-2,0
540 X=INT((15*RND(1))+1)
550 C=53248+X*40+1
560 RETURN
570 FOR ZZ=A-5 TO A-30 STEP-1
580 POKE ZZ,81
590 IF PEEK(ZZ-1)=200 THEN GOTO 490
600 POKE ZZ+1,0
610 NEXT ZZ
620 POKE ZZ+1,0
630 MUSIC"A0A0"
640 RETURN
650 PRINT"■■■■"
660 PRINT"        YOU HAVE SCORED";SC;" POINTS"
670 FOR T=1 TO 5000:NEXTT
680 PRINT"■■■■■■■■■■■■                ANOTHER GO?"
690 GET A$:IF A$="Y" THEN RUN
700 IF A$="N" THEN END
710 GOTO 690
720 LI=LI-1:REM DEAD
730 POKE C-1,0:POKE C+39,0:POKE C-41,0
740 POKE C,0
750 MUSIC"A"
760 X=INT((15*RND(1))+1)
770 C=53248+X*40+1
780 RETURN
```

CRYSTAL ADVENTURE
and
HORSE RACING

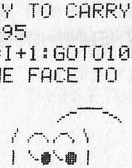
```
1 TEMPO7:HI=0
2 GOSUB9000:DIMB(255):SC=0:PRINT"■Please wait for 45 seconds for me to "
3 PRINT"■sort out all the rooms."
5 DIMC(225)
10 DIMA$(225)
15 DIMB$(225)
20 FORI=1TO225
30 V=INT(10*RND(1))+1
40 IFV=<6THENA$(I)=".":B$(I)="THIS ROOM IS EMPTY":C(I)=0
50 IFV=7THENA$(I)=CHR$(104):B$(I)="THIS IS A ROOM WITH A MONSTER":C(I)=0
60 IFV=8ORV=9THENA$(I)="▙":B$(I)="THIS IS A ROOM WITH A TRANSPORTER":C(I)=0
70 IFV=10THENA$(I)="*":B$(I)="CRYSTALS ARE HERE":C(I)=INT(3*RND(1))+1
80 NEXTI
90 A=0:I=0
100 PRINT"■"
110 FORN=0TO38:PRINT@0,N;"▒":NEXT
115 IFW<0THENW=0
120 PRINT"▒        ROOM";A;"      CRYSTALS";SC
125 PRINT@1,38;"▒"
130 FORN=0TO38:PRINT@2,N;"▒":NEXT
140 FORN=1TO10:PRINT"▒";TAB(38);"▒":NEXT
150 FORN=0TO38:PRINT@12,N;"▒":NEXT
160 IFA=0THENPRINT@4,2;"THIS IS THE ROOM STARTER"
170 IFA<>1THENPRINT@4,2:B$(I)
175 IFA$(I)="*"THENFORU=1TO3000:NEXT:GOTO1000
176 IFA$(I)="▙"THENFORU=1TO3000:NEXT:GOTO900
177 IFA$(I)=CHR$(104)THENFORU=1TO3000:NEXT:GOTO1100
180 N=A+15:S=A-15:W=A-1:E=A+1
185 IFN>225THENPRINT@6,2;"N=":GOTO195
190 PRINT@6,2;"N=";A$(N)
195 IFS<0THENPRINT@7,2;"S=":GOTO205
200 PRINT@7,2;"S=";A$(S)
205 IFE>225THENPRINT@8,2;"E=":GOTO215
210 PRINT@8,2;"E=";A$(E)
215 IFW<0THENPRINT@9,2;"W=":GOTO230
220 PRINT@9,2;"W=";A$(W)
230 PRINT@14,0;".=EMPTY ROOM
240 PRINT@15,0;"▙=TRANSPORTER
250 PRINT@16,0;CHR$(104);"=MONSTER"
260 PRINT@17,0;"*=CRYSTAL"
261 PRINT@19,0;"Hi Score=";HI;"■crystals."
265 PRINT@10,0;"                                                    "
270 INPUT"■■■■■■■■■■■■■■■■■■■■INPUT DIRECTION (N/S/E/W) ?";P$
280 IFP$="N"THEN400
290 IFP$="S"THEN500
300 IFP$="E"THEN600
310 IFP$="W"THEN700
320 IFP$="Q"THEN2000
330 GOTO270
```

```
400 A=A+15:I=I+15
405 IFA>225THENA=A-15:I=I-15:GOTO270
410 IFA$(I)="◣"THEN900
420 IFA$(I)="*"THEN1000:REM*CRYSTALS*
430 IFA$(I)=CHR$(104)THEN1100:REM*MON*
440 IFA$(I)="."THEN3000
450 GOTO270
500 A=A-15:I=I-15
505 IFA<0THENA=A+15:I=I+15:GOTO270
510 IFA$(I)="◣"THEN900
520 IFA$(I)="*"THEN1000:REM*CRYSTALS*
530 IFA$(I)=CHR$(104)THEN1100:REM*MON*
540 IFA$(I)="."THEN3000
550 GOTO270
600 A=A+1:I=I+1
605 IFA>225THENA=A-1:I=I-1:GOTO270
610 IFA$(I)="◣"THEN900
620 IFA$(I)="*"THEN1000:REM*CRYSTALS*
630 IFA$(I)=CHR$(104)THEN1100:REM*MON*
640 IFA$(I)="."THEN3000
650 GOTO270
700 A=A-1:I=I-1
705 IFA<0THENA=A+1:I=I+1:GOTO270
710 IFA$(I)="◣"THEN900
720 IFA$(I)="*"THEN1000:REM*CRYSTALS*
730 IFA$(I)=CHR$(104)THEN1100:REM*MON*
740 IFA$(I)="."THEN3000
750 GOTO270
800 PRINT"■YOU HAVE BEEN DEMOLISHED BY THE MONSTER"
810 PRINT"■AND HAVE NO CRYSTALS LEFT"
820 PRINT"■REPLAY Y/N"
830 GETR$:IFR$="Y"THEN2
840 IFR$="N"THENEND
850 GOTO830
900 PRINT"■";@3,0;"          ◣
901 PRINT    @4,0;"          ◣■
902 PRINT    @5,0;"          ■■◣
903 PRINT    @6,0;"        /■■■◣
904 PRINT    @7,0;"        \■■■■◣
905 PRINT@10,0;"THIS IS A"
906 FORO=1TO5:FOROO=1TO50
907 PRINT@10,10;"TRANSPORTER"
908 NEXTOO
909 FOROO=1TO50
910 PRINT@10,10;"              "
911 NEXTOO
912 NEXTO
913 PRINT@10,10;"TRANSPORTER":OO=INT(225*RND(1))+1
914 PRINT"■■YOU ARE NOW MOVED TO SQARE";OO:FORU=1TO1000:NEXT
915 A=OO:I=OO
916 GOTO100
1000 PRINT"■"
1002 IFA=B(A)THEN1006
1004 B(A)=A
1005 GOTO1050
1006 PRINT"■I AM SORRY BUT YOU HAVE BEEN HERE      "
1008 PRINT"■BEFORE AND SO CANNOT RECEIVE ANY       "
1010 PRINT"■CRYSTALS AGAIN"
1020 MUSIC"C9E3GEGB7"
1030 PRINT"■PRESS ANY KEY"
```

```
1035 A$(A)=".":B$(A)="THIS ROOM IS EMPTY"
1040 GETR$:IFR$=""THEN1040
1045 A=A+1:I=I+1:GOTO100
1050 PRINT"■YOU HAVE FOUND";C(I);"■CRYSTALS"
1060 IFC(I)=1THENPRINT@3,0;"
1061 IFC(I)=1THENPRINT@4,0;"
1062 IFC(I)=1THENPRINT@5,0;"
1064 IFC(I)=1THENPRINT@6,0;"
1065 IFC(I)=1THENPRINT@7,0;"
1066 IFC(I)=1THENPRINT@8,0;"
1067 IFC(I)=1THENPRINT@9,0;"
1068 IFC(I)=1THENPRINT@10,0;"
1070 IFC(I)=2THENPRINT@3,0;"
1071 IFC(I)=2THENPRINT@4,0;"
1072 IFC(I)=2THENPRINT@5,0;"
1073 IFC(I)=2THENPRINT@6,0;"
1074 IFC(I)=2THENPRINT@7,0;"
1075 IFC(I)=2THENPRINT@8,0;"
1076 IFC(I)=2THENPRINT@9,0;"
1077 IFC(I)=2THENPRINT@10,0;"
1078 IFC(I)=3THENPRINT@3,0;"
1079 IFC(I)=3THENPRINT@4,0;"
1080 IFC(I)=3THENPRINT@5,0;"
1081 IFC(I)=3THENPRINT@6,0;"
1082 IFC(I)=3THENPRINT@7,0;"
1083 IFC(I)=3THENPRINT@8,0;"
1084 IFC(I)=3THENPRINT@9,0;"
1085 IFC(I)=3THENPRINT@10,0;"
```



```
1090 PRINT"■PRESS ANY KEY TO CARRY ON"
1095 GETR$:IFR$=""THEN1095
1098 SC=SC+C(I):A=A+1:I=I+1:GOTO100
1100 PRINT"■YOU HAVE COME FACE TO FACE WITH A  "
1101 PRINT"■MONSTER!!!!!
1102 PRINT"■■■
1103 PRINT"
1104 PRINT"
1105 PRINT"
1106 PRINT"
1107 PRINT"
1108 PRINT"
```



```
1109 PRINT"■■HE IS DECIDING WHETHER TO EAT YOU OR "
1110 PRINT"■NOT."
1120 EU=INT(50*RND(1))+1
1130 FORWE=1TO5000
1140 NEXT
1150 IFEU=50THEN1300
1160 PRINT"■■■HE DOES NOT WANT TO EAT YOU BUT HE   "
1170 PRINT"■WANTS 10 CRYSTALS OFF YOU"
1180 IFSC=>10THENSC=SC-10:GOTO1200
1190 PRINT"■BUT YOU HAVEN'T GOT 10 CRYTALS.BYE BYE!!":FORWE=1TO5000:NEXT
1195 GOTO800
1200 PRINT"■YOU HAVE 10 CRYSTALS"
1210 PRINT"■PRESS ANY KEY"
1220 GETR$:IFR$=""THEN1220
1230 A=A+1:I=I+1:GOTO100
1300 PRINT"■■■HE HAS DECIDED TO EAT YOU..."
1310 FORWE=1TO1000:NEXT
1320 PRINT"■■■HE HAS DECIDED TO EAT YOU...GULP"
1330 FORWE=1TO1000:NEXT:GOTO800
2000 PRINT"■You have escaped from the monster maze"
2010 PRINT"■and have";SC;"■crystals with you."
```

```
2011 IFSC=0THENL$="rubbish"
2012 IFSC=>1ANDSC=<3THENL$="badly"
2013 IFSC=>4ANDSC=<9THENL$="not too badly"
2014 IFSC=>10ANDSC=<25THENL$="very well"
2015 IFSC=>25THENL$="incredibly well"
2020 PRINT"■You have done■";L$
2030 MUSIC"G5FEDEDEDC0","C9","_C5_D_E_F_G_A_B9_#A9_#A9
2032 IFSC>HITHENHI=SC:PRINT"■■You also have the high score."
2034 IFSC<HITHENPRINT"■■You did not get the high score."
2040 PRINT"■■■PRESS ANY KEY FOR ANOTHER GO"
2050 GETR$:IFR$=""THEN2050
2060 GOTO2
3000 PO=INT(10*RND(1))+1
3010 IFPO=9ORPO=10THEN3030
3020 GOTO100
3030 INPUT"■DO YOU WANT TO LOOK AT THE MAP ?";UI$
3040 IFUI$="Y"THEN3050
3045 IFUI$="N"THEN100
3049 GOTO3030
3050 PRINT"■"
3060 FORMM=0TO15:PRINT@0,MM;"※":NEXT
3070 FORMM=0TO16:PRINT@MM,0;"※";TAB(16);"※":NEXT
3080 FORMM=0TO15:PRINT@16,MM;"※":NEXT
3090 MM=15:MN=1:SP=1
3095 PRINT@MM,MN;A$(SP)
3100 MN=MN+1:SP=SP+1:IFMN=16THENMM=MM-1
3105 IFSP=255THEN3140
3110 IFMN=16THENMN=1
3120 IFMM=0THEN3140
3130 GOTO3095
3140 PRINT@5,30;"■=YOU"
3150 MM=15:MN=1:SP=0
3160 MN=MN+1:SP=SP+1:IFMN=16THENMM=MM-1
3170 IFSP=ATHEN3400
3180 IFMN=16THENMN=1
3190 IFMM=0THEN3400
3200 GOTO3160
3400 PRINT@MM,MN-1;"■":FORXS=1TO10
3410 PRINT@20,0;"PRESS ANY KEY"
3420 GETR$:IFR$<>""THEN100
3430 NEXT
3440 PRINT@MM,MN-1;A$(I):FORXS=1TO10
3450 GETR$:IFR$<>""THEN100
3460 NEXT
3470 GOTO3400
```

```
9000 PRINT"■          CRYSTAL ADVENTURE"
9010 PRINT"
9020 PRINT"■■You have been flown to the planet ZEON"
9030 PRINT"where there is an underground maze of   "
9040 PRINT"rooms.Each room may have a MONSTER      "
9050 PRINT"TRANSPORTER or a CRYSTAL in it.You must"
9060 PRINT"avoid the monsters at all times unless "
9070 PRINT"you have 10 crystals.The object is to   "
9080 PRINT"■try to collect as many crystals as "
9090 PRINT"possible and then escape by pressing 'Q'"
9095 PRINT"■for Quit."
9100 PRINT"■If you move into a room with a         "
9110 PRINT"transporter it will transport you into"
9120 PRINT"any of the 255 rooms of the maze."
9130 PRINT"This can be dangerous as the outcome    "
9140 PRINT"could be that you are transported to a "
9150 PRINT"room with a monster in."
9160 PRINT"■■PRESS 'C' TO CONTINUE"
9170 GETR$:IFR$="C"THEN9190
9180 GOTO9170
9190 PRINT"■■To move around the maze you are asked"
9200 PRINT"to input the directions of the compass."
9210 PRINT"■Every time you move into an empty room"
9220 PRINT"you will be given a report on what room"
9230 PRINT"you are in,the number of crystals you   "
9240 PRINT"have,what is North of you,what is South"
9250 PRINT"of you,what is East of you and what is "
9260 PRINT"West of you."
9270 PRINT"■Now and then you will be allowed to    "
9280 PRINT"look at the underground map stating     "
9290 PRINT"where you are and where everything else"
9295 PRINT"is."
9300 PRINT"■■PRESS 'S' TO PLAY"
9310 GETR$:IFR$="S"THENRETURN
9320 GOTO9310
```

```
50 REM ########## ROCK QUIZ ############
52 REM COPYRIGHT R.L.BENNETT
60 PRINT"■"
70 PRINT"          ____      "
72 PRINT"         |    |          "
74 PRINT"         |    |           "
76 PRINT"         |    |             "
78 PRINT"       ◜|  ◜|              "
79 PRINT"       ◡|  ◡|            |  "
80 PRINT"                          |  "
81 PRINT"                          |  "
82 PRINT"     ROCK  QUIZ         ◜|  "
83 PRINT"                        ◡|    "
84 PRINT"                           "
85 PRINT"                           "
86 PRINT"                           "
87 PRINT"                                |  "
88 PRINT"       |      BY R.L. BENNETT  |  "
89 PRINT"       |                       |  "
90 PRINT"       |                      ◜|  "
92 PRINT"     ◜|                      ◡|  "
93 PRINT"     ◡|                        "
95 PRINT"                              "
```

Have fun with
ROCK QUIZ
by
R.L. Bennett

```
97 S=0
153 R3$="#G3+C1B4+D1+C4+C1B4#G2G7R4"
154 R2$="+F3+#D1+D4+C2+C5G+C#A"
155 R1$="+C3+D+#D4+#D1+#D4+#D1"
156 FORT=1TO3
157 TEMPO6:MUSICR1$;R2$;R3$
158 NEXTT
160 PRINT""
161 PRINT"WELCOME TO ROCK QUIZ"
162 FORT=1TO2000:NEXT
163 PRINT"________________________"
165 PRINT"ROCK QUIZ IS A QUIZ ON ROCK MUSIC"
170 PRINT"SINCE 1955 TO 1983. "
172 PRINT"FROM LITTLE RICHARD & ELVIS PRESLEY"
173 PRINT"TO BOB DYLAN & ELTON JOHN."
175 PRINT"FROM THE BEATLES & THE ROLLING STONES"
176 PRINT"TO FUN BOY THREE & CULTURE CLUB."
178 PRINT"PRESS [C] TO CONTINUE"
179 GETE$:IFE$=""THEN179
180 IFE$="C"THEN189
182 GOTO179
189 MUSIC"+#C2"
190 PRINT"YOU WILL BE GIVEN 60 QUESTIONS"
192 PRINT"AT THE END OF THE QUIZ YOU WILL BE "
193 PRINT"TOLD HOW WELL YOU DID"
195 PRINT"(OR BADLY YOU DID)."
200 FORT=1TO4000:NEXT
205 INPUT"Name:- ";A$
207 FORT=1TO1000:NEXT
210 PRINT""
212 PRINT"";A$;" WHEN YOU ARE READY"
213 PRINT"PRESS [S] TO START"
215 GETE$:IFE$=""THEN215
216 IFE$="S"THEN250
217 GOTO215
250 PRINT"QUESTION 1"
252 PRINT"WHAT IS LITTLE RICHARD'S SURNAME ?"
254 PRINT"1:-PENNANT"
256 PRINT"2:-PENNIMAN"
257 PRINT"3:-PENNACK"
259 GETE$:IFE$=""THEN259
260 IFE$="1"THENGOSUB55000:GOTO270
261 IFE$="2"THENGOSUB50000:GOTO270
262 IFE$="3"THENGOSUB55000:GOTO270
263 GOTO259
270 PRINT"QUESTION 2"
271 PRINT"WHAT WAS ELVIS COSTELLO'S FIRST HIT"
272 PRINT"IN BRITAIN ?"
273 PRINT"1:-RADIO RADIO"
274 PRINT"2:-OLIVER'S ARMY"
275 PRINT"3:-WATCHING THE DETECTIVES"
276 GETE$:IFE$=""THEN276
277 IFE$="1"THENGOSUB55000:GOTO281
278 IFE$="2"THENGOSUB55000:GOTO281
279 IFE$="3"THENGOSUB50000:GOTO281
280 GOTO276
```

```
281 PRINT"■■■■QUESTION 3"
282 PRINT"■■QUESTION 3"
283 PRINT"■WHAT WAS YAZOO'S FIRST HIT IN BRITAIN ?"
284 PRINT"■■1:-ONLY YOU"
285 PRINT"■2:-NOBODYS DIARY"
286 PRINT"■3:-BAD CONNECTION"
287 GETE$:IFE$=""THEN287
288 IFE$="1"THENGOSUB50000:GOTO300
289 IFE$="2"THENGOSUB55000:GOTO300
290 IFE$="3"THENGOSUB55000:GOTO300
292 GOTO287
300 PRINT"■■■■UESTION 4"
301 PRINT"■WHICH OF THE FOLLOWING ELVIS PRESLEY"
302 PRINT"HITS DIDN'T GET TO No.1 ?"
304 PRINT"■■1:-CRYING IN THE CHAPEL"
305 PRINT"■2:-TEDDY BEAR"
306 PRINT"■3:-ALL SHOOK UP"
307 GETE$:IFE$=""THEN307
308 IFE$="1"THENGOSUB55000:GOTO315
309 IFE$="2"THENGOSUB50000:GOTO315
310 IFE$="3"THENGOSUB55000:GOTO315
311 GOTO307
315 PRINT"■■■■UESTION 5"
316 PRINT"■WHEN WAS MADNESS'S FIRST HIT ?"
317 PRINT"■■1:-OCTOBER 1979"
318 PRINT"■2:-MARCH 1980"
319 PRINT"■3:-SEPTEMBER 1979
320 PRINT"■4:-AUGUST 1981
321 GETE$:IFE$=""THEN321
322 IFE$="1"THENGOSUB55000:GOTO330
323 IFE$="2"THENGOSUB55000:GOTO330
324 IFE$="3"THENGOSUB50000:GOTO330
325 IFE$="4"THENGOSUB55000:GOTO330
326 GOTO321
330 PRINT"■■■■UESTION 6"
331 PRINT"■WHICH OF THESE SONGS WAS NOT WRITTEN"
332 PRINT"BY LENNON AND McCARTNEY ?"
333 PRINT"■■1:-DRIVE MY CAR"
334 PRINT"■2:-TAXMAN"
335 PRINT"■3:-BIRTHDAY"
336 PRINT"■4:-I'M DOWN"
337 GETE$:IFE$=""THEN337
338 IFE$="1"THENGOSUB55000:GOTO350
339 IFE$="2"THENGOSUB50000:GOTO350
340 IFE$="3"THENGOSUB55000:GOTO350
341 IFE$="4"THENGOSUB55000:GOTO350
343 GOTO337
350 PRINT"■■■■UESTION 7"
351 PRINT"■WHICH ONE WAS CHUCK BERRY'S ONLY No.1"
352 PRINT"HIT IN BRITAIN ?"
353 PRINT"■■1:-SCHOOL DAY"
354 PRINT"■2:-MY DING-A-LING"
355 PRINT"■3:-NaDINE"
356 GETE$:IFE$=""THEN356
357 IFE$="1"THENGOSUB55000:GOTO365
358 IFE$="2"THENGOSUB50000:GOTO365
359 IFE$="3"THENGOSUB55000:GOTO365
360 GOTO356
```

```
365 PRINT"■■■■QUESTION 8"
366 PRINT"■'I WANNA HOLD YOUR HAND'  WHEN DID THIS"
367 PRINT"BECOME HIT FOR DOLLAR ?"
368 PRINT"■■1:-NOVEMBER 1979"
369 PRINT"■■2:-AUGUST 1979"
370 PRINT"■■3:-MAY 1979"
371 GETE$:IFE$=""THEN371
372 IFE$="1"THENGOSUB50000:GOTO380
373 IFE$="2"THENGOSUB55000:GOTO380
374 IFE$="3"THENGOSUB55000:GOTO380
375 GOTO371
380 PRINT"■■■■QUESTION 9"
381 PRINT"■WHAT IS BOB DYLAN'S REAL NAME ?"
382 PRINT"■■1:-ROBERT WOODY"
383 PRINT"■■2:-ROBERT PARKS"
384 PRINT"■■3:-ROBERT ZIMMERMAN"
386 GETE$:IFE$=""THEN386
387 IFE$="1"THENGOSUB55000:GOTO395
388 IFE$="2"THENGOSUB55000:GOTO395
389 IFE$="3"THENGOSUB50000:GOTO395
390 GOTO386
395 PRINT"■■■■QUESTION 10"
396 PRINT"■WHAT WAS BOB MARLEY AND THE WAILERS"
397 PRINT"FIRST HIT IN BRITAIN ?"
398 PRINT"■■1:-JAMMING"
399 PRINT"■■2:-NO WOMAN NO CRY"
400 PRINT"■■3:-EXODUS"
401 GETE$:IFE$=""THEN401
402 IFE$="1"THENGOSUB55000:GOTO410
403 IFE$="2"THENGOSUB50000:GOTO410
404 IFE$="3"THENGOSUB55000:GOTO410
406 GOTO401
410 PRINT"■■■■QUESTION 11"
411 PRINT"■APART FROM 'CARS' WHICH OTHER GARY"
412 PRINT"NUMAN SONG GOT TO No.1 ?"
413 PRINT"■■1:-WE ARE GLASS"
414 PRINT"■■2:-COMPLEX"
415 PRINT"■■3:-ARE 'FRIENDS' ELECTRIC"
416 GETE$:IFE$=""THEN416
417 IFE$="1"THENGOSUB55000:GOTO425
418 IFE$="2"THENGOSUB55000:GOTO425
419 IFE$="3"THENGOSUB50000:GOTO425
420 GOTO416
425 PRINT"■■■■QUESTION 12"
426 PRINT"■WHICH ONE BELOW WAS A No.1 HIT FOR THE"
427 PRINT"POLICE ?"
428 PRINT"■■1:-MESSAGE IN A BOTTLE"
429 PRINT"■■2:-SO LONELY"
430 PRINT"■■3:-CAN'T STAND LOSING YOU"
431 GETE$:IFE$=""THEN431
432 IFE$="1"THENGOSUB50000:GOTO440
433 IFE$="2"THENGOSUB55000:GOTO440
434 IFE$="3"THENGOSUB55000:GOTO440
435 GOTO431
440 PRINT"■■■■QUESTION 13"
441 PRINT"■WHO IS FUN BOY THREE'S LEAD SINGER ?"
442 PRINT"■■1:-NEVILLE STAPLES"
443 PRINT"■■2:-TERRY HALL"
444 PRINT"■■3:-LYNVAL GOLDING"
445 GETE$:IFE$=""THEN445
```

```
446 IFE$="1"THENGOSUB55000:GOTO455
447 IFE$="2"THENGOSUB50000:GOTO455
448 IFE$="3"THENGOSUB55000:GOTO455
450 GOTO445
455 PRINT"■■■■QUESTION 14"
456 PRINT"■WHICH ONE DID NOT GET TO No.1 ?"
457 PRINT"■■1:-JAILHOUSE ROCK : ELVIS PRESLEY"
458 PRINT"■2:-I'M A BELIEVER : MONKEES"
459 PRINT"■3:-IN THE MIDNIGHT HOUR :  WILSON"
460 PRINT"PICKETT"
461 GETE$:IFE$=""THEN461
462 IFE$="1"THENGOSUB55000:GOTO470
463 IFE$="2"THENGOSUB55000:GOTO470
464 IFE$="3"THENGOSUB50000:GOTO470
466 GOTO461
470 PRINT"■■■■QUESTION 15"
471 PRINT"■WHO HAD A HIT WITH 'YOU SEND ME' ?"
472 PRINT"■■1:-RUSS CONWAY"
473 PRINT"■2:-EAGLES"
474 PRINT"■3:-SAM COOKE"
475 PRINT"■4:-BILLY PRESTON"
476 GETE$:IFE$=""THEN476
477 IFE$="1"THENGOSUB55000:GOTO485
478 IFE$="2"THENGOSUB55000:GOTO485
479 IFE$="3"THENGOSUB50000:GOTO485
480 IFE$="4"THENGOSUB55000:GOTO485
481 GOTO476
485 PRINT"■■■■QUESTION 16"
486 PRINT"■WHO HAD A HIT WITH 'BAD BOYS' ?"
487 PRINT"■■1:-WHAM"
488 PRINT"■2:-DAVID BOWIE"
489 PRINT"■3:-KAJAGOOGOO"
490 PRINT"■4:-MICHAEL JACKSON"
491 GETE$:IFE$=""THEN491
492 IFE$="1"THENGOSUB50000:GOTO500
493 IFE$="2"THENGOSUB55000:GOTO500
494 IFE$="3"THENGOSUB55000:GOTO500
495 IFE$="4"THENGOSUB55000:GOTO500
496 GOTO491
500 PRINT"■■■■QUESTION 17"
501 PRINT"■CHUCK BERRY AND DAVE BERRY BOTH HAD"
502 PRINT"HIT WITH ?"
503 PRINT"■■1:-NADINE"
504 PRINT"■2:-MEMPHIS TENNESSEE"
505 PRINT"■3:-MAMA"
506 GETE$:IFE$=""THEN506
507 IFE$="1"THENGOSUB55000:GOTO515
508 IFE$="2"THENGOSUB50000:GOTO515
509 IFE$="3"THENGOSUB55000:GOTO515
510 GOTO506
515 PRINT"■■■■QUESTION 18"
516 PRINT"■JAMES BROWN'S FIRST HIT IN BRITAIN WAS ?"
517 PRINT"■■1:-I GOT YOU"
518 PRINT"■2:-BODY HEAT"
519 PRINT"■3:-PAPA'S GOT A BRAND NEW BAG"
520 GETE$:IFE$=""THEN520
521 IFE$="1"THENGOSUB55000:GOTO530
522 IFE$="2"THENGOSUB55000:GOTO530
523 IFE$="3"THENGOSUB50000:GOTO530
525 GOTO520
```

```
530 PRINT"■■■■QUESTION 19"
531 PRINT"■DIONNE WARWICK MADE A COME BACK TO THE"
532 PRINT"POP CHARTS IN 1982 WITH WHICH SONG "
533 PRINT"■■1:-WALK ON BY"
534 PRINT"■2:-HEARTBREAKER"
535 PRINT"■3:-DO YOU KNOW THE WAY TO SAN JOSE"
536 GETE$:IFE$=""THEN536
537 IFE$="1"THENGOSUB55000:GOTO545
538 IFE$="2"THENGOSUB50000:GOTO545
539 IFE$="3"THENGOSUB55000:GOTO545
540 GOTO536
545 PRINT"■■■■QUESTION 20"
546 PRINT"■NICK HEYWARD'S FIRST SOLO HIT WAS ?"
547 PRINT"■■1:-TAKE THAT SITUATION"
548 PRINT"■2:-WHISTLE DOWN THE WIND"
549 PRINT"■3:-FANTASTIC DAY"
550 GETE$:IFE$=""THEN550
551 IFE$="1"THENGOSUB55000:GOTO560
552 IFE$="2"THENGOSUB50000:GOTO560
553 IFE$="3"THENGOSUB55000:GOTO560
555 GOTO550
560 PRINT"■■■■QUESTION 21"
561 PRINT"■WHICH ONE OF THE FOLLOWING BOB MARLEY"
562 PRINT"AND THE WAILERS SONGS WAS A HIT IN 1983?"
563 PRINT"■■1:-BUFFALO SOLDIER"
564 PRINT"■2:-WAITING IN VAIN"
565 PRINT"■3:-IS THIS LOVE"
566 GETE$:IFE$=""THEN566
567 IFE$="1"THENGOSUB50000:GOTO575
568 IFE$="2"THENGOSUB55000:GOTO575
569 IFE$="3"THENGOSUB55000:GOTO575
570 GOTO566
575 PRINT"■■■■QUESTION 22"
576 PRINT"■CULTURE CLUB'S FIRST HIT WAS ?"
577 PRINT"■■1:-TIME(CLOCK OF THE HEART)"
578 PRINT"■2:-DO YOU REALLY WANT TO HURT ME"
579 PRINT"■3:-I'LL TUMBLE 4 YA"
580 GETE$:IFE$=""THEN580
581 IFE$="1"THENGOSUB55000:GOTO590
582 IFE$="2"THENGOSUB50000:GOTO590
583 IFE$="3"THENGOSUB55000:GOTO590
584 GOTO580
590 PRINT"■■■■QUESTION 23"
591 PRINT"■WHAT HAVE LITTLE RICHARD,JERRY LEE LEWIS"
592 PRINT"AND FATS DOMINO GOT 'N COMMON ?"
593 PRINT"■■1:-GUITAR"
594 PRINT"■2:-DRUMS"
595 PRINT"■3:-PIANO"
596 GETE$:IFE$=""THEN596
597 IFE$="1"THENGOSUB55000:GOTO605
598 IFE$="2"THENGOSUB55000:GOTO605
599 IFE$="3"THENGOSUB50000:GOTO605
600 GOTO596
605 PRINT"■■■■QUESTION 24 "
606 PRINT"■WHAT YEAR DID THE BEATLES ENTER THE"
607 PRINT"BRITISH CHARTS ?"
608 PRINT"■■1:-1963"
609 PRINT"■2:-1962"
610 PRINT"■3:-1961"
```

```
611 GETE$:IFE$=""THEN611
612 IFE$="1"THENGOSUB55000:GOTO620
613 IFE$="2"THENGOSUB50000:GOTO620
614 IFE$="3"THENGOSUB55000:GOTO620
615 GOTO611
620 PRINT"■■■■QUESTION 25"
621 PRINT"■WHICH YEAR DID DIANA ROSS GO SOLO ?"
622 PRINT"■■1:-1970"
623 PRINT"■2:-1969"
624 PRINT"■3:-1971"
625 GETE$:IFE$=""THEN625
626 IFE$="1"THENGOSUB55000:GOTO634
627 IFE$="2"THENGOSUB55000:GOTO634
628 IFE$="3"THENGOSUB55000:GOTO634
629 GOTO625
634 PRINT"■■■■QUESTION 26"
635 PRINT"■THESE ARTISTS HAD A HIT WITH THE SAME "
636 PRINT"SONG :-PAT BOONE , FATS DOMINO AND THE"
637 PRINT"FOUR SEASONS"
638 PRINT"■■1:-I'LL BE HOME"
639 PRINT"■2:-LET'S HANG ON"
640 PRINT"■3:-AIN'T THAT A SHAME"
641 GETE$:IFE$=""THEN641
642 IFE$="1"THENGOSUB55000:GOTO650
643 IFE$="2"THENGOSUB55000:GOTO650
644 IFE$="3"THENGOSUB50000:GOTO650
645 GOTO641
650 PRINT"■■■■QUESTION 27"
651 PRINT"■THE LEAD SINGER OF CULTURE CLUB IS ?"
652 PRINT"■■1:-GIRL GEORGE"
653 PRINT"■2:-BOY GEORGE"
654 PRINT"■3:-PETE BURNS"
655 GETE$:IFE$=""THEN655
656 IFE$="1"THENGOSUB55000:GOTO665
657 IFE$="2"THENGOSUB50000:GOTO665
658 IFE$="3"THENGOSUB55000:GOTO665
660 GOTO655
665 PRINT"■■■■QUESTION 28"
666 PRINT"■WHO PLAYS THE DRUMS IN GENESIS ?"
667 PRINT"■■1:-JOHN KEEBLE"
668 PRINT"■2:-KEITH MOON"
669 PRINT"■3:-PHIL COLLINS"
670 GETE$:IFE$=""THEN670
671 IFE$="1"THENGOSUB55000:GOTO680
672 IFE$="2"THENGOSUB55000:GOTO680
673 IFE$="3"THENGOSUB50000:GOTO680
675 GOTO670
680 PRINT"■■■■QUESTION 29"
681 PRINT"■WITH WHICH RECORD DID ROD STEWART GET"
682 PRINT"TO No.1 IN JUNE 1983 ?"
683 PRINT"■■1:-YOU WEAR IT WELL"
684 PRINT"■2:-PASSION"
685 PRINT"■3:-BABY JANE"
686 GETE$:IFE$=""THEN686
687 IFE$="1"THENGOSUB55000:GOTO695
688 IFE$="2"THENGOSUB55000:GOTO695
689 IFE$="3"THENGOSUB50000:GOTO695
690 GOTO686
```

```
695 PRINT"■■■■■QUESTION 30"
696 PRINT"■E.L.O. HAD A HIT IN 1983 WITH ?"
697 PRINT"■■1:-LONG LIVE ROCK 'N' ROLL"
698 PRINT"■2:-ROCK 'N' ROLL IS KING"
699 PRINT"■3:-ROCK 'N' ROLL QUEEN"
700 GETE$:IFE$=""THEN700
701 IFE$="1"THENGOSUB55000:GOTO710
702 IFE$="2"THENGOSUB50000:GOTO710
704 IFE$="3"THENGOSUB55000:GOTO710
705 GOTO700
710 PRINT"■■■■■QUESTION 31"
711 PRINT"■THE BEACH BOYS HAD A No.1 HIT IN 1966"
712 PRINT"WHAT WAS IT ?"
713 PRINT"■■1:-SURFIN' USA"
714 PRINT"■2:-BARBARA ANN"
715 PRINT"■3:-GOOD VIRBRATIONS"
716 GETE$:IFE$=""THEN716
717 IFE$="1"THENGOSUB55000:GOTO725
718 IFE$="2"THENGOSUB55000:GOTO725
720 IFE$="3"THENGOSUB50000:GOTO725
721 GOTO716
725 PRINT"■■■■■QUESTION 32"
726 PRINT"■WHO HAD HIT WITH THESE SONGS : BEN ; "
727 PRINT"AIN'T NO SUNSHINE ; GOT TO BE THERE "
728 PRINT"■■1:-JERMAINE JACKSON"
729 PRINT"■2:-JOE JACKSON"
730 PRINT"■3:-MICHAEL JACKSON"
731 GETE$:IFE$=""THEN731
732 IFE$="1"THENGOSUB55000:GOTO740
733 IFE$="2"THENGOSUB55000:GOTO740
734 IFE$="3"THENGOSUB50000:GOTO740
735 GOTO731
740 PRINT"■■■■■QUESTION 33"
741 PRINT"■WHAT WAS OTIS REDDING ONLY TOP TEN HIT ?"
742 PRINT"■■1:-MY GIRL"
743 PRINT"■2:-DOCK OF THE BAY"
744 PRINT"■3:-SHAKE"
745 GETE$:IFE$=""THEN745
746 IFE$="1"THENGOSUB55000:GOTO755
747 IFE$="2"THENGOSUB50000:GOTO755
748 IFE$="3"THENGOSUB55000:GOTO755
749 GOTO745
755 PRINT"■■■■■QUESTION 34"
756 PRINT"■WHICH DUO HAD A HIT WITH 'IT TAKES TWO'"
757 PRINT"IN 1967 ?"
758 PRINT"■■1:-MARVIN GAYE & KIM WESTON"
759 PRINT"■2:-MARVIN GAYE & DIANA ROSS"
760 PRINT"■3:-MARVIN GAYE & MARY WELLS"
761 GETE$:IFE$=""THEN761
762 IFE$="1"THENGOSUB50000:GOTO770
763 IFE$="2"THENGOSUB55000:GOTO770
764 IFE$="3"THENGOSUB55000:GOTO770
766 GOTO761
770 PRINT"■■■■■QUESTION 35"
771 PRINT"■WHICH DUO HAD A HIT WITH"
772 PRINT"'DON'T GO BREAKING MY HEART'  IN 1976 ?"
773 PRINT"■■1:-MARVIN GAYE & MARY WELLS"
774 PRINT"■2:-ELTON JOHN & KIKI DEE"
775 PRINT"■3:-DIANA ROSS & MICHAEL JACKSON"
776 GETE$:IFE$=""THEN776
777 IFE$="1"THENGOSUB55000:GOTO785
778 IFE$="2"THENGOSUB50000:GOTO785
779 IFE$="3"THENGOSUB55000:GOTO785
```

```
780 GOTO776
785 PRINT"■■■■■QUESTION 36"
786 PRINT"■WHO WAS THE FIRST TO HAVE A HIT WITH"
787 PRINT"'POISON IVY' ?"
788 PRINT"■■1:-LAMBRETTAS"
789 PRINT"■2:-COASTERS"
790 PRINT"■3:-PARAMOUNTS"
792 GETE$:IFE$=""THEN792
793 IFE$="1"THENGOSUB55000:GOTO800
794 IFE$="2"THENGOSUB50000:GOTO800
795 IFE$="3"THENGOSUB55000:GOTO800
797 GOTO792
800 PRINT"■■■■■QUESTION 37"
801 PRINT"■WHO HAD A HIT WITH 'RAVE ON'  ?"
803 PRINT"■■1:-BUDDY HOLLY"
804 PRINT"■2:-HOLLIES"
805 PRINT"■3:-BILLY JOEL"
806 GETE$:IFE$=""THEN806
807 IFE$="1"THENGOSUB50000:GOTO815
808 IFE$="2"THENGOSUB55000:GOTO815
809 IFE$="3"THENGOSUB55000:GOTO815
810 GOTO806
815 PRINT"■■■■■QUESTION 38 "
816 PRINT"■WHAT WAS THE KINKS'S FIRST No.1 HIT ?"
817 PRINT"■■1:-ALL DAY AND ALL OF THE NIGHT"
818 PRINT"■2:-SUNNY AFTERNOON"
819 PRINT"■3:-YOU REALLY GOT ME"
820 GETE$:IFE$=""THEN820
821 IFE$="1"THENGOSUB55000:GOTO830
822 IFE$="2"THENGOSUB55000:GOTO830
824 IFE$="3"THENGOSUB50000:GOTO830
826 GOTO820
830 PRINT"■■■■■QUESTION 39"
831 PRINT"■CLIFF RICHARD'S FIRST HIT WAS ?"
832 PRINT"■■1:-LIVIN' LOVIN' DOLL"
833 PRINT"■2:-HIGH CLASS BABY"
835 PRINT"■3:-MOVE IT"
836 GETE$:IFE$=""THEN836
837 IFE$="1"THENGOSUB55000:GOTO845
838 IFE$="2"THENGOSUB55000:GOTO845
839 IFE$="3"THENGOSUB50000:GOTO845
840 GOTO836
845 PRINT"■■■■■QUESTION 40"
846 PRINT"■HERE ARE SOME LYRIC'S FROM A SONG WHO"
847 PRINT"SUNG IT  ?"
848 PRINT"■■'GONNA TELL AUNT MARY 'BOUT UNCLE JOHN"
849 PRINT"HE SAYS HE HAS THE BLUES,BUT HE HAS A "
850 PRINT"A LOT OF FUN.  OH BA-BY'"
851 PRINT"■■1:-LONG TALL SALLY : LITTLE RICHARD"
852 PRINT"■2:-JOHNNY B. GOODE : CHUCK BERRY"
853 PRINT"■3:-SUMMERTIME BLUES : EDDIE COCHRAN"
854 GETE$:IFE$=""THEN854
855 IFE$="1"THENGOSUB50000:GOTO890
856 IFE$="2"THENGOSUB55000:GOTO890
857 IFE$="3"THENGOSUB55000:GOTO890
858 GOTO854
890 PRINT"■■■■■QUESTION 41"
891 PRINT"■HERE ARE SOME LYRIC'S FROM A SONG WHO"
892 PRINT"SUNG IT ?"
893 PRINT"■■'ALL WE NEED IS MUSIC,SWEET MUSIC"
894 PRINT"THERE'LL BE MUSIC EVERYWHERE"
895 PRINT"THERE'LL BE SWINGING AND SWAYING AND"
```

```
896 PRINT"RECORDS PLAYING'"
897 PRINT"■■1:-WHAT A PARTY : FATS DOMINO"
898 PRINT"■2:-THATS WHAT I WANT : MARAUDERS"
899 PRINT"■3:-DANCING IN THE STREET :MARTHA REEVES"
900 PRINT"AND THE VANDELLAS"
901 GETE$:IFE$=""THEN901
902 IFE$="1"THENGOSUB55000:GOTO910
903 IFE$="2"THENGOSUB55000:GOTO910
904 IFE$="3"THENGOSUB50000:GOTO910
905 GOTO901
910 PRINT"■■■■QUESTION 42 "
911 PRINT"■HERE ARE SOME LYRIC'S FROM A SONG WHO"
912 PRINT"SUNG IT ?"
913 PRINT"■■'DON'T LEAVE ME HANGING ON THE LINE"
914 PRINT"I KNEW YOU WHEN YOU HAD NO ONE TO TALK"
915 PRINT"TO, NOW YOU ARE MOVING IN HIGH SOCIETY"
916 PRINT"■■1:-COME LIVE WITH ME : HEAVEN 17"
917 PRINT"■2:-BABY JANE : ROD STEWART"
918 PRINT"■3:-IT'S OVER : FUNK MASTERS"
919 GETE$:IFE$=""THEN919
920 IFE$="1"THENGOSUB55000:GOTO925
921 IFE$="2"THENGOSUB50000:GOTO925
922 IFE$="3"THENGOSUB55000:GOTO925
923 GOTO919
925 PRINT"■■■■QUESTION 43"
926 PRINT"■HERE ARE SOME LYRIC'S FROM A SONG WHO"
927 PRINT"SUNG IT ?"
928 PRINT"■■'SOMEONE GLIMPSED ACROSS THE DANCE  "
929 PRINT"FLOOR NOT GOING HOME AND LOVING IN  "
930 PRINT"DOORWAYS, A ROOM TO REMEMBER WHO TO MEET"
931 PRINT"IN, SECRETS RUN THROUGH YOUR HEAD....."
932 PRINT"■■1:-WE CAME TO DANCE : ULTRAVOX"
933 PRINT"■2:-DREAM TO SLEEP : H2O"
934 PRINT"■3:-WHEN WE WERE YOUNG :BUCK FIZZ"
935 GETE$:IFE$=""THEN935
936 IFE$="1"THENGOSUB55000:GOTO942
937 IFE$="2"THENGOSUB50000:GOTO942
938 IFE$="3"THENGOSUB55000:GOTO942
940 GOTO935
942 PRINT"■■■■QUESTION 44"
943 PRINT"■WHO HAD A HIT WITH THE SONG"
944 PRINT"'THE GREAT PRETENDER'     ?"
945 PRINT"■■1:-DRIFTERS "
946 PRINT"■2:-COASTERS "
947 PRINT"■3:-PLATTERS"
948 GETE$:IFE$=""THEN948
949 IFE$="1"THENGOSUB55000:GOTO955
950 IFE$="2"THENGOSUB55000:GOTO955
951 IFE$="3"THENGOSUB50000:GOTO955
952 GOTO948
955 PRINT"■■■■QUESTION 45"
956 PRINT"■DION HAD A HIT WITH 'RUNAROUND SUE' IN"
957 PRINT"1961. WHO HAD A HIT WITH IT IN 1980 ?"
958 PRINT"■■1:-RACEY"
959 PRINT"■2:-SHOWADDYWADDY"
960 PRINT"■3:-MUD"
961 GETE$:IFE$=""THEN961
962 IFE$="1"THENGOSUB50000:GOTO970
964 IFE$="2"THENGOSUB55000:GOTO970
966 IFE$="3"THENGOSUB55000:GOTO970
967 GOTO961
```

```
970 PRINT"■■■■■QUESTION 46"
971 PRINT"■WHICH ARTIST HAD THE MOST WEEKS IN "
972 PRINT"THE CHARTS IN 1956    ?"
973 PRINT"■■1:-BILLY HALEY"
974 PRINT"■2:-ELVIS PRESLEY"
975 PRINT"■3:-RUBY MURRY"
976 GETE$:IFE$=""THEN976
977 IFE$="1"THENGOSUB50000:GOTO985
978 IFE$="2"THENGOSUB55000:GOTO985
979 IFE$="3"THENGOSUB55000:GOTO985
980 GOTO976
985 PRINT"■■■■■QUESTION 47"
986 PRINT"■WHAT WAS SHALAMAR'S FIRST HIT IN BRITAIN"
988 PRINT"■■1:-I OWE YOU ONE"
989 PRINT"■2:-TAKE THAT TO THE BANK"
990 PRINT"■3:-UPTOWN FESTIVAL"
992 GETE$:IFE$=""THEN992
993 IFE$="1"THENGOSUB55000:GOTO1000
994 IFE$="2"THENGOSUB55000:GOTO1000
995 IFE$="3"THENGOSUB50000:GOTO1000
996 GOTO992
1000 PRINT"■■■■■QUESTION 48"
1002 PRINT"■UB40'S FIRST HIT WAS ?"
1003 PRINT"■■1:-MY WAY OF THINKING"
1004 PRINT"■2:-KING"
1005 PRINT"■3:-THE EARTH DIES SCREAMING"
1006 GETE$:IFE$=""THEN1006
1007 IFE$="1"THENGOSUB55000:GOTO1015
1008 IFE$="2"THENGOSUB50000:GOTO1015
1009 IFE$="3"THENGOSUB55000:GOTO1015
1010 GOTO1006
1015 PRINT"■■■■■QUESTION 49"
1016 PRINT"■IN WHICH COUNTRY WAS KID CREOLE BORN ?"
1017 PRINT"■■1:-HAITI"
1018 PRINT"■2:-AMERICA"
1019 PRINT"■3:-FRANCE"
1020 GETE$:IFE$=""THEN1020
1021 IFE$="1"THENGOSUB50000:GOTO1030
1023 IFE$="2"THENGOSUB55000:GOTO1030
1025 IFE$="3"THENGOSUB55000:GOTO1030
1028 GOTO1020
1030 PRINT"■■■■■QUESTION 50"
1032 PRINT"■VINCE CLARKE LEFT A GROUP TO FORM YAZOO"
1033 PRINT"NAME THE GROUP HE LEFT ?"
1035 PRINT"■■1:-ORANGE JUICE"
1036 PRINT"■2:-MUSICAL YOUTH"
1037 PRINT"■3:-DEPECHE MODE"
1038 GETE$:IFE$=""THEN1038
1039 IFE$="1"THENGOSUB55000:GOTO1045
1040 IFE$="2"THENGOSUB55000:GOTO1045
1041 IFE$="3"THENGOSUB50000:GOTO1045
1042 GOTO1038
1045 PRINT"■■■■■QUESTION 51"
1046 PRINT"■WHICH FAMOUS ROCK 'N' ROLL FILM DID"
1047 PRINT"JAYNE MANSFIELD STAR IN    ?"
1048 PRINT"■■1:-MISTER ROCK 'N' ROLL"
1049 PRINT"■2:-THE GIRL CAN'T HELP IT"
1050 PRINT"■3:-ROCK AROUND THE CLOCK"
1051 GETE$:IFE$=""THEN1051
1052 IFE$="1"THENGOSUB55000:GOTO1060
1054 IFE$="2"THENGOSUB50000:GOTO1060
1055 IFE$="3"THENGOSUB55000:GOTO1060
1058 GOTO1051
```

```
1060 PRINT"■■■■QUESTION 52"
1061 PRINT"■THE EVERLY BROTHERS HIT 'BIRD DOG'"
1062 PRINT"ENTERED THE  CHARTS IN ?"
1063 PRINT"■■1:-AUGUST 1957"
1064 PRINT"■2:-MARCH 1958"
1065 PRINT"■3:-SEPTEMBER 1958"
1066 GETE$:IFE$=""THEN1066
1067 IFE$="1"THENGOSUB55000:GOTO1075
1068 IFE$="2"THENGOSUB55000:GOTO1075
1069 IFE$="3"THENGOSUB50000:GOTO1075
1072 GOTO1066
1075 PRINT"■■■■QUESTION 53"
1076 PRINT"■ADAM AND THE ANTS FIRST HIT WAS "
1078 PRINT"■■1:-KING OF THE WILD FRONTIER"
1079 PRINT"■2:-ANTMUSIC"
1081 PRINT"■3:-DOG EAT DOG"
1082 GETE$:IFE$=""THEN1082
1083 IFE$="1"THENGOSUB50000:GOTO1090
1084 IFE$="2"THENGOSUB55000:GOTO1090
1085 IFE$="3"THENGOSUB55000:GOTO1090
1086 GOTO1082
1090 PRINT"■■■■QUESTION 54"
1091 PRINT"■WHAT WAS BAD MANNERS NEXT HIT AFTER"
1092 PRINT"'LIP UP FATTY'"
1093 PRINT"■■1:-LORRAINE"
1094 PRINT"■2:-NE-NE-NA-NA-NU-NU"
1095 PRINT"■3:-SPECIAL BREW"
1096 GETE$:IFE$=""THEN1096
1097 IFE$="1"THENGOSUB55000:GOTO1105
1098 IFE$="2"THENGOSUB55000:GOTO1105
1099 IFE$="3"THENGOSUB50000:GOTO1105
1100 GOTO1096
1105 PRINT"■■■■QUESTION 55"
1106 PRINT"■WHO IS THE LEAD SINGER OF THE "
1107 PRINT"ROLLING STONES  ?"
1108 PRINT"■■1:-MICKY DOLENZ"
1109 PRINT"■2:-KEITH RICHARD"
1110 PRINT"■3:-MICK JAGGER"
1111 GETE$:IFE$=""THEN1111
1112 IFE$="1"THENGOSUB55000:GOTO1120
1113 IFE$="2"THENGOSUB55000:GOTO1120
1114 IFE$="3"THENGOSUB50000:GOTO1120
1115 GOTO1111
1120 PRINT"■■■■QUESTION 56 "
1121 PRINT"■WHO IS THE LEAD SINGER OF DURAN DURAN ?"
1122 PRINT"■■1:-PAUL SIMON"
1123 PRINT"■2:-SIMON LE BON"
1124 PRINT"■3:-JOHN TAYLOR"
1125 GETE$:IFE$=""THEN1125
1126 IFE$="1"THENGOSUB55000:GOTO1140
1127 IFE$="2"THENGOSUB50000:GOTO1140
1128 IFE$="3"THENGOSUB55000:GOTO1140
1130 GOTO1125
1140 PRINT"■■■■QUESTION 57"
1141 PRINT"■WHAT WAS THE MOJO'S ONLY TOP TEN HIT ?"
1142 PRINT"■■1:-EVERTHING'S ALRIGHT"
1143 PRINT"■2:-SEVEN DAFFODILS"
1144 PRINT"■3:-WHY NOT TONIGHT"
1145 GETE$:IFE$=""THEN1145
1146 IFE$="1"THENGOSUB50000:GOTO1155
1147 IFE$="2"THENGOSUB55000:GOTO1155
1148 IFE$="3"THENGOSUB55000:GOTO1155
```

```
1149 GOTO1145
1155 PRINT"■■■■QUESTION 58"
1156 PRINT"■JAPAN HAD A HIT WITH A SPOOKY SONG"
1158 PRINT"WHAT WAS IT ?"
1159 PRINT"■■1:-SPIRIT"
1160 PRINT"■2:-GHOSTS"
1161 PRINT"■3:-DUPPY"
1162 GETE$:IFE$=""THEN1162
1163 IFE$="1"THENGOSUB55000:GOTO1170
1164 IFE$="2"THENGOSUB50000:GOTO1170
1165 IFE$="3"THENGOSUB55000:GOTO1170
1168 GOTO1162
1170 PRINT"■■■■QUESTION 59"
1171 PRINT"■WHICH ONE OF THESE ALBUM TITLES IS NOT"
1172 PRINT"THE WORK OF ALTERED IMAGES  ?"
1173 PRINT"■■1:-PINKY BLUE"
1174 PRINT"■2:-KISSING TO BE CLEVER"
1175 PRINT"■3:-BITE"
1176 GETE$:IFE$=""THEN1176
1177 IFE$="1"THENGOSUB55000:GOTO1190
1178 IFE$="2"THENGOSUB50000:GOTO1190
1179 IFE$="3"THENGOSUB55000:GOTO1190
1180 GOTO1176
1190 PRINT"■■■■QUESTION 60"
1191 PRINT"■WHAT IS THE LINK BETWEEN SOFT CELL "
1192 PRINT"AND THE SUPREMES   ?"
1193 PRINT"■■1:-STONED LOVE"
1194 PRINT"■2:-WHERE DID OUR LOVE GO"
1195 PRINT"■3:-BABY LOVE"
1196 GETE$:IFE$=""THEN1196
1198 IFE$="1"THENGOSUB55000:GOTO5000
1199 IFE$="2"THENGOSUB50000:GOTO5000
1200 IFE$="3"THENGOSUB55000:GOTO5000
1201 GOTO1196
5000 PRINT"■"
5002 R1$="+C3+D+#F+E1+#F+E1+#F"
5004 R2$="+C3+C+#A+B1+#A+G1+#F"
5006 TEMPO5
5007 MUSICR1$;R2$;R1$;R2$;R1$;R2$
5009 PRINTA$;"  YOUR SCORE IS :-";S;" OUT OF 60"
5010 IFS=60THEN5015
5011 IFS<60THEN5018
5015 FORT=1TO2500:NEXT:GOTO5100
5018 FORT=1TO2500:NEXT:GOTO6000
5100 FORW=1TO1
5102 TEMPO5
5104 MUSIC"C1+C+E+C+G+E+C+ECA+DA+F+DA+D"
5106 NEXTW
5108 PRINT"■"
5110 PRINT"■■■■■";A$;" YOU HAVE SCORED A MAXIMUM OF 60"
5120 GOTO6000
6000 PRINT"■■■DO YOU WANT ANOTHER GAME (Y/N) ?"
6005 GETY$:IFY$=""THEN6005
6007 IFY$="Y"THENRUN
6009 IFY$="N"THENEND
6010 GOTO6005
50000 S=S+1
50005 MUSIC"+#B+A+#G+#B"
50010 PRINT"■■";A$;"  YOU ARE RIGHT"
50012 FORT=1TO1500:NEXT
50014 RETURN
55000 MUSIC"-B2-D2"
55002 PRINT"■■";A$;"  YOU ARE WRONG"
55004 FORT=1TO1000:NEXT
55005 RETURN
```

```
. GOSUB5000:M=100
2 PRINT"■"
3 P=INT(7*RND(1))+1
4 IFP<=3THENRR$="Slow"
5 IFP=4THENRR$="Medium"
6 IFP=5THENRR$="Medium"
7 IFP=>5THENRR$="Fast"
8 IFM=0THEN1000
10 TEMPOP:INPUT"WHICH HORSE DO YOU BET ON ? (1 TO 5)";PP
11 IFPP>5THEN10
15 IFM=0THEN1000
20 PRINT"■YOU HAVE";M;"■POUNDS."
30 INPUT"■HOW MUCH DO YOU GAMBLE ?";FF
40 IFFF>MTHENPRINT"■SORRY I'M AFRAID YOU HAVEN'T GOT £";FF:GOTO30
50 M=M-FF
70 DIMA$(23)
80 A$(1)="RAVEN'S CHOICE":A$(2)="WHAT A PILLOCK":A$(3)="MIDNIGHT RACER"
82 A$(4)="MOONSHINE":A$(5)="WILL SHE DO IT":A$(6)="JO-JO":A$(7)="RACER"
83 A$(8)="BABY BOY":A$(9)="MANS FIRST SON":A$(10)="GOLLY GHOST"
84 A$(11)="TRIUMPH":A$(12)="BAD NEWS":A$(13)="RIGHTING RAMBLER"
85 A$(14)="LEGS ELEVEN":A$(15)="LOVELY STARTER":A$(16)="STARBURST"
86 A$(17)="ROMANCANDLE":A$(18)="SKITTLES":A$(19)="PULLER BACK"
87 A$(20)="GOOD LUCK":A$(21)="NIGHT MERE":A$(22)="TOO FAST"
88 A$(23)="BEST NIGHT OUT"
89 Y=INT(23*RND(1))+1
90 PRINT"■YOUR HORSES NAME IS...."
95 PRINT"■";A$(Y)
100 PRINT"■PRESS ANY KEY TO PLAY"
110 GET R$:IFR$=""THEN110
120 PRINT"■Name of horse■";A$(Y)
130 PRINT"■In lane number";PP;"■■■■■■■Pace=";RR$
140 REM**NUMBER 1**
150 A=5:B=0
160 REM**NUMBER 2**
170 C=9:D=0
180 REM**NUMBER 3**
190 E=13:F=0
200 REM**NUMBER 4**
210 G=17:H=0
220 REM**NUMBER 5**
230 I=21:J=0
240 FORB=0TO39:PRINT@A,B;"~":NEXT
250 FORD=0TO39:PRINT@C,D;"~":NEXT
260 FORF=0TO39:PRINT@E,F;"~":NEXT
270 FORH=0TO39:PRINT@G,H;"~":NEXT
280 FORJ=0TO39:PRINT@I,J;"~":NEXT
290 A=A-1:C=C-1:E=E-1:G=G-1:I=I-1
292 B=0:D=0:F=0:H=0:J=0
300 PRINT@A,B;CHR$(97);@C,D;CHR$(97);@E,F;CHR$(97);@G,H;CHR$(97)
310 PRINT@I,J;CHR$(97)
320 T=INT(5*RND(1))+1
330 IFT=1THENPRINT@A,B;" ":MUSIC"C0":B=B+1:IFB=39THEN2000
335 IFT=1THENPRINT@A,B;CHR$(97)
340 IFT=2THENPRINT@C,D;" ":MUSIC"D0":D=D+1:IFD=39THEN2000
345 IFT=2THENPRINT@C,D;CHR$(97)
350 IFT=3THENPRINT@E,F;" ":MUSIC"E0":F=F+1:IFF=39THEN2000
355 IFT=3THENPRINT@E,F;CHR$(97)
360 IFT=4THENPRINT@G,H;" ":MUSIC"F0":H=H+1:IFH=39THEN2000
365 IFT=4THENPRINT@G,H;CHR$(97)
370 IFT=5THENPRINT@I,J;" ":MUSIC"G0":J=J+1:IFJ=39THEN2000
375 IFT=5THENPRINT@I,J;CHR$(97)
```

```
380 GOTO320
1000 PRINT"■                    ■
1005 PRINT"■  ■   ◢■◣   ■ ■ /  ■■■ ■▀▀
1010 PRINT"◥■◤  ■ ■   ■ ■    ■ ■ ■▬
1020 PRINT" ■   ■ ■   ■ ■    ■■■ ■
1030 PRINT" ■   ▼▼   ▼▼    ■ ▼ ■■■
1080 PRINT"■■■◢■◣  ■     ■■  ■■ ■ ◢■■◣
1090 PRINT"■    ■ ◢  ■   ■ ■ ■   ■
1100 PRINT"▼■◣  ■◢   ■   ■ ▮ ■   ■
1110 PRINT"  ■  ■    ■   ■ ■■   ■
1120 PRINT"▼■   ■▼◣  ■■  ■  ■   ■
1130 PRINT"■■THAT'S THE TRADE MARK OF A USELESS"
1140 PRINT"■GAMBLER IF YOU WANT TO MAKE MORE THAN "
1150 PRINT"■BIG FAT NOTHING THEN PRESS THE LETTER S "
1160 GETR$:IFR$="S"THENM=100:GOTO2
1170 GOTO1160
1900 GOTO1900
2000 IFB=39THENG=1
2010 IFD=39THEND=2
2020 IFF=39THENO=3
2030 IFH=39THENO=4
2040 IFJ=39THENO=5
2050 IFPP=0THEN2070
2055 PRINT"■HARD LUCK SONNY!"
2056 PRINT"■YOUR HORSE DIDN'T COME FIRST DID IT ?"
2057 GOTO2120
2060 GOTO2
2070 PRINT"■WELL DONE!!!"
2090 PRINT"■YOUR HORSE HAS WON."
2100 PRINT"■YOU BETTED £";FF
2110 PRINT"■AT ODDS OF 4-1 YOU HAVE WON £";4*FF
2115 M=M+4*FF
2120 PRINT"■YOU HAVE A TOTAL OF £";M
2130 INPUT"■DO YOU WANT TO STOP BETTING ?";BB$
2140 IFBB$="Y"THEN2156
2150 IFBB$="N"THEN2
2155 GOTO2130
2156 IFM=>11ANDM<=20THENL$="RUBBISH"
2157 IFM=>2ANDM<=10THENL$="PATHETIC"
2158 IFM=>21ANDM<=100THENL$="PRETTY GOOD"
2159 IFM=>101ANDM<=250THENL$="GOOD"
2160 IFM=>251ANDM<=500THENL$="VERY GOOD"
2161 IFM=>501ANDM<=1000THENL$="BRILLIANT"
2162 IFM=>1001THENL$="SUPER SONIC"
2163 IFM=0THEN1000
2180 PRINT"■YOU HAVE £";M;" LEFT.
2190 PRINT"■THIS MAKES YOU A■";L$;"■GAMBLER"
2200 PRINT"■PRESS 'S' TO PLAY AGAIN"
2210 GETPR$:IFPR$="S"THENM=100:GOTO2
2220 GOTO2210
5000 PRINT"■              HORSE RACING"
5010 PRINT"              ▔▔▔▔▔▔▔▔▔▔▔▔"
5020 PRINT"■■You are a horse racing gambler and   "
5030 PRINT"can't resist a little flutter now and  "
5040 PRINT"then.You will be betting on horses with"
5050 PRINT"such famous names as LEGS ELEVEN and   "
5060 PRINT"FAST STARTER."
5070 PRINT"■I am going to give you £100 so spend it"
5080 PRINT"slowly!The horses will be in lanes,lane"
5090 PRINT"1 will be at the top of the screen and "
5100 PRINT"lane 5 at the bottom.There will also be"
5110 PRINT"a PACER and this will determine at which"
5120 PRINT"■speed the horses will race."
5130 PRINT"■When you are ready press S and good luck"
5140 GETWE$:IFWE$="S"THENRETURN
5150 GOTO5140
```

EUROLOCK MII
for the
MZ-80A
by N. Alam

```
1 Z3=0:POKE6637,8
2 IFPEEK(4975)<>0THENZ3=1:GOSUB65036
3 IFZ3THENPOKE8768,60:POKE8769,34:POKE7935,183:POKE7936,30
4 IFZ3THENPOKE5415,0:POKE17357,0:POKE17352,0:POKE17342,0:POKE4687,68
5 IFZ3THENPOKE4874,169:POKE4875,85:POKE4876,72:POKE4877,79:POKE4878,32
6 IFZ3THENPOKE4879,32:POKE4857,76:POKE4858,79:POKE4859,67:POKE4860,75
7 IFZ3THENPOKE4861,32
8 IFZ3THENPRINT"▣";:FORI=1TO23:PRINT"▒";TAB(38);"▒":NEXT
9 IFZ3THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ "
10 IFZ3THENPRINT"▒▒▒▒▒▒▒▒This is program is protected by : "
11 IFZ3THENPRINT"▒▒▒▒▒▒▒ E U R O   L O C K   MK II "
12 IFZ3THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒Written by N.Alam"
13 IFZ3THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒────────── ──── ────────"
14 IFZ3THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ "
15 IFZ3THENPRINT"▒▒▒▒▒▒▒▒▒▒▒▒PRESS ANY KEY TO CONTINUE";:USR(2483)
16 POKE10407,0
65000 REM ** ENABLE BASIC **
65007 A=8768:POKEA,205:POKEA+1,192:POKE7935,190:POKE7936,202:A=0
65014 POKE5415,35:POKE19357,0:POKE17352,0:POKE17342,0:POKE4687,0
65021 POKE4874,98:POKE4875,23:POKE4876,205:POKE4877,195:POKE4878,25
65028 POKE4879,235:POKE4857,74:POKE4858,205:POKE4859,242:POKE4861,205
65035 POKE4975,32:POKE5343,51:USR(4514)
65036 A=4937:POKEA,69:POKEA+1,85:POKEA+2,82:POKEA+3,79:POKEA+4,76:POKEA+5,7
65037 POKEA+6,67:POKEA+7,75:POKEA+8,32:POKEA+9,83:POKEA+10,89:POKEA+11,83
65038 POKEA+12,84:POKEA+13,69:POKEA+14,77:POKEA+15,32:POKEA+16,65:POKEA+17,
65039 POKEA+18,84:POKEA+19,73:POKEA+20,86:POKEA+21,65:POKEA+22,84:POKEA+23,
65040 POKEA+24,68:POKEA+25,32
65041 POKE4980,51:POKE4985,78:POKE4986,46:POKE4987,65:POKE4988,76:POKE4989,
65042 POKE4989,77:POKE4990,32:POKE4991,32:RETURN
```

SHARPSOFT USER NOTES

1984 S U B S C R I P T I O N F O R M

Please complete this form in BLOCK CAPITALS.

Membership No: / / / /

NAME: .....

ADDRESS: .....

.....

.....

.....

1984 Membership @ £5.50 (£12.00 overseas) |_|

1983 Back Issues @ £5.50 (£10.00 overseas) |_|

1982 Back Issues @ £7.50 (£12.00 overseas) |_|

1981 Back Issues @ £3.00 (£6.00 overseas) |_|

I enclose Cheque.P.O. made payable to SHARPSOFT LTD for £.....

Charge my Access/Barclaycard No: |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|

Expiry Date / /

Signature .....

Please complete the section below so that we may endeavour to cater for all our Members' interests and future requirements:

Please indicate your system type:

MZ-80A MZ-80B MZ-80K

Whether you have any of the following:

EXPANSION UNIT ONLY

FLOPPY DISCS

PRINTER P3 P4 P5 P6 EPSON DAISYWHEEL OTHER

OTHER

Which area(s) your system is used in:

PERSONAL BUSINESS EDUCATION MEDICAL

SHARPSOFT Sharpsoft Ltd., 86-90 Paul Street, London EC2A 4NE
Printed by Oldham Press (T.U.), Chatham, Kent.