



User Notes No 17

BEGINNERS' TUTORIAL GUIDE TO PASCAL

PART - 6

SHARPSOFT USER NOTES

SHARPSOFT USER NOTES

ISSUE NO:17

C O N T E N T S

| | <u>PAGE NO</u> |
|---|----------------|
| Issue 17 Editorial | 2 |
| PASCAL | |
| Beginners' Tutorial Guide to PASCAL - Part 6 | 3 |
| PASCAL letter writing program by D. Willey | 5 |
| MZ-80K NOTES, LETTERS AND LISTINGS | |
| Dimensional Analysis by Dr. Alan Stevens | 14 |
| Interrupting, Pardon? Interrupting the MZ-80K by William Howard. | 26 |
| A Business Game by Roger Hinton | 43 |
| MZ-80A NOTES, LETTERS AND LISTINGS | |
| Music Composer and Suicide Run by Simon Jones | 59 64 |
| Flying Mission and Spatial Hunt by Letizia Bizzari | 71 72 |
| MZ-80B NOTES, LETTERS AND LISTINGS | |
| Tape Loader by B.L. Page | 74 |
| Graphics - Stereoscopic slicing by J. Arens | 76 |
| READERS' LETTERS | 79 |

SHARPSOFT USER NOTES

ISSUE NO:17

This Issue should reach readers before the holiday season gets into full swing. The period July to September tends to be a "slack" time for the micro industry - perhaps this Issue will stimulate our members into writing new and exciting programs? Please send your efforts to SHARPSOFT. Our end of year Issue could be a "bumper edition" with your help. Learning to program in Z80 machine code or understanding the operation of your computer's hardware is a difficult but interesting task. One of our main feature articles in this Issue allows readers an insight into the operation of the Z80 microprocessor.

In his article William Howard carefully explains, and demonstrates with examples, how the Z80 interrupts may be used - do read this article and try the programs.

In the real world physical objects have a definite size - Dr. Alan Stevens in his article on Dimensional Analysis demonstrates how a personal computer can be used to investigate the relationships between different systems of physical units.

Once again other sections of this Issue are devoted to more listings and, of course, your letters.

The next Issue in the 1985 series will be published towards the end of 1985.

Mike Brinson

EDITOR

BEGINNER'S TUTORIAL GUIDE TO PASCAL

PART - 6

More Data Types continued

44. RECORDS.

A records components can be of different types.

TYPE

```
<IDENTIFIER> = RECORD
    <field list>
END;
```

For each field in <field list>

```
<field name> : type;
```

For example

```
TYPE
    TICKET = RECORD
        THEATRE : ARRAY [1..15] OF CHAR;
        BLOCK   : (LEFT, RIGHT, CENTER);
        ROW      : 1..20;
        SEATNO   : 1..100;
        DATE     : RECORD
            YEAR  : 1985..1990;
            MONTH : 1..12;
            DAY   : 1..31
        END
    END;
```

VAR

```
TICKET1, TICKET2, TICKET3 : TICKET;
```

45. Accessing RECORD fields.

syntax <RECORD name> . <field name>

For example:-

```
TICKET2.ROW      : = 15;
TICKET1.SEATNO  : = 99;
TICKET3.THEATRE : = 'ODEON';
TICKET2.DATE.MONTH : = 5;
TICKET1.DATE.DAY   : = 10;
TICKET3.DATE.YEAR  : = 1985;
```

46. The WITH statement
- a short-hand notation

```
WITH TICKET1 DO
BEGIN
  THEATRE := 'OBEON';
  BLOCK  := RIGHT;
  ROW    := 12;
  SEAT   := 5;
  WITH DATE DO
  BEGIN
    MONTH := 5;
    DAY   := 7;
    YEAR  := 1985;
  END
END;
```

47. Hisoft PASCAL does not allow
VARIANT RECORDS.

48. SETS

TYPE

<IDENTIFIER> = SET OF <ordinal type>

for example:-

TYPE

```
LINES = SET OF PAGESIZE;
CHARS = SET OF LINESIZE;
```

VAR

```
LINESC : LINES;
CHARSC : CHARS;
```

49. Values

1. All possible subsets of the base type.
2. Empty set - denoted by [].

50. Operations

Assignment :=

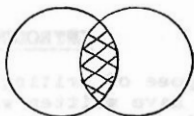
Operators

1. + union



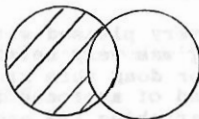
A + B

2. * intersection



A * B

3. - difference



A - B
(Remove from
A all the
elements which
are also in B)

Relational Operators

1. equality =
2. inequality <>
3. "contains" >=
4. "is contained in" <=
5. "is a member of" IN
 $\langle \text{value} \rangle$ IN $\langle \text{set variable} \rangle$

For example:

VAR

CH : CHAR;

CH: = 1 1;

REPEAT READ(CH) UNTIL CH IN ['A','B','C'];

More PASCAL next Issue

PASCAL LETTER WRITING PROGRAM

by MR. D. WILLEY

MACHINE MZ-80K (48K) with P3 PRINTER

LANGUAGE PASCAL SP-4015

PROGRAM SUMMARY Menu driven with the following options:

SETTING the page margins and length of line.

SAVING the type page on a file.

LOADING from the file (obviously!).

ALTERING by re-entering any line found by No.

VIEWING the page on the CRT.

PRINTING the page on the P3 PRINTER.

INTRODUCTION

The main purpose of writing this letter is to demonstrate the program that I have written while I have been learning how to handle the PASCAL language which I recently bought from you.

I have been very pleased with PASCAL and the enforced neatness of programming was very helpful when developing and amending it. One reason for doing this kind of program was the challenge of a different kind of approach that was required because of the lack of string variables (A\$ and all that) and of course the problem of putting in a string from the keyboard when on the run. It may be of help to early users of PASCAL who have not yet got over that particular hurdle.

I have not used a mix of upper and lower case as I find personally that the capitals are more legible but the program is capable of lower case. The switching between small and caps is done by poking, a tip that came from the User Notes. Otherwise pressing the usual controls gave pictures of little men or cars!

No attempt has been made to keep the program length down, rather the aim has been to make it clear and easy to amend.

INSTRUCTIONS FOR LETTER PROGRAM

When the Menu is displayed, press the numbers corresponding to the required action. Either set the page format to suit your requirements or leave it to set itself to a standard layout. If a page is loaded from a file then initialising is not needed.

On typing the letter the line numbers shown are numbered from the top of the page (not from the first line entered).

If when requested to enter the page number '1' is entered then the program will automatically format the address entry and will also indent the first line of the letter.

A marker will indicate where the end of the line is coming up. When entry is finished before the end of the page is reached the entry of the star (multiply/asterisk) character will end the letter, otherwise entry will stop when the line number reaches the foot margin.

It is better to save the typed page immediately otherwise there is a possibility of losing it if the wrong number (e.g. 1 to initialise) is pressed or the printer is not switched on if you go to that option - SO BEWARE!

The page may be viewed on the CRT, the line numbers will help to identify a line that you may wish to correct.

These instructions were produced using the program, on fanfold paper size 9.5" x 11".

To write a second page or more it is necessary to initialise the array again to clear the last page out or one can break into the program and start with G.

```

%...LETTER WRITING WITH P3.PRINTER....%
% PROGRAM BY D.WILLEY %
% 17.ST.MICHAEL AT PLEAS %
% NORMICH %
% NRS 1EP %

```

```
VAR A,B,N,Z: INTEGER;
```

```
CH: CHAR;
```

```
HEADMARGIN, LEFTHANDMARGIN, FOOTMARGIN, LINELENGTH, PAGEFULL, PAGENO: INTEGER;
```

```
CHNO, INDENT, LINENO: INTEGER;
```

```
LETTER: FILE OF ARRAY[64,80] OF INTEGER;
```

```
F1, F2, F3, F4, F5, F6: FILE OF INTEGER;
```

```
FINISH, ENDOFLETTER, FLAG, INITIALISED: BOOLEAN;
```

```
PROCEDURE TITLE;
```

```
BEGIN
```

```
  CLEAR;
```

```
  DOWN(10);
```

```
  LINE(40);
```

```
  WRITELN("
```

```
          PASCAL LETTER
```

```
");
```

```
  LINE(40);
```

```
  DELAY(3000);
```

```
END;
```

```
PROCEDURE INITIALISE;
```

```
  BEGIN -
```

```
    HEADMARGIN:=5;
```

```
    LEFTHANDMARGIN:=10;
```

```
    LINELENGTH:=55;
```

```
    FOOTMARGIN:=10;
```

```
    PAGEFULL:=64-FOOTMARGIN;
```

```
    CLEAR;
```

```
    DOWN(3); RIGHT(2);
```

```
    WRITELN("THE FOLLOWING IS THE INITIAL FORMAT");
```

```
    LINE(40);
```

```
    WRITELN("#####HEAD MARGIN = 5");
```

```
    WRITELN("#####LEFTHAND MARGIN = 10");
```

```
    WRITELN("#####FOOT MARGIN = 10");
```

```
    WRITELN("#####LINE LENGTH = 55");
```

```
    WRITELN("#####NUMBER OF LINES = 49");
```

```
    WRITELN();
```

```
    LINE(40);
```

```
    DOWN(4);
```

```
    INITIALISELETTER;
```

```
    INDENT:=0;
```

```
    LINENO:=1;
```

```
    ENDOFLETTER:=FALSE;
```

```
END;
```

```
PROCEDURE SETPAGEFORMAT;
```

```
  BEGIN
```

```
    CLEAR;
```

```
    REPEAT
```

```
      WRITELN("SET PAGE FORMAT");
```

```
      LINE(40);
```

```
      DOWN(3);
```

```
      WRITELN("HOW MANY LINES OF HEAD MARGIN");
```

```
      REPEAT
```

```
        READLN(HEADMARGIN);
```

```
      UNTIL(HEADMARGIN<=50)AND(HEADMARGIN>0);
```

```
      WRITELN("SET LEFT HAND MARGIN");
```

```
      REPEAT
```

```
        READLN(LEFTHANDMARGIN);
```

```
      UNTIL(LEFTHANDMARGIN<=51)AND(LEFTHANDMARGIN>0);
```

STANDARD LAYOUT

CHOOSE OWN LAYOUT

```

WRITELN("SET FOOT MARGIN");
REPEAT
  READLN(FOOTMARGIN);
  UNTIL (FOOTMARGIN<50) AND (FOOTMARGIN>0);
  IF (HEADMARGIN+FOOTMARGIN)>=63 THEN WRITELN("OUT OF RANGE RE-ENTER");
UNTIL (HEADMARGIN+FOOTMARGIN)<63;
REPEAT
  WRITELN("SET LINE LENGTH");
  WRITELN("MAX 80 INCLUDING LEFT HAND MARGIN");
  WRITE("MIN 30");LINE(10);
  READLN(LINELENGTH);
  IF (LINELENGTH<29) AND (LINELENGTH+LEFTHANDMARGIN>80) THEN
    WRITELN("OUT OF RANGE");
  UNTIL (LINELENGTH>29) AND (LEFTHANDMARGIN+LINELENGTH<80);
  WRITE("NUMBER OF USABLE LINES PER PAGE= ");
  PAGEFULL:=64-FOOTMARGIN;
  WRITELN(PAGEFULL-HEADMARGIN:3);
  DOWN<2>;
  INITIALISED:=TRUE;
  INITIALISELETTER;
END;
PROCEDURE PAGECOUNT;
BEGIN
  IF PAGENO>1 THEN BEGIN
    LETTER(1,40):=PAGENO+48;
    END;
END;
PROCEDURE DOWN(N:INTEGER);
BEGIN
  FOR A:=0 TO N DO WRITE("▣");
END;
PROCEDURE UP(N:INTEGER);
BEGIN
  FOR A:=0 TO N DO WRITE("▣");
END;
PROCEDURE RIGHT(N:INTEGER);
BEGIN
  FOR A:=0 TO N DO WRITE("▣");
END;
PROCEDURE LEFT(N:INTEGER);
BEGIN
  FOR A:=0 TO N DO WRITE("▣");
END;
PROCEDURE HOME;
BEGIN
  WRITE("▣");
END;
PROCEDURE CLEAR;
BEGIN
  WRITE("▣");
END;
PROCEDURE DELAY(N:INTEGER);
BEGIN
  FOR A:=1 TO N DO
END;
PROCEDURE LINE(N:INTEGER);
BEGIN
  FOR A:=1 TO N DO WRITE("-");
END;
%...WAIT FOR KEY PRESS....%

```

PUTS PAGE No. AT TOP OF PAGE

GRAPHICS 41D SECTION.

```

PROCEDURE GETKEY;
BEGIN
  CH:=KEY;
  WHILE ORD(CH)=0 DO BEGIN CH:=KEY;
  B:=ORD(CH);END;
END;

PROCEDURE INITIALISELETTER;
BEGIN
  INITIALISED:=TRUE;
  WRITELN("INITIALISING ARRAY PLEASE WAIT");
  FOR LINENO:=1 TO 64 DO BEGIN
    FOR CHNO:=1 TO 80 DO BEGIN
      LETTER[LINENO,CHNO]:=0;
    END;
  END;
END;

PROCEDURE GETLINE(LINENO,CHNO:INTEGER);
BEGIN
  WRITE(" ");
  WHILE FLAG=FALSE DO BEGIN
    CHNO:=LEFTHANDMARGIN+INDENT;
    N:=0;
    REPEAT
      CH:=KEY;
      IF ORD(CH)=96 THEN BEGIN
        WRITE("  ");
        CHNO:=CHNO-1;
        N:=N-1;
      END;
      B:=ORD(CH);
      IF B=99 THEN BEGIN
        POKE(CHR(1),4464);
        WRITE(" ");
      END;
      IF B=98 THEN BEGIN
        POKE(CHR(0),4464);
      END;
    UNTIL (B=102)OR((B>31)AND(B<=255))AND(B<>96)AND(B<>98)AND(B<>99);
    IF B<>102 THEN
      BEGIN
        LETTER[LINENO,CHNO]:=B;
        WRITE(CHR(LETTER[LINENO,CHNO]):1);
        CHNO:=CHNO+1;
        N:=N+1;
        IF CHNO=(LEFTHANDMARGIN+LINELENGTH-12) THEN BEGIN
          WRITE("XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX");
          CALL(62)Z,Z;
        END;
      END;
    WRITE(" ");
    IF CH=* THEN ENDOFLETTER:=TRUE;
    IF CH=# THEN POKE(CHR(0),4464);
  UNTIL (B=102)OR(CHNO=LEFTHANDMARGIN+LINELENGTH);
  B:=0;
  CHECK;
END;
FLAG:=FALSE;
END;
PROCEDURE LETTERWRITE:

```

CLEARs ARRAY
64 x 80 FILLING
EACH LOCATION
WITH 0

TYPE LINE IN.

DELETE KEY 96

SWITCH TO LOWER
CASE AND BACK
TO UPPER CASE

SEE LINE ON CRT

END OF LINE
MARKER

BACK TO UPPER CASE

```

BEGIN
  ENDOFLETTER:=FALSE;
  LINENO:=1;
  CLEAR;
  IF NOT INITIALIZED THEN INITIALISE;
  CLEAR;
  WRITELN("PAGE NUMBER 1 WILL START WITH ADDRESS ENTRY");
  WRITE("PAGE NUMBERS 2 OR MORE ARE BCONTINUATION ");
  WRITELN("SHEETS");
  LINE(40);
  WRITELN("ENTER PAGE NUMBER");
  READLN(PAGENO);
  PAGECOUNT;
  CLEAR;
  LINE(40);
  WRITELN("BEGIN ENTRY");
  LINE(40);
  WHILE (PAGENO=1) AND (LINENO<>PAGEFULL) AND (LINENO<HEADMARGIN+1) DO BEGIN
    CH:= ' ';
    WRITELN("ENTER FIRST LINE OF ADDRESS");
    LINENO:=HEADMARGIN+1;
    WRITELN("L.No ".LINENO:2);
    INDENT:=LEFTHANDMARGIN+LINELENGTH-36;
    LINENO:=LINENO+1;
    GETLINE(LINENO, INDENT+LEFTHANDMARGIN);
    WRITELN("L.No ".LINENO:2);
    WRITELN("2 ND LINE OF ADDRESS");
    INDENT:=LEFTHANDMARGIN+LINELENGTH-34;
    LINENO:=LINENO+1;
    GETLINE(LINENO, INDENT+LEFTHANDMARGIN);
    WRITELN("L.No ".LINENO:2);
    WRITELN("3 RD LINE OF ADDRESS");
    INDENT:=LEFTHANDMARGIN+LINELENGTH-32;
    LINENO:=LINENO+1;
    GETLINE(LINENO, INDENT+LEFTHANDMARGIN);
    WRITELN("L.No ".LINENO:2);
    WRITELN("4 TH AND LAST LINE OF ADDRESS");
    INDENT:=LEFTHANDMARGIN+LINELENGTH-30;
    LINENO:=LINENO+1;
    GETLINE(LINENO, INDENT+LEFTHANDMARGIN);
    WRITELN("L.No ".LINENO+1:2);
    WRITELN("DATE.");
    INDENT:=LEFTHANDMARGIN+LINELENGTH-28;
    LINENO:=LINENO+2;
    GETLINE(LINENO, INDENT+LEFTHANDMARGIN);
    WRITELN("L.No ".LINENO+1:2);
    WRITELN("WRITE DEAR...");
    INDENT:=0; IN:=0;
    LINENO:=LINENO+2;
    GETLINE(LINENO, LEFTHANDMARGIN);
    WRITELN();
    LINE(40);
    WRITELN(" + ENTERED AS OK WILL END THE PAGE");
    LINE(40);
    WRITELN("BCONTINUE WITH LETTER LINE", LINENO:2);
    INDENT:=IN;
    LINENO:=LINENO+1;
    GETLINE(LINENO, LEFTHANDMARGIN+INDENT);
  END;
  INDENT:=0;

```

ADDRESS
SECTION

```

IF PAGENO>1 THEN LINENO:=HEADMARGIN+1;
REPEAT
  CHNO:=LEFTHANDMARGIN;
  WRITELN(' ');
  WRITELN("L.No.",LINENO:3);
  LINENO:=LINENO+1;
  GETLINE(LINENO,CHNO);
UNTIL(LINENO)=PAGEFULL OR ENDOFLETTER=TRUE;
LINENO:=HEADMARGIN;
WRITELN("END OF PAGE REMEMBER TO SAVE ON FILES IF YOU WISH");
DELAY(4000);
POKE(CHR(0),4464);

```

} LETTER WRITING LOOP

```

END;
PROCEDURE CHECK;
BEGIN
  FLAG:=TRUE;
  WRITELN("OK? (Y/N)");
  REPEAT
    GETKEY;
  UNTIL(CH='Y')OR(CH='N')OR(CH='y')OR(CH='n');
  IF(CH='N')OR(CH='n')THEN BEGIN
    FLAG:=FALSE;
    FOR CHNO:=1 TO 80 DO
      LETTER(LINENO,CHNO):=0;
    WRITELN("RE-ENTER THE LINE");
  END;
  DOWN(6);UP(3);
  IF(PAGEFULL-LINENO)=3 THEN WRITE("THREE LINES BEFORE END OF PAGE");
  IF(PAGEFULL-LINENO)=3 THEN MUSIC("A7");

```

LIVE CHECK WITH OPTION TO RE-ENTER

```

END;
PROCEDURE VIEWPAGE;

```

SEE PAGE ON CRT.

```

BEGIN
  CLEAR;
  IF NOT INITIALISED THEN
  BEGIN
    WRITELN("INITIALISE OR LOAD A FILE FIRST");LINE(40);
    DELAY(3000);
  END;
  IF INITIALISED THEN BEGIN
    FOR LINENO:=1 TO 64 DO BEGIN
      WRITELN("L ",LINENO:2);
      FOR CHNO:=1 TO 80 DO BEGIN
        IF CHR(LETTER(LINENO,CHNO))='*' THEN BEGIN
          LINENO:=62;CHNO:=78;END;
          IF LETTER(LINENO,CHNO)=0 THEN WRITE(" ");
          WRITE(CHR(LETTER(LINENO,CHNO)):1);
          IF LINENO=66-FOOTMARGIN THEN
            BEGIN
              LINENO:=64;CHNO:=80;
            END;
        END;
      END;
    END;
  END;

```

OUT OF LOOP TO OBTAIN SPACES BETWEEN WORDS

OUT OF LOOP

```

END;
PROCEDURE PRINTPAGE;
BEGIN
  CLEAR;
  IF NOT INITIALISED THEN
  BEGIN
    WRITELN("INITIALISE OR LOAD A FILE FIRST");LINE(40);

```

```

        DELAY(3000);
END;
IF INITIALISED THEN BEGIN
    WRITELN("IS PRINTER ON AND THE PAPER IN
    GETKEY;
    IF CH='Y' THEN BEGIN
        IF INITIALISED THEN BEGIN
            PAGECOUNT;
            FOR A:=1 TO 64 DO BEGIN
                FOR N:=1 TO 80 DO BEGIN
                    IF LETTER(A,N)=0 THEN PWRITE(" ");
                    IF CHR(LETTER(A,N))='*' THEN BEGIN
                        A:=64;N:=80;
                    END;
                    IF A=65-FOOTMARGIN THEN BEGIN
                        A:=64;N:=80;
                    END;
                    PWRITE(CHR(LETTER(A,N)):1);
                END;
            END;
        END;
    END;
END;
END;
END;
END;
PROCEDURE LOADFILE;
BEGIN
    READ(LETTER[]);
    READ(F1,F2,F3,F4,F5,F6);
    HEADMARGIN:=F1;LEFTHANDMARGIN:=F2;FOOTMARGIN:=F3;LINELENGTH:=F4;
    PAGEFULL:=F5;PAGENO:=F6;
    INITIALISED:=TRUE;
    INDENT:=0;
END;
PROCEDURE FINIS;
BEGIN
    FINISH:=TRUE
END;
PROCEDURE TEST;% NOT IN USE %
BEGIN
    WRITELN("TEST AREA");
    WRITELN("CHNO ",CHNO:4);
    WRITELN("LINENO ",LINENO:4);
    WRITELN("INDENT ",INDENT:4);
    WRITELN("A,N,B",A:3,N:3,B:3);
    WRITELN("PAGEFULL ",PAGEFULL:3);
    WRITELN("PAGENO ",PAGENO:2);
    GETKEY;
END;
PROCEDURE SAVEPAGE;
BEGIN
    CLEAR;
    IF NOT INITIALISED THEN
        BEGIN
            WRITELN("INITIALISE OR LOAD A FILE FIRST");LINE(40);
            DELAY(3000);
        END;
    IF INITIALISED THEN BEGIN
        WRITE(LETTER[]);
        F1:=HEADMARGIN;F2:=LEFTHANDMARGIN;F3:=FOOTMARGIN;F4:=LINELENGTH;
        F5:=PAGEFULL;F6:=PAGENO;
    END;

```

POSITION?(Y/N)?:

PAGE NO. PLACED

OUT OF LOOP

PRINT LETTER

TEST AREA.

TO USE - INSERT

TEST; AT ANY
POINT IN THE PROGRAM
TO TRACE FAULTS.

SAVES ARRAY & FORM

```

WRITE(F1,F2,F3,F4,F5,F6)
END:
END:
PROCEDURE ALTERLINE:
BEGIN
  CLEAR:
  IF NOT INITIALISED THEN
  BEGIN
    WRITELN("INITIASE OR LOAD A FILE FIRST");
    DELAY(3000);
  END:
  IF INITIALISED THEN BEGIN
    WRITELN("ENTER THE NUMBER OF THE LINE ");
    REPEAT
      READLN(LINENO);
    UNTIL(LINENO>0)AND(LINENO<65);
    DOWN(3);
    FOR A:=1 TO 80 DO
      WRITE(CHR(LETTER(LINENO,A))):1);
    WRITELN():
    WRITELN("DO YOU WISH TO ALTER IT?");
    GETKEY:
    IF CH='Y' THEN BEGIN
      FOR CHNO:=1 TO 80 DO LETTER(LINENO,CHNO):=0:
      WRITELN("ENTER NEW LINE");
      WRITE(" ");
      GETLINE(LINENO,LEFTHANDMARGIN)
    END:
  END:
END:
%.....MAIN CONTROL%
BEGIN
  TITLE:
  INITIALISED:=FALSE:
  CLEAR:
  FINISH:=FALSE:
  REPEAT
    CLEAR:
    RIGHT(15):
    WRITELN(" MENU");
    DOWN(2);LINE(40):
    WRITELN("01 TO INITIALISE ARRAY");
    WRITELN("02 TO LOAD PAGE FROM FILE");
    WRITELN("03 TO SAVE PAGE ON FILE");
    WRITELN("04 TO SET PAGE FORMAT");
    WRITELN("05 TO WRITE LETTER");
    WRITELN("06 TO VIEW PAGE");
    WRITELN("07 TO PRINT THE PAGE");
    WRITELN("08 TO ALTER ONE LINE");
    WRITELN("09 TO FINISH");
    GETKEY:
    CASE CH OF
      '1':INITIALISE;
      '2':LOADFILE;
      '3':SAVEPAGE;
      '4':SETPAGEFORMAT;
      '5':LETTERWRITE;
      '6':VIEWPAGE;
      '7':PRINTPAGE;
      '8':ALTERLINE;
      '9':FINIS;
  END:
  UNTIL FINISH:
END.

```

FILL LINE WITH
BLANKS.

MAIN LOOP

DIMENSIONAL ANALYSIS

by DR. ALAN STEVENS

"I wrote this program and article a couple of years ago, but considered it too long and technical to be of interest to many SUN readers. However, Mr. Way's letter in SUN 14 has prompted me to send it. Like Mr. Way my main interest is in engineering which is where my MZ-80K finds most use.

Most of my serious programming now is done using (Hisoft) PASCAL. The Dimensional Analysis program was written before I knew about PASCAL and I can't be bothered to translate it!"

What do Concorde, the Dinorwic Pumping Station and the Humber Bridge all have in common? Indeed, what do they have in common with almost all major engineering undertakings and countless other minor ones? The answer is that their designers probably used scale models at some stage in their design.

Now the words 'scale model' immediately conjure up a vision of a small version of something big. But there is more to 'scale' than mere size. If we want to use a small scale model to investigate the effect 100 mph winds have upon a bridge there is no sense in subjecting the model to 100 mph winds! We must scale down the wind speed also. In general, we need to scale all those physical parameters which are relevant to the question of interest so that the model response is appropriate to its size yet similar to the response of the real thing in the real situation.

We use dimensional analysis in order to decide what parameters need to be scaled. We also use dimensional analysis in situations where we have made measurements on a large number of physical parameters and want to know how they all relate to each other. ('Large' often means 'more than 3!'). To determine the effectiveness of a refrigerator cooling system, for example, might require our making measurements on 7 or 8 different parameters. In order to assimilate the information provided by all the measurements we would group several of the parameters together and consider the relationships which exist among the reduced number of groups - hopefully an easier task.

Dimensional analysis tells us what are the sensible ways to group together the various parameters.

Physical parameters (like volume, weight, speed, density and so on) are all dimensional. That is, they can all be expressed in terms of a few primary dimensions like Mass (M), Length (L) and Time (T): Thus 'volume' has dimensions of L^3 (length cubed), 'speed' has dimensions of LT^{-1} (length per unit time) and so on. Our choice of what to nominate as a primary dimension is arbitrary as long as we can express all our physical parameters in terms of the set of primary dimensions we decide upon. for

present purposes we will adopt the conventional choices of Mass, Length, Time and Temperature (with apologies to electrical engineers for the absence of Charge or Current). Table 1 shows several examples of parameters and their characteristic dimensions.

The importance of dimensional analysis rests on the fact that if an equation exists, which relates all the physical parameters in a situation of interest, then that equation must be dimensionally 'homogeneous'. What this means is that if we were to go through the equation 'cancelling' dimensions in each term as though they were numbers, we would eventually find that every term in the equation could be reduced to an identical set of dimensions. If we were left with a term with a different set of dimensions the equation would not have correctly described the situation in the first place.

Let's illustrate this with an example taken from elementary kinematics. If an object travelling with speed 'u' accelerates at a uniform rate 'f' it will travel a distance 's' in a time 't'. The equation describing this motion is:

$$s=ut+(1/2)ft^2$$

Referring to table 1 we can see that the dimension of the first term in the above equation is simply L. The dimensions of the second term are (L/T)xT, which by 'cancelling' the T's we reduce to L. The third term has dimensions (L/T²)xT² which by 'cancelling' the T-squareds we again reduce to L. Thus each of the three terms in the equation is dimensionally the same (a length).

If we rearrange this equation we can alter the final dimension of each term. For example, if we divide through by 's' the equation becomes:

$$1=ut/s+(1/2)ft^2/s$$

The first term, the number one, is now dimensionless. The second term has dimensions (L/T)xTx(1/L), which, since all the L's and T's cancel out, is dimensionless. Similarly, for the third term (L/T²)xT²x(1/L) is dimensionless. So, although we have changed the form of the equation and the final dimension in each term (dimensionless now) we still find that, because the equations correctly describe the situation, all the terms have the same dimensionality. The equations are dimensionally 'homogeneous'.

It is this feature of homogeneity that allows us to perform a dimensional analysis, even when we do NOT know the equation involved. All we need to know are the relevant physical parameters. We simply need to know that we can group combinations of them together in such a way that every group has the same set of dimensions when all the 'cancelling' has taken place.

In particular, we can look for groups which are dimensionless, like ut/s and ft^2/s above. These groups are known as dimensionless ratios and are represented in general by the symbol π . If we have a set of physical parameters A, B, C etc, our method of performing a Dimensional Analysis is to assume that a dimensionless ratio π can be expressed as a product of powers a, b, c etc, of these parameters in the form:

$$\pi = A^a B^b C^c \dots$$

By finding values of the powers a, b, c etc which make the above produce dimensionless, we should be able to come up with a set of dimensionless ratios.

Let's demonstrate the method using our kinematics example. This time we pretend we do not know the equations given above, only that 's', 'u', 't' and 'f' are the relevant physical parameters. We begin by expressing π in terms of these parameters as follows:

$$\pi = s^a u^b t^c f^d$$

In terms of the primary dimensions the right-hand side of this equation is $(L)^a (LT^{-1})^b (T)^c (L^2 T^{-2})^d$ or $(L)^{a+b+d} (T)^{-b+c-2d}$. Now π is dimensionless, so in terms of the primary dimensions we can express it as $(L)^0 (T)^0$. To make this agree with our previous expression we must have:

$$a+b+d=0 \quad \text{from Length}$$

$$-b+c-2d=0 \quad \text{from Time}$$

We are free to choose any values for a, b, c , and d which satisfy the above two equations. For example, the values $a=-1$, $b=1$, $c=1$ and $d=0$ satisfy them. Putting these values back into our original equation for π gives us our first dimensionless ratio: $\pi = ut/s$. The values $a=-1$, $b=0$, $c=2$ and $d=1$ also satisfy the two equations, giving us our second dimensionless ratio: $\pi = ft^2/s$.

There are of course an infinite number of other choices we could make for a, b, c and d which would satisfy our two equations. We would find however, that if we did make a third choice, the resulting dimensionless ratio could also be obtained by combining the two we already have in some way. There are no more independent ratios - that is, ratios which cannot be expressed as a combination of the two we already have.

How can we tell, in general, if we have found the maximum number of independent, dimensionless ratios?

We do so using the Buckingham pi theorem, which says the number of independent, dimensionless ratios is equal to the total number of physical parameters involved minus the number of primary dimensions required to represent them. (Strictly, this is a slight oversimplification, but the complication need not concern

us here.) In our kinematics example there are 4 parameters, 's', 'u', 't' and 'f', represented by 2 primary dimensions, length and time. Hence there are $4-2=2$ independent, dimensionless ratios.

An important feature of dimensional analysis, illustrated by our example, is that although it produces sensible combinations of physical parameters in the right numbers, it says nothing about the precise relationships they bear to each other.

Dimensional analysis could not reproduce our original kinematics equation.

The very generality of the nature of dimensional analysis is both its strength and its weakness. We can never obtain a complete solution to any problem from dimensional reasoning alone. However, with only a little knowledge we can obtain a partial solution to almost any problem.

The Dimensional Analysis program produces a complete set of independent, dimensionless ratios for a particular situation, given as input the relevant physical parameters and their dimensional characteristics. I chose to restrict the number of primary dimensions involved to 4, simply because one very rarely needs more! For those who absolutely must have more the effort required in extending the program to cope should be minimal.

The principal problem to be overcome in producing a computerised method of dimensional analysis lies with the simultaneous equations that have to be solved. The problem is that, in general, these equations are 'underdetermined'. That is, they do not have a unique solution, as we saw in our example above. There is also a degree of arbitrariness in the values we assign to some of the powers that appear in the equations.

The program deals with this problem by first of all deciding how many dimensionless ratios there should be using Buckingham's pi theorem. If there are R such ratios it chooses R parameters (in the first instance these are just the first R it comes across) each of which is to appear in one and only one ratio.

For a particular ratio it assigns the value unity to the power of one of these R parameters, and zero to the powers of the other R-1. A different parameter is assigned the value of unity for a different ratio. This choice is sufficient to remove the 'underdetermined' aspect of the equations. The remaining simultaneous equations have either a unique solution or no solution at all. If a solution exists the program proceeds to calculate it using Cramer's rule. If there is no solution the program makes a different choice of R ratios and tries again.

The existence or otherwise of a solution is determined by calculating the determinant of the matrix of coefficients of the unknown powers. If the determinant is zero there is no solution. In practice, when the program selects the R parameters (lines 1060 to 1200) it applies a simple test which may eliminate some (but only some) of the combinations for which there is no solution.

The use of Cramer's rule to solve a set of simultaneous equations is unusual outside of a school mathematics lesson! The

rule is best illustrated by means of a simple example. Suppose we have the two simultaneous equations:

$$aX+bY=c$$

$$dX+eY=f$$

where 'a' to 'f' are known numbers and 'X' and 'Y' are the unknowns. Cramer's rule says that:

$$X=(ce-bf)/(ae-bd) \quad \text{and} \quad Y=(af-cd)/(ae-bd)$$

where the bracketed terms are various determinants obtained from the coefficients of the equations.

The trouble with Cramer's rule is that for a larger number of simultaneous equations the time taken to calculate the solutions can be very long, even for a computer. For present purposes, because we have no more than four primary dimensions, we have at most four simultaneous equations in any one calculation (we may often have only two or three). Four is a sufficiently small number that calculation time using Cramer's rule is not a problem. On the other hand, Cramer's rule is less likely to be affected by rounding errors than the faster techniques like Gaussian elimination or LU-decomposition.

Since, in dimensional analysis, the solutions are often small (+ve or -ve) integers, it is desirable that they appear as such, rather than as numerically very close non-integers.

The method the program uses to calculate determinants is worth commenting on. The value of a determinant can generally be expressed as a sum of terms, each of which is a number multiplied by a smaller determinant. This feature of determinants, that they are a recursive technique be used to calculate them. The problem with recursion in Basic is the absence of local variables. If simple variables are used in a recursive Basic routine their global nature means that values assigned at one depth of recursion destroy values required at another depth. The program overcomes this problem by using subscripted variables as 'pseudo-local' variables. Each time another depth of recursion is called for the subscript is incremented by one; on returning to a higher level the subscript is decremented again.

The program was written for the MZ-80K with a toolkit for the PRINT@s, but should translate with little difficulty to the 'A' and 'B'. 'K' owners without a toolkit may substitute for PRINT@X,Y;"text" directly by POKE4465,X : POKE4466,Y : PRINT"text". Alternatively, cursor control characters may be used.

Table 2 provides data for a test of the program using all four primary dimensions. Figure 1 shows the output which should be obtained with this data. The example comes from the study of convective heat transfer.

```

10 PRINT"@"
20 PRINT@10,5;"DIMENSIONAL ANALYSIS"
30 PRINTTAB(10);"....."
40 PRINT@7,12;"Do you want instructions ?"
50 GOSUB 2260:REM Get reply
60 IF K$="Y" THEN GOSUB 2420:REM Fetch instructions
70 REM
80 REM Data entry
90 REM
100 PRINT"@"
110 PRINT@0,10;"How many physical parameters are to be?"
120 INPUT"considered ? ";Q$
130 REM Test validity of input
140 GOSUB 2320:IF G=1 THEN 100
150 Q=VAL(Q$):IF Q<2 THEN 100
160 REM Input valid so proceed
170 DIM V(4),V$(4),PD$(4),NM$(Q),Q(4,Q)
180 PD$(1)="MASS.....":PD$(2)="LENGTH....."
190 PD$(3)="TIME.....":PD$(4)="TEMPERATURE.."
200 REM
210 REM J-loop gets data for each
220 REM parameter in turn
230 REM
240 FOR J=1 TO Q
250 PRINT"@"Enter name of parameter number";J;": "
260 INPUT NM$(J)
270 PRINT@0,5;"To what power should each of the"
280 PRINT"following primary dimensions be raised?"
290 PRINT"in ";NM$(J);" ?"
300 REM
310 REM I-loop gets data for each
320 REM primary dimension in turn
330 REM
340 FOR I=1 TO 4
350 PRINTTAB(10);" ";PD$(I);
360 INPUT Q$
370 REM Test validity of input
380 GOSUB 2320:IF G=1 THEN PRINT" ";SPC(38);" ":GOTO 350
390 REM Input valid so proceed
400 Q(I,J)=VAL(Q$)
410 S$=LEFT$(Q$+"0",2)
420 V$(I)=V$(I)+S$
430 V(I)=V(I)+ABS(SGN(VAL(Q$)))
440 REM V(I) counts how many times
450 REM primary dimension I occurs
460 NEXT I
470 PRINT"@" Are all the powers correct ?"
480 GOSUB 2260:REM Get reply
490 IF K$="Y" THEN 580
500 REM Strip off incorrect data
510 FOR I=1 TO 4
520 L=LEN(V$(I))
530 V=VAL(RIGHT$(V$(I),2))
540 IF V<>0 THEN V(I)=V(I)-1
550 V$(I)=LEFT$(V$(I),L-2)
560 NEXT I
570 PRINT"@" :GOTO 270
580 NEXT J
590 REM Data entry complete
600 REM
610 REM Now if any primary dimension
620 REM occurs only once, eliminate

```

```

630 REM appropriate parameter and
640 REM 'close up' the matrix
650 REM
660 PRINT"Q Data analysis indicates :Q"
670 FOR I=1 TO 4
680 IF V(I)=1 THEN GOSUB 1760
690 NEXT
700 IF (V(1)=1)+(V(2)=1)+(V(3)=1)+(V(4)=1) THEN 670
710 REM Test for independence
720 FOR I=1 TO 3
730 IF V(I)=0 THEN 770
740 FOR K=I+1 TO 4
750 IF V(I)=V(K) THEN 2850
760 NEXT K
770 NEXT I
780 REM Passed test so proceed
790 REM
800 PD=SGN(V(1))+SGN(V(2))+SGN(V(3))+SGN(V(4))
810 IF PD=0 THEN 2790
820 R=Q-PD
830 IF R<1 THEN 2800
840 C=Q:R=0
850 PRINT"QThere are";Q;" significant parametersQ"
860 PRINT"involving";PD;" primary dimensions.Q"
870 PRINT"Therefore, there are";R;" dimensionlessQ"
880 PRINT"ratios."
890 REM Close up rows of matrix
900 IF PD=4 THEN 970:REM No need!
910 K=0:FOR I=1 TO PD
920 K=K+1
930 IF V(K)=0 THEN 920
940 FOR J=1 TO Q
950 Q(I,J)=Q(K,J):V(I)=V(K)
960 NEXT J,I
970 REM
980 REM Set up arrays prior to
990 REM finding ratios
1000 DIM JN(R),J(PD),S(PD),U(PD),N$(PD)
1010 DIM D(PD,PD),DET(PD,PD)
1020 REM
1030 REM Find R independent parameters
1040 REM Label them with JN(L)
1050 REM
1060 FOR I=1 TO PD
1070 U(I)=V(I)
1080 NEXT
1090 J=0
1100 FOR L=1 TO R
1110 J=J+1
1120 TE=0
1130 FOR I=1 TO PD
1140 IF Q(I,J)<>0 THEN U(I)=U(I)-1
1150 IF U(I)>0 THEN 1170
1160 U(I)=1:TE=1:I=PD
1170 NEXT
1180 IF TE=1 THEN 1110
1190 JN(L)=J
1200 NEXT L
1210 REM
1220 REM Collect dependent parameters
1230 REM in matrix D(I,X) and put
1240 REM their names in N$(X)
1250 REM

```

```

1260 FOR I=1 TO PD
1270 X=0:FOR J=1 TO Q
1280 TE=0:FOR L=1 TO R
1290 IF J=JN(L) THEN TE=1:L=R
1300 NEXT
1310 IF TE=1 THEN 1330
1320 X=X+1:D(I,X)=Q(I,J):N$(X)=NM$(J)
1330 NEXTJ,I
1340 GOSUB 2020:REM Calc determinant
1350 C=C-1
1360 D=DET(0,0):IFD<>0THENB=1
1370 IF D=0 THEN 1550
1380 REM
1390 REM Calculate and display the
1400 REM dimensionless ratios
1410 REM
1420 PRINT"Dimensionless ratios are :";
1430 FOR L=1 TO R
1440 PRINT";L;". ";NM$(JN(L));" to the power... ";1
1450 FOR N=1 TO PD
1460 PRINTTAB(5);N$(N);" to the power... ";
1470 GOSUB 2220:GOSUB 2020
1480 PRINTDET(0,0)/D
1490 GOSUB2220
1500 NEXTN,L
1510 PRINT";That's all the dimensionless ratios."
1520 PRINT"Do you want an alternative set ?"
1530 GOSUB 2260
1540 IF K$="N" THEN 2940
1550 IF C>0 THEN 1600
1560 IF B>0 THEN 1590
1570 PRINT"The parameters are not all independent."
1580 GOTO 2900
1590 PRINT";That's all I can find.":GOTO2940
1600 PRINT";Please wait a moment."
1610 REM Cycle order of parameters
1620 REM in Q(I,J)
1630 FOR I=1 TO PD
1640 TM=Q(I,1)
1650 FOR J=1 TO Q-1
1660 Q(I,J)=Q(I,J+1)
1670 NEXT
1680 Q(I,Q)=TM
1690 NEXT
1700 TM$=NM$(1)
1710 FOR J=1TOQ-1
1720 NM$(J)=NM$(J+1)
1730 NEXT
1740 NM$(Q)=TM$
1750 GOTO 1260:REM New ratios
1760 REM Eliminate parameter and
1770 REM 'close up' columns
1780 FOR J=1 TO Q
1790 IF Q(I,J)<>0 THEN L=J:J=Q
1800 NEXT
1810 PRINT";NM$(L);" does not contribute."
1820 IF L=Q THEN 1860
1830 FOR J=L TO Q-1
1840 NM$(J)=NM$(J+1)
1850 NEXT
1860 FOR K=1 TO 4
1870 IF Q(K,L)<>0 THEN V(K)=V(K)-1
1880 IF L=1 THEN V$(K)=RIGHT$(V$(K),2*Q-2):GOTO 1910

```

MZ-80K NOTES & LISTINGS

```

1890 IF L=Q THEN V$(K)=LEFT$(V$(K),2*Q-2):GOTO 1940
1900 V$(K)=LEFT$(V$(K),2*L-2)+RIGHT$(V$(K),Q+Q-L-L)
1910 FOR J=L TO Q-1
1920 Q(K,J)=Q(K,J+1)
1930 NEXTJ
1940 NEXTK
1950 Q=Q-1
1960 RETURN
1970 REM
1980 REM Calculate determinant using
1990 REM Cramer's rule.
2000 REM The subroutine calls part of
2010 REM itself recursively.
2020 FOR J=1TOPD:DET(PD,J)=1:NEXT
2030 I=0:J(0)=0
2040 I=I+1:S(I)=1
2050 DET(I-1,J(I-1))=0:J(I)=0
2060 J(I)=J(I)+1:TE=0
2070 FORT=0TOI-1
2080 IFJ(I)=J(T)THENTE=1:T=I
2090 NEXT
2100 IFTE=1THEN2140
2110 IFI<PDTHENGOSUB2040
2120 DET(I-1,J(I-1))=DET(I-1,J(I-1))+D(I,J(I))*DET(I,J(I))*S(I)
2130 S(I)=-S(I)
2140 IFJ(I)<PDTHEN2060
2150 I=I-1:RETURN
2160 REM
2170 REM Put fresh column in matrix
2180 REM as per Cramer's rule
2190 REM before calculating det.
2200 REM Also put old column back
2210 REM afterwards.
2220 FOR K=1 TO PD
2230 TM=D(K,N):D(K,N)=-Q(K,JN(L)):Q(K,JN(L))=-TM
2240 NEXT
2250 RETURN
2260 REM Get yes/no reply
2270 PRINT@7,24;"Press Y for Yes, N for No?"
2280 K$=""
2290 GETK$
2300 IF (K$="Y")+ (K$="N") THENPRINTSPC(33);"":RETURN
2310 GOTO2290
2320 REM
2330 REM Test for valid input
2340 G=0
2350 FOR Z=1 TO LEN(Q$)
2360 Z$=MID$(Q$,Z,1)
2370 IF (ASC(Z$)>47)*(ASC(Z$)<58) THEN2400
2380 IF (Z=1)*(Z$="-")*(LEN(Q$)>1) THEN2400
2390 G=1
2400 NEXT
2410 RETURN
2420 REM      Instructions
2430 REM
2440 PRINT"ⓂThis program determines dimensionless"
2450 PRINT"ratios of physical parameters using"
2460 PRINT"Buckingham's pi theorem.Ⓜ"
2470 PRINT"You must enter the number of parameters"
2480 PRINT"you wish to be considered, followed"
2490 PRINT"by the pertinent information for each"
2500 PRINT"parameter in turn as requested.Ⓜ"
2510 PRINT"The information required for each"

```

```

2520 PRINT"parameter consists of it's name and"
2530 PRINT"the power to which each of the four"
2540 PRINT"primary dimensions, MASS, LENGTH, TIME"
2550 PRINT"and TEMPERATURE contribute to it's"
2560 PRINT"dimensionality."
2570 PRINT"The program then calculates how many"
2580 PRINT"dimensionless ratios there are, and"
2590 PRINT"for each one displays the names of those"
2600 PRINT"parameters and the powers to which they"
2610 PRINT"are raised in that ratio."
2620 PRINTTAB(8);"Press SPACE to continue."
2630 K$=""
2640 GET K$:IF K$=""THEN 2640
2650 PRINT"Although the number of dimensionless"
2660 PRINT"ratios for a particular problem is"
2670 PRINT"unique the arrangement of the physical"
2680 PRINT"parameters among those ratios need"
2690 PRINT"not be."
2700 PRINT"Possible alternatives will be calculated"
2710 PRINT"by the program if you wish. (The program"
2720 PRINT"will ask about this after calculating"
2730 PRINT"and displaying one set of ratios.)"
2740 PRINTTAB(8);"Press SPACE to begin."
2750 K$=""
2760 GET K$:IF K$=""THEN 2760
2770 RETURN
2780 REM
2790 PRINT"There are no primary dimensions left":GOTO2820
2800 PRINT"There are no more parameters than"
2810 PRINT"there are primary dimensions."
2820 PRINT"No dimensionless ratios can be formed."
2830 PRINT"Try using more parameters next time."
2840 GOTO2940
2850 PRINT"The primary dimensions : "
2860 PRINT"...";PD$(I);PD$(K);" "
2870 PRINT"bear the same relationship to each"
2880 PRINT"other in all parameters in which"
2890 PRINT"they occur, and so are not independent."
2900 PRINT"Thus Buckingham's theorem is not"
2910 PRINT"applicable. Examine your parameters"
2920 PRINT"carefully and try to remove this"
2930 PRINT"feature."
2940 END

```

Table 1.

Dimensions of some physical parameters

| Parameter | <u>Power to which primary dimensions are raised</u> | | | | |
|----------------------|---|----|----|----|----------------|
| | M | L | T | TP | |
| Distance | 0 | 1 | 0 | 0 | M=Mass |
| Speed | 0 | 1 | -1 | 0 | L=Length |
| Time | 0 | 0 | 1 | 0 | T=Time |
| Acceleration | 0 | 1 | -2 | 0 | TP=Temperature |
| Density | 1 | -3 | 0 | 0 | |
| Force | 1 | 1 | -2 | 0 | |
| Thermal conductivity | 1 | 1 | -3 | -1 | |
| Specific heat | 0 | 2 | -2 | -1 | |
| Stress | 1 | -1 | -2 | 0 | |
| Viscosity | 1 | -1 | -1 | 0 | |

Table 2.

Input data for test problem

| Parameter name: | h | D | k | rho | v | mu | cp |
|-----------------|----|---|----|-----|----|----|----|
| MASS | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| LENGTH | 0 | 1 | 1 | -3 | 1 | -1 | 2 |
| TIME | -3 | 0 | -3 | 0 | -1 | -1 | -2 |
| TEMPERATURE | -1 | 0 | -1 | 0 | 0 | 0 | -1 |

INTERPRETING PASCAL'S PROGRAMMING LANGUAGE

and this book will help you to understand the language better. It is a good idea to read this book before you start programming in Pascal. It will help you to understand the language better and to avoid many of the common mistakes that beginners make.

Data analysis indicates :

There are 7 significant parameters involving 4 primary dimensions. Therefore, there are 3 dimensionless ratios.

Dimensionless ratios are :

- 1. h to the power... 1
- rho to the power... -1
- v to the power... -1
- mu to the power... 0
- cp to the power... -1
- 2. D to the power... 1
- rho to the power... 1
- v to the power... 1
- mu to the power... -1
- cp to the power... 0
- 3. k to the power... 1
- rho to the power... 0
- v to the power... 0
- mu to the power... -1
- cp to the power... -1

That's all the dimensionless ratios.

Figure 1.

Figure 1 shows the dimensionless ratios derived from the analysis of the data. The ratios are listed in three groups, numbered 1, 2, and 3. Each group contains a set of ratios that are dimensionless. The ratios are listed with their corresponding exponents.

INTERRUPTING, Pardon? INTERRUPTING THE SHARP MZ-80K

by

William Howard

Introduction

This article deals with interrupts, what they are, how to generate them on the Sharp MZ-80K and, briefly, what they can be used for. All Programs and program segments contained within this article are copyright of Mr. W. Howard although the code may be copied for private use by the owner of this article. Also no responsibility is accepted for any damage to the computer or corruption of files that may occur through the use of any information contained within this article.

What is an Interrupt?

An interrupt is an externally (to the Z-80 CPU) generated signal to tell the processor to temporarily suspend the current operation and to transfer control to another program.

e.g. The Pardon? in the title of this article can be considered an interrupt as it temporarily suspended the title and asked for the previous word to be repeated, after which control was returned to the title.

If you're still confused either look up interrupt in a computing book or read the rest of this article as it is far easier to explain how to generate and use interrupts than it is to define them!

What can we do with Interrupts?

There are two types of interrupts, ones which come at regular intervals and those which come at the occurrence of some event. Regular occurring interrupts are perhaps the most useful as they allow us to do an event at regular intervals independent of what other event is currently running. One-off interrupts are useful in that they can signify the end of an event e.g. we could set a time up to time a 3 minute interval, say to cook an egg, and get the timer to generate an interrupt at the end of the 3 minutes so that we could then remove the egg. This article deals mainly with regular interrupts as they are by far the most useful for the hobbyist.

How do we generate Interrupts?

There are two ways, either to take the $\overline{\text{NMI}}$ or the $\overline{\text{INT}}$ line low, simple!

Unfortunately it is not as simple as this. Although we could build some circuits to plug into the computer it is certainly no easy task and I would neither risk it or write an article on it!

Fortunately for us there is an IC within the Sharp which has one of its outputs connected to the INT line of the Z-80 CPU. Hence if we can reprogram this chip we can easily generate interrupts.

What is the difference between the NMI and INT lines? The NMI (from now on I shall refer to this simply as the NMI as overlining is awkward!) line is the Non-Maskable Interrupt line, the overlining is used to show that it is active low i.e. an interrupt is generated if a logic 0 is applied to it. A Non-Maskable Interrupt is one which the Z-80 can't ignore and therefore is used for special purposes within the system and is therefore little interest to us. The INT line (also active low) is the normal interrupt line and is therefore of interest to us, especially as of the aforementioned IC's output is connected to it, although not directly.

The Z-80 CPU has two instructions EI and DI, DI tells the CPU to ignore all interrupts generated by the INT line and EI tells it to acknowledge interrupts from the INT line.

Z-80 Interrupt Modes

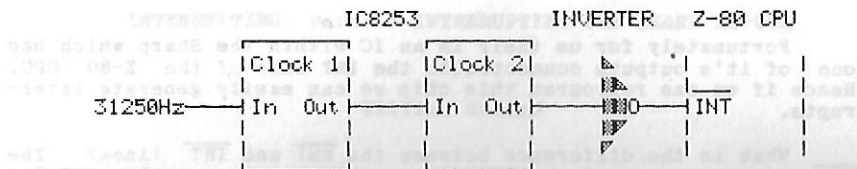
The Z-80 CPU has three modes for use with INT generated interrupts. These are mode, 0, 1 & 2. It is beyond the scope of this article to explain why three modes are needed or to explain how they work. It is sufficient to say that we shall be using mode 1.

When an interrupt is generated in this mode the CPU effectively executes a RST 38H instruction i.e. the Program Counter is stacked and reloaded with 0038H thus causing execution to be transferred to address 0038H. As this is in monitor it may seem a bit stupid but as there is a JP 1038H instruction at this address control is effectively passed to location 1038H which is above the Monitor and in RAM. Unfortunately there is not enough spare memory here (we are in the Monitor's workspace after all) to write any useful code so we invariably have to load another Jump instruction here to jump to our interrupt routine in User RAM. Returning control to the main program again is extremely simple though as the PC was stacked before we jumped to address 0038H all we need to do is execute a RET instruction!

Back to Generating Interrupts

We now (hopefully) know what an interrupt is and how to generally generate one and what happens (in Mode 1) after we do. However we have not yet found out how to generate one on the Sharp (other than saying that we reprogram the on-board IC)

The IC we are going to reprogram is IC8253. This IC has three on-chip clocks, a 'clock' in this case is a counter which is externally driven and can count either up or down from preset values and do clever things on a count of 0! Although there are three clocks on the chip 0, 1 & 2 we are only interested in clocks 1 & 2. These clocks are configured as shown below:-



What these clocks currently do is to supply the internal clock, BASIC's TI\$. This means that if we reprogram these clocks then we will not be able to use the internal clock, or in general we must not call the Monitor subroutines TIMEST (0033H) or TIMERD (003BH).

How do we reprogram these clocks? Again this is outside of the scope of this article to cover this in detail so only the bare necessities will be given. Both clocks count pulses, clock one counts the pulses from the 31250Hz external clock (looks like a strange number but multiply it by 2 six times, makes sense now?) while clock 2 counts pulses from clock 1.

IC8253 has 6 modes that the clocks can be programmed in, mode 2 is probably the most useful as it effectively divides the input signal by N, where N is the number we load into the clock to start with. Hence if we program clock 1 in mode 2 then the output signal, and hence the input signal to clock 2, is at a frequency of 31250/N Hz. For most of our applications N=7DH (125 decimal) is a good choice for N as this gives an input to clock 2 of 250Hz.

Now if we also program clock 2 in mode 2 with an initial count of M then we should get interrupts at a frequency of 250/M HZ or 250/M interrupts a second. Unfortunately life isn't that simple (due to that inverter) and if we do program clock 2 in mode 2 then we get nothing but interrupts except for the occasional return to the main program once every 250/M th of a second! Fortunately there is another mode, mode 0 and although it is not as useful it can be used. The main disadvantage is that after each interrupt we need to reprogram the clock again. However an advantage is that if we disable the interrupts with a DI instruction and then get an interrupt, when we enable the interrupts with an EI instruction we will still get the interrupt i.e. we don't lose it.

Well so much for the theory, the following code programs the clocks and gives two test programs. USER1 is an infinite loop and you will either have to reset or switch off the machine to escape, so save the program to take before running it!! USER2 is identical except that it initiates the interrupts first, hence all the 0's. You will also need to specify values for PRESCL and CLK2CT for the test programs these should be 007DH and 0019H respectively.

Try changing line 24 page 2 to B4H and deleting line 3 page 4 and running USER2 now, what you have done is to reprogram clock 2 to mode 2, see above.

** Z80 ASSEMBLER SP-2102 PAGE 01 **

```

01 0000 ;
02 0000 ;IC8253 CLOCK SUBROUTINES.
03 0000 ;=====
04 0000 ;
05 0000 P CTLWRD: EQU E007H ;CONTROL WORD
06 0000 P CLKPT1: EQU E005H ;CLOCK PORT 1
07 0000 P CLKPT2: EQU E006H ;CLOCK PORT 2
08 0000 ;
09 0000 PRESCL: ENT
10 0000 0000 DEFW 0000H ;VALUE OF THE PRESCALAR
11 0002 CLK2CT: ENT
12 0002 0000 DEFW 0000H ;COUNT BEFORE INTERRUPT
13 0004 ;
14 0004 ;SUBROUTINE TO SET CLOCK 1, THE PRESCALAR.
15 0004 ;-----
16 0004 ;
17 0004 ;FUNCTION = SETS CLOCK 1 TO DIVIDE THE INPUT FREQUENCY
18 0004 ;OF 31250Hz BY THE VALUE IN PRESCL AND IT IS THIS
19 0004 ;FREQUENCY WHICH DRIVES CLOCK 2.
20 0004 ;
21 0004 ;REGISTERS= ALL SAVED.
22 0004 ;
23 0004 ;TIME = 52.5 uSEC.
24 0004 ;
25 0004 CLOCK1: ENT ;EXTERNAL ENTRY POINT
26 0004 F5 PUSH AF ;SAVE REGISTERS
27 0005 ;
28 0005 ;OUTPUT 74H (01110100 BINARY) TO CLOCK CONTROL WORD TO
29 0005 ;SPECIFY THAT WE WANT TO PROGRAM CLOCK 1 IN MODE 2 AND
30 0005 ;THAT WE ARE GOING TO LOAD THE COUNTER WITH A TWO BYTE
31 0005 ;NUMBER.
32 0005 ;
33 0005 3E74 LD A,74H
34 0007 3207E0 LD (CTLWRD),A
35 000A ;
36 000A ;OUTPUT PRESCALAR TO CLOCK 1 NOW.
37 000A ;
38 000A 3A0000 LD A,(PRESCL)
39 000D 3205E0 LD (CLKPT1),A ;LSB FIRST
40 0010 3A0100 LD A,(PRESCL+1)
41 0013 3205E0 LD (CLKPT1),A ;MSB SECOND
42 0016 F1 POP AF ;RESTORE REGISTERS
43 0017 C9 RET ;EXIT
44 0018 SKP H

```

```

01 0016 ;
02 0018 ;SUBROUTINE TO SET CLOCK 2, THE INTERRUPT GENERATOR.
03 0018 ;-----
04 0018 ;
05 0018 ;FUNCTION = SETS CLOCK 2 TO PRODUCE AN INTERRUPT ONCE
06 0018 ;THE COUNTER REACHES ZERO. THE COUNTER IS DECREMENTED
07 0018 ;ONCE EVERY 31250/PRESCL th OF A SECOND.
08 0018 ;
09 0018 ;REGISTERS= ALL SAVED.
10 0018 ;
11 0018 ;TIME = 52.5 uSEC.
12 0018 ;
13 0018 CLOCK2: ENT ;EXTERNAL ENTRY POINT
14 0018 F5 PUSH AF ;SAVE REGISTERS
15 0019 ;
16 0019 ;OUTPUT B0H (10110000 BINARY) TO CLOCK CONTROL WORD TO
17 0019 ;SPECIFY THAT WE WANT TO PROGRAM CLOCK 2 IN MODE 0 AND
18 0019 ;THAT WE ARE GOING TO LOAD THE COUNTER WITH A TWO BYTE
19 0019 ;NUMBER.
20 0019 ;
21 0019 3EB0 LD A,B0H
22 001B 3207E0 LD (CTLWRD),A
23 001E ;
24 001E ;OUTPUT COUNT TO CLOCK 2 NOW.
25 001E ;
26 001E 3A0200 LD A,(CLK2CT)
27 0021 3206E0 LD (CLKPT2),A ;LSB FIRST
28 0024 3A0300 LD A,(CLK2CT+1)
29 0027 3206E0 LD (CLKPT2),A ;MSB SECOND
30 002A F1 POP AF ;RESTORE REGISTERS
31 002B C9 RET ;EXIT
32 002C SKP H
    
```

```

01 002C ;
02 002C ;SUBROUTINE TO REDIRECT INTERRUPTS.
03 002C ;-----
04 002C ;
05 002C ;FUNCTION = BY LOADING 1038H WITH C3H AND 1039H & 103AH
06 002C ;WITH AN ADDRESS ALL INTERRUPTS GENERATED BY THE CLOCK
07 002C ;END UP AT THE ADDRESS SPECIFIED.
08 002C ;
09 002C ;REGISTERS= ALL SAVED.
10 002C ;
11 002C ;TIME = 51.0 uSEC.
12 002C ;
13 002C VECTOR: ENT ;EXTERNAL ENTRY POINT
14 002C F5 PUSH AF
15 002D E5 PUSH HL ;SAVE REGISTERS
16 002E 3EC3 LD A,C3H ;JUMP INSTRUCTION
17 0030 323810 LD (<1038H>),A ;INTERRUPTS ARRIVE HERE
18 0033 210000 LD HL,INT ;INTERRUPT ROUTINE START
19 0036 223910 LD (<1039H>),HL ;LOAD JUMP ADDRESS
20 0039 E1 POP HL
21 003A F1 POP AF ;RESTORE REGISTERS
22 003B C9 RET ;EXIT
23 003C SKP 5
    
```

```

29 003C      ;
30 003C      ;SUBROUTINE TO INITIALISE THE FIRST INTERRUPT.
31 003C      ;
32 003C      ;
33 003C      ;FUNCTION = SETS THE CORRECT INTERRUPT MODE, SETS UP
34 003C      ;CLOCK 1, SETS UP THE INTERRUPT VECTOR AND STARTS THE
35 003C      ;INTERRUPT SEQUENCE BY ENABLING THE INTERRUPT LINE AND
36 003C      ;BY SETTING CLOCK 2.
37 003C      ;
38 003C      ;REGISTERS= ALL SAVED.
39 003C      ;
40 003C      ;TIME      = 192.5 uSEC.
41 003C      ;
42 003C      IIRUPT: ENT                ;EXTERNAL ENTRY POINT
43 003C      IN      1                ;SET INTERRUPT MODE 1
44 003E      CD0400      CALL  CLOCK1
45 0041      CD2C00      CALL  VECTOR
46 0044      CD1800      CALL  CLOCK2
47 0047      FB                    ;ENABLE INTERRUPTS
48 0048      C9                    ;EXIT
49 0049      END

```

```

CLK2CT  0002  CLKPT1  E005  CLKPT2  E006  CLOCK1  0004  CLOCK2  0018
CTLWRD  E007  IIRUPT  003C  PRESCL  0000  VECTOR  002C

```

```

01 0000      ;
02 0000      ;TEST PROGRAMS FOR IC8253 CLOCK SUBROUTINES.
03 0000      ;
04 0000      ;
05 0000      ;SUBROUTINE DEPENDANCE :- IC8253 CLOCK SUBROUTINES.
06 0000      ;
07 0000      ;
08 0000      ;
09 0000      ;INTERRUPT SUBROUTINE.
10 0000      ;
11 0000      ;
12 0000      ;FUNCTION = EVERY TIME AN INTERRUPT IS GENERATED IT
13 0000      ;ARRIVES HERE. THIS INTERRUPT SUBROUTINE PRINTS
14 0000      ;A '0' EVERY TIME IT IS CALLED.
15 0000      ;
16 0000      ;REGISTERS= ALL SAVED.
17 0000      ;
18 0000      ;TIME      = 93.5 + TIME FOR MONITOR ROUTINE PRNT.
19 0000      ;
20 0000      INT:      ENT                ;INTERRUPT ROUTINE ENTRY
21 0000      F3                    ;INTERRUPT THIS ROUTINE?
22 0001      CD0000      E      CALL  CLOCK2      ;RESTART INTERRUPTS
23 0004      F5                    ;
24 0005      3E30      LD      A,"0"
25 0007      CD1200      CALL  0012H      ;MONITOR ROUTINE PRNT
26 000A      F1                    ;
27 000B      FB                    ;
28 000C      C9                    ;EXIT
29 000D      SKP  H                ;

```

```

01 0000 ;
02 0000 ;TEST PROGRAMS.
03 0000 ;-----
04 0000 ;
05 0000 ;FUNCTION = USER1 IS AN INFINITE LOOP THAT PRINTS
06 0000 ;'1' AND AS SUCH THE MACHINE WILL HAVE TO BE RESET
07 0000 ;OR SWITCHED OFF TO ESCAPE FROM IT.
08 0000 ;USER2 IS SIMILIAR EXCEPT THAT IT STARTS THE
09 0000 ;INTERRUPTS GOING BEFORE ENTERING THE LOOP.
10 0000 ;
11 0000 ;REGISTERS= NOT CRITICAL.
12 0000 ;
13 0000 ;TIME = NOT CRITICAL.
14 0000 ;
15 0000 USER2: ENT ;TEST PROGRAM 2
16 0000 CD0000 E CALL IIRUPT ;START INTERRUPTS
17 0010 SKP 5

```

```

23 0010 USER1: ENT ;TEST PROGRAM 1
24 0010 3E31 LD A,"1"
25 0012 CD1200 LOOP: CALL 0012H ;MONITOR ROUTINE PRINT
26 0015 18FB JR LOOP
27 0017 END

```

INT 0000 LOOP 0012 USER1 0010 USER2 0000

>M 2000 205F

```

2000 F3 CD 2F 20 F5 3E 30 CD 12 00 F1 FB C9 CD 53 20
2010 [3E] 31 CD 12 00 18 FB 7D 00 19 00 F5 3E 74 32 07
2020 E0 3A 17 20 32 05 E0 3A 18 20 32 05 E0 F1 C9 F5
2030 3E 04 32 07 E0 3A 19 20 32 06 E0 3A 1A 20 32 06
2040 E0 F1 C9 F5 E5 3E C3 32 38 10 21 00 20 22 39 10
2050 E1 F1 C9 ED 56 CD 1B 20 CD 43 20 CD 2F 20 FB C9

```

8253 TEST1.ABS

>M 2000 205F

```

2000 F3 CD 00 00 F5 3E 30 CD 12 00 F1 FB C9 CD 53 20
2010 [3E] 31 CD 12 00 18 FB 7D 00 19 00 F5 3E 74 32 07
2020 E0 3A 17 20 32 05 E0 3A 18 20 32 05 E0 F1 C9 F5
2030 3E 04 32 07 E0 3A 19 20 32 06 E0 3A 1A 20 32 06
2040 E0 F1 C9 F5 E5 3E C3 32 38 10 21 00 20 22 39 10
2050 E1 F1 C9 ED 56 CD 1B 20 CD 43 20 CD 2F 20 FB C9

```

8253 TEST2.ABS

() Differences between Test1 & Test2

[] Entry Points 2000 & 2010 for User2 & User1 respectively

/ / Data words PRESCL & CLK2CT respectively

A Smattering of Mathematics

PRESCL is the program name for what I called N above and CLK2CT is it's name for M. Changing the values of these two variables changes the frequency of the interrupts as dictated by the formula:-

$$\text{Frequency} = \frac{31250}{\text{PRESCL} * \text{CLK2CT}} \text{ Hz.}$$

Some common frequencies are:-

| <u>Frequency</u> | <u>PRESCL</u> | <u>CLK2CT</u> |
|------------------|---------------|---------------|
| 1 Hz | 125 | 250 |
| 2 Hz | 125 | 125 |
| 10 Hz | 125 | 25 |
| 25 Hz | 125 | 10 |
| 50 Hz | 125 | 5 |
| 50 Hz | 25 | 25 |
| 125 Hz | 25 | 10 |
| 250 Hz | 25 | 5 |
| 250 Hz | 5 | 25 |

Unfortunately we can't get 100 Hz out, which would be nice, the best we can do is with PRESCL = 2H and CLK2CT = 009CH which gives a frequency of 100.2 Hz which for most applications is probably accurate enough.

Users of Interrupts

We now have a set of routines that we can generate interrupts at almost any desired frequency, so what can we do with them, other than price 0's and 1's? One obvious use is to program a real time clock. The following code does just this. A CALL to START starts the clock while a CALL to STOP stops it. The data it works on is held in Binary Coded Decimal (BCD) format in locations C800H to C802H and some other code will need to be written to initialise these. However this code does not need to be in the interrupt routine. This code could easily be altered to print out, on the screen, the time a real time game has been running for. (See REAL-TIME CLOCK PROGRAM on next page).

This program is not very time efficient as it prints the entire clock to the screen every time there is an interrupt. The program would be better if it only updated those digits which changed, but more about efficiency later.

It was at this stage that I discovered a very interesting and annoying fact. If the above program is run with PRESCL=0002H and CLK2CT=3D04H a more accurate result is obtained than if 007DH and 00FAH are used!!! After much experimenting it was found that the higher the number in PRESCL the greater is the error. This means that if accurate results are needed PRESCL should be small, but not equal to one! This is rather annoying but only really important if we need accurate frequencies, if we don't we can still use the above table and formula.

** Z80 ASSEMBLER SP-2102 PAGE 01 **

```

01 0000      ;
02 0000      ;A REAL-TIME CLOCK.
03 0000      ;=====
04 0000      ;
05 0000      ;SUBROUTINE DEPENDANCE :- IC8253 CLOCK SUBROUTINES.
06 0000      ;
07 0000 P    SCREEN: EQU    D011H      ;MIDDLE OF TOP ROW
08 0000      ;
09 0000      ;INTERRUPT SUBROUTINE.
10 0000      ;-----
11 0000      ;
12 0000      ;FUNCTION = INTERRUPTS ARRIVE HERE AT THE RATE OF
13 0000      ;ONE A SECOND. THIS SUBROUTINE UPDATES THE REAL-
14 0000      ;TIME CLOCK AND OUTPUTS IT TO THE SCREEN.
15 0000      ;
16 0000      ;REGISTERS= ALL SAVED.
17 0000      ;
18 0000      ;TIME      = 314.5 / 340.5 / 366.5 / 367.5 uSEC.
19 0000      ;
20 0000      INT:      ENT          ;INTERRUPT ROUTINE ENTRY
21 0000 F3     DI          ;INTERRUPT THIS ROUTINE?
22 0001 CD0000 E    CALL   CLOCK2    ;RESTART INTERRUPTS
23 0004 F5     PUSH  AF          ;SAVE REGISTERS
24 0005 E5     PUSH  HL          ;
25 0006 219700 LD    HL,SECS    ;POINT TO SECONDS STORE
26 0009 7E     LD    A,(HL)
27 000A C601   ADD    A,01H      ;ADD 1 TO SECONDS
28 000C 27     DAA
29 000D FE60   CP     60H        ;ONE MINUTE ?
30 000F 201A   JR    NZ,EXIT    ;NO SO LEAVE
31 0011 3E00   LD    A,00H        ;YES SO ZERO SECONDS
32 0013 77     LD    (HL),A    ;RESTORE SECONDS
33 0014 23     INC   HL         ;POINT AT MINUTES
34 0015 7E     LD    A,(HL)
35 0016 C601   ADD    A,01H      ;ADD 1 TO MINUTES
36 0018 27     DAA
37 0019 FE60   CP     60H        ;ONE HOUR ?
38 001B 200E   JR    NZ,EXIT    ;YES SO LEAVE
39 001D 3E00   LD    A,00H        ;NO SO ZERO MINUTES
40 001F 77     LD    (HL),A    ;RESTORE MINUTES
41 0020 23     INC   HL         ;POINT AT HOURS
42 0021 7E     LD    A,(HL)
43 0022 C601   ADD    A,01H      ;ADD 1 TO HOURS
44 0024 27     DAA
45 0025 FE24   CP     24H        ;1 DAY ?
46 0027 2002   JR    NZ,EXIT    ;YES SO LEAVE
47 0029 3E00   LD    A,00        ;NO SO ZERO HOURS
48 002B 77     LD    (HL),A    ;RESTORE VALUE
49 002C      EXIT:      LD    SKP   H

```

MZ-80K NOTES & LISTINGS

** Z80 ASSEMBLER SP-2102 PAGE 02 **

```

01 002C          ;
02 002C          ;PRINT NEW TIME ON SCREEN
03 002C          ;
04 002C D5      PUSH DE          ;SAVE REGISTER
05 002D 1111D0  LD DE,SCREEN      ;CLOCK AT TOP LEFT
06 0030 219900  LD HL,HOURS       ;PRINT HOURS FIRST
07 0033 7E      LD A,(HL)     ;GET HOURS IN BCD
08 0034 E6F0    AND F0H       ;MASK OUT LSB
09 0036 CB3F    SRL A          ;
10 0038 CB3F    SRL A          ;
11 003A CB3F    SRL A          ;
12 003C CB3F    SRL A          ;LSB TO MSB
13 003E C620    ADD A,20H      ;DISPLAY CODE OFFSET
14 0040 12      LD (DE),A     ;OUTPUT TO SCREEN
15 0041 13      INC DE        ;POINT TO NEXT PLACE
16 0042 7E      LD A,(HL)     ;GET HOURS BACK
17 0043 E60F    AND 0FH       ;MASK OUT MSB
18 0045 C620    ADD A,20H      ;DISPLAY CODE OFFSET
19 0047 12      LD (DE),A     ;OUTPUT TO SCREEN
20 0048 13      INC DE        ;POINT TO NEXT PLACE
21 0049 3E4F    LD A,4FH     ;DISPLAY CODE FOR :
22 004B 12      LD (DE),A     ;OUTPUT TO SCREEN
23 004C 13      INC DE        ;POINT TO NEXT PLACE
24 004D 2B      DEC HL        ;POINT TO MINUTES
25 004E          SKP 5

31 004E 7E      LD A,(HL)     ;REPEAT FOR MINUTES
32 004F E6F0    AND F0H
33 0051 CB3F    SRL A
34 0053 CB3F    SRL A
35 0055 CB3F    SRL A
36 0057 CB3F    SRL A
37 0059 C620    ADD A,20H
38 005B 12      LD (DE),A
39 005C 13      INC DE
40 005D 7E      LD A,(HL)
41 005E E60F    AND 0FH
42 0060 C620    ADD A,20H
43 0062 12      LD (DE),A
44 0063 13      INC DE
45 0064 3E4F    LD A,4FH
46 0066 12      LD (DE),A
47 0067 13      INC DE
48 0068 2B      DEC HL        ;POINT TO SECONDS
49 0069          SKP H

```

** Z80 ASSEMBLER SP-2102 PAGE 03 **

```

01 0069 7E                LD    A,(HL),          ;REPEAT FOR SECONDS
02 006A E6F0             AND    F0H
03 006C CB3F             SRL    A
04 006E CB3F             SRL    A
05 0070 CB3F             SRL    A
06 0072 CB3F             SRL    A
07 0074 C620             ADD    A,20H
08 0076 12              LD    (DE),A
09 0077 13              INC    DE
10 0078 7E              LD    A,(HL)
11 0079 E60F             AND    0FH
12 007B C620             ADD    A,20H
13 007D 12              LD    (DE),A
14 007E D1              POP    DE              ;RESTORE REGISTERS
15 007F E1              POP    HL
16 0080 F1              POP    AF
17 0081 FB              EI                      ;ENABLE INTERRUPTS
18 0082 C9              RET                      ;EXIT
19 0083                SKP    10

30 0083                ;
31 0083                ;SUBROUTINE TO START THE CLOCK.
32 0083                ;-----
33 0083                ;
34 0083                ;FUNCTION = SETS THE CORRECT VALUES IN PRESCL AND
35 0083                ;CLK2CT AND STARTS THE CLOCK BY CALLING IIRUPT.
36 0083                ;
37 0083                ;REGISTERS= ALL SAVED.
38 0083                ;
39 0083                ;TIME      = NON CRITICAL.
40 0083                ;
41 0083                START: ENT    ;EXTERNAL ENTRY POINT
42 0083 E5                PUSH   HL              ;SAVE REGISTERS
43 0084 217D00           LD    HL,007DH
44 0087 220000           LD    (PRESCL),HL    ;125 DECIMAL
45 008A 21FA00           LD    HL,00FAH
46 008D 220000           LD    (CLK2CT),HL    ;250 DECIMAL
47 0090 CD0000           CALL  IIRUPT          ;START THE CLOCK
48 0093 E1              POP    HL              ;RESTORE REGISTERS
49 0094 C9              RET                      ;EXIT
50 0095                SKP    H
    
```

```

01 0095      ;
02 0095      ;SUBROUTINE TO STOP THE CLOCK.
03 0095      ;-----
04 0095      ;
05 0095      ;FUNCTION = STOPS THE REAL-TIME CLOCK
06 0095      ;BY EXECUTING A DI INSTRUCTION.
07 0095      ;
08 0095      ;REGISTERS= ALL SAVED.
09 0095      ;
10 0095      ;TIME      = NON CRITICAL.
11 0095      ;
12 0095      STOP:  ENT          ;EXTERNAL ENTRY POINT
13 0095 F3     DI          ;STOP THE INTERRUPTS
14 0096 C9     RET         ;EXIT
15 0097      SKP      5

```

```

21 0097      ;
22 0097      ;MEMORY FOR REAL-TIME CLOCK.
23 0097      ;-----
24 0097      ;
25 0097      ;ALL DATA IS HELD IN BCD FORMAT.
26 0097      ;
27 0097      SECS:  DEFS  01H      ;SECONDS
28 0098      MINS:  DEFS  01H      ;MINUTES
29 0099      HOURS: DEFS  01H      ;HOURS
30 009A      END

```

```

EXIT  002B  HOURS  0099  INT    0000  MINS    0098  SCREEN  D011
SECS  0097  START  0083  STOP   0095

```

>M C800 C8E2

```

C800 F3 CD B2 C8 F5 E5 21 97 C8 7E C6 01 27 FE 60 20
C810 1A 3E 00 77 23 7E C6 01 27 FE 60 20 0E 3E 00 77
C820 23 7E C6 01 27 FE 24 20 02 3E 00 77 05 11 11 D0
C830 21 99 C8 7E E6 F0 C8 3F C8 3F C8 3F C8 3F C6 20
C840 12 13 7E E6 0F C6 20 12 13 3E 4F 12 13 2B 7E E6
C850 F0 C8 3F C8 3F C8 3F C8 3F C6 20 12 13 7E E6 0F
C860 C6 20 12 13 3E 4F 12 13 2B 7E E6 F0 C8 3F C8 3F
C870 C8 3F C8 3F C6 20 12 13 7E E6 0F C6 20 12 D1 E1
C880 F1 FB C9/E5/ 21 7D 00 22 9A C8 21 FA 00 22 9C C8
C890 CD D6 C8 E1 C9 /F3/ C9 (00:00 00)00 00 00 00 F5 3E
C8A0 74 32 07 E0 3A 9A C8 32 05 E0 3A 9B C8 32 05 E0
C8B0 F1 C9 F5 3E B0 32 07 E0 3A 9C C8 32 06 E0 3A 9D
C8C0 C8 32 06 E0 F1 C9 F5 E5 3E C3 32 38 10 21 00 C8
C8D0 22 39 10 E1 F1 C9 ED 56 CD 9E C8 CD C6 C8 CD B2
C8E0 C8 FB C9

```

RTCLOCK.ABS

- () Clock Data Bytes, hold secs, mins & hours in BCD Format
- [] Data words PRESCL & CLK2CT. These will be set to appropriate values to a call to start
- / / Entry points start & stop respectively
(C883) (C895)

The machine code program isn't a very friendly program so the following program in PASCAL interfaces between the user and the machine.

```

621D 10 PROGRAM RealTimeClockTest ;
621D 20
621D 30 (*A program to interface with the machine code RTCLOCK.ABS.*)
621D 40 (*You will need to limit RAM Top & RAM Top for T to 51000.*)
621D 50 (*Author W.Howard Date 22/11/84.*)
621D 60
621D 70 PROCEDURE Startclock ;
6229 80
6229 90 (*Starts the clock by a machine code call.*)
6229 100
6229 110 BEGIN
6241 120 USER(#C883)
6245 130 END ;
624D 140
624D 150 PROCEDURE Stopclock ;
6250 160
6250 170 (*Stops the clock by a machine code call.*)
6250 180
6250 190 BEGIN
6268 200 USER(#C895)
626C 210 END ;
6274 220
6274 230 FUNCTION BCD ( n : INTEGER ) : CHAR ;
6277 240
6277 250 (*A function to convert the integer n into BCD format.*)
6277 260
6277 270
6277 280 BEGIN
628F 290 BCD := CHR((n DIV 10) * 16 + (n MOD 10))
6287 300 END ;
62C1 310
62C1 320 PROCEDURE Clockinitialise ;
62C4 330
62C4 340 (*Initialises the clocks registers.*)
62C4 350
62C4 360 VAR input : INTEGER ;
62C4 370
62C4 380 BEGIN
62DC 390 WRITELN('Please input the hour.') ;
6300 400 READLN(input) ;
630C 410 POKE(#C899,BCD(input)) ;
6325 420 WRITELN('Please input the number of minutes.') ;
6356 430 READLN(input) ;
6362 440 POKE(#C898,BCD(input)) ;
6378 450 WRITELN('Please input the number of seconds.') ;
63AC 460 READLN(input) ;
6388 470 POKE(#C897,BCD(input))
63D0 480 END ;
63D8 490
63D8 500 BEGIN
63E1 510 WRITELN('Have you loaded RTCLOCK.ABS ?') ;
640C 520 Clockinitialise ;
6411 530 Startclock ;
6416 540 WRITELN(CHR(22)) ; (*Clear the screen.*)
6420 550 WRITELN ; WRITELN ; WRITELN ;
6429 560 WRITELN('Press any key to return to PASCAL.') ;
6459 570 REPEAT
6459 580 UNTIL INCH (<) CHR(0) ;
646E 590 Stopclock
646E 600 END.

```

End Address: 6475

The next program acts as a Real-Time Timer capable of counting 100th's of a second. Every time an interrupt is generated the location pointed at by PTRIND is incremented by one. As an accurate clock is required and a frequency of 100Hz is needed PRESCL=0002H and CLK2CT=009C. The machine code routines are not very easy to manipulate so a PASCAL program to simulate a Stop-Watch is also included after the machine code.

```

01 0000      ;
02 0000      ; INTERRUPT COUNTING VARIABLE.
03 0000      ; =====
04 0000      ;
05 0000      ; SUBROUTINE DEPENDANCE :- I08253 CLOCK SUBROUTINES.
06 0000      ;
07 0000      PTRIND: ENT          ; INDIRECT POINTER TO
08 0000      DEFS 02H           ; COUNTING VARIABLE
09 0002      ;
10 0002      ; INTERRUPT SUBROUTINE.
11 0002      ; -----
12 0002      ;
13 0002      ; FUNCTION = THIS SUBROUTINE INCREMENTS THE LOCATION
14 0002      ; POINTED AT BY THE ADDRESS IN PTRIND.
15 0002      ;
16 0002      ; REGISTERS= ALL SAVED.
17 0002      ;
18 0002      ; TIME      = 135.5 uSEC.
19 0002      ;
20 0002      INT:      ENT          ; EXTERNAL ENTRY POINT
21 0002 F3      DI
22 0003 CD0000  E      CALL  TIMER2      ; RESTART INTERRUPTS
23 0006 F5      PUSH  AF
24 0007 E5      PUSH  HL
25 0008 D5      PUSH  DE          ; SAVE REGISTERS
26 0009 2A0000  LD     HL,(PTRIND)    ; GET POINTER
27 000C 5E      LD     E,(HL)      ; GET LSB POINTED AT
28 000D 23      INC  HL          ; POINT TO MSB
29 000E 56      LD     D,(HL)      ; AND GET IT
30 000F 13      INC  DE          ; INCREMENT WORD
31 0010 72      LD     (HL),D      ; RESTORE MSB
32 0011 2B      DEC  HL          ; POINT TO LSB
33 0012 73      LD     (HL),E      ; RESTORE IT
34 0013 D1      POP  DE
35 0014 E1      POP  HL
36 0015 F1      POP  AF          ; RESTORE REGISTERS
37 0016 FB      EI              ; ENABLE INTERRUPTS
38 0017 C9      RET              ; EXIT
39 0018      END

```

```

>M C800 C86F
C800 (00 00) F3 CD 30 C8 F5 E5 D5 2A 00 C8 5E 23 56 13
C810 72 2B 73 D1 E1 F1 FB C9 00 00 00 00 F5 3E 74 32  VAR COUNT.ARS
C820 07 E0 3A 18 C8 32 05 E0 3A 19 C8 32 05 E0 F1 C9
C830 F5 3E B0 32 07 E0 3A 1A C8 32 06 E0 3A 1B C8 32
C840 06 E0 F1 C9 F5 E5 3E C3 32 38 10 21 02 C8 22 39
C850 10 E1 F1 C9 ED 56 CD 1C C8 CD 44 C8 CD 30 C8 FB
C860 C9 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

() PTRIND location

[] PRESCL & CLK2CT locations

// ITRUPT entry point

INT 0002 PTRIND 0000

```

62EB 10 PROGRAM stopwatch ;
62EB 20
62EB 30 TYPE laprange = 0 .. 9 ;
62EB 40
62EB 50 VAR T : INTEGER ;
62F4 60     key : CHAR ;
62F4 70
62F4 80 PROCEDURE initialise ;
62F7 90
62F7 100    VAR i : INTEGER ;
62F7 110
62F7 120    BEGIN
630F 130        i := #0002 ;
6318 140        POKE(#C818 , i) ; (*INITIALISE PRESCL*)
6327 150        i := #009C ;
6330 160        POKE(#C81A , i) ; (*INITIALISE CLK2CT*)
633F 170        USER(#C854) ; (*CALL IIRUPT*)
6345 180        INLINE(#F3) ; (*STOP THE INTERRUPTS*)
6346 190        POKE(#C800 , ADDR(T)) ; (*ADDRESS OF VARIABLE TO BE INCREMENTED*)
6352 200        T := 0 (*ZERO THE VARIABLE*)
6356 210    END ;
635F 220
635F 230 PROCEDURE stop ;
6362 240
6362 250    BEGIN
637A 260        INLINE(#F3) ; (*DI*)
637B 270        POKE(4466,CHR(3)) ;
6385 280        WRITELN((T DIV 100):4 , '.', (T MOD 100):2)
63B1 290    END ;
63BA 300
63BA 310 PROCEDURE start ;
63B0 320
63B0 330    BEGIN
63D5 340        INLINE(#FB) (*EI*)
63D6 350    END ;
63D0 360
63D0 370 PROCEDURE reset(VAR i : INTEGER) ;
63DF 380
63DF 390    BEGIN
63F7 400        i := 0
6402 410    END ;
6400 420 PROCEDURE lap(i : laprange) ;
6410 430
6410 440    BEGIN
6428 450        POKE(4466,CHR(i + 11)) ;
643E 460        WRITE(' ', i , ' ');
6468 470        WRITELN((T DIV 100):4 , '.', (T MOD 100):2)
6494 480    END ;
649E 490
649E 500 BEGIN
64A7 510     initialise ;
64AC 520     WRITE(CHR(22)) ;
64B3 530     WRITELN(' STOP WATCH ') ;
64D0 540     reset(T) ;
64D9 550     WRITELN ;
64DC 560     WRITELN ;
64DF 570     WRITELN((T DIV 100):4 , '.', (T MOD 100):2) ;
650E 580     WRITELN ;

```

```

6511 590  WRITELN('S to Start.') ;
652A 600  WRITELN('D to Stop.') ;
6542 610  WRITELN('R to Reset.') ;
655B 620  WRITELN('0 to 9 for Lap Times.') ;
657E 630  WRITELN('P to Return to PASCAL.') ;
65A2 640  REPEAT
65A2 650      key := INCH ;
65AB 660      POKE(4466 , CHR(3)) ;
65B5 670      WRITELN('T DIU 100):4 , '.', (T MOD 130):2) ;
65E4 680      CASE key OF
65E7 690          'S' : start ;
65F4 700          'D' : stop ;
6601 710          'R' : reset(T) ;
6612 720          '0','1','2','3','4','5','6','7','8','9' : lap(ORD(key) - 48)
6654 730      END
6659 740  UNTIL key = 'P' ;
666A 750  WRITE(CHR(22))
6671 760  END.
End Address: 6673
Run?N

```

A Question of Efficiency

Any interrupt routine must be efficient in terms of speed. For example if we are generating 50 interrupts a second then if the interrupt routine takes longer than 1/50th of a second then the next interrupt will have arrived before we have finished processing the current one! This obviously means that we do nothing but execute the interrupt routine which defeats the whole point of interrupts!

In general if the interrupt routine takes longer than one half of the time between interrupts then the interrupt routine should be considered inefficient as the main program is receiving less than one half of the processors time. Changing the interrupt generators speed or rewriting the interrupt routine may be possible.

Accuracy of the Interrupt Generator

As the IC8253 clocks can only divide by integers (and then with strange results) the precise clock frequency cannot be predicted. This should not be a problem for the hobbyist. The clock frequency is further corrupted by the response time of the CPU to the interrupt i.e. the time between generating the interrupt and the processor starting to execute the interrupt routine. Further resetting Clock 2 every time will further slow down the clock frequency! To add insult to injury the input frequency the input frequency of 31250Hz to Clock 1 is not that accurate! What this means is that if you want really accurate interrupts you have bought the wrong machine!!

Odds and Ends

As we reprogram Clock 2 every time an interrupt occurs we have two options, either to reprogram it at the start of the interrupt routine or at the end. If we reprogram it at the start then we get interrupts every 150th (say) of a second. If however we reprogram it at the end of the routine then we will spend 1/50th of a second in the main program between interrupts which has its uses.

Calling any Monitor routine to use the cassette recorder will switch the interrupts off! It is however possible to counteract this by calling the Monitor routine TIMEST at 0033H before ANY reprogramming of IC8253 occurs. However it is still possible to switch the interrupts off if we try to save a file of 0 bytes in length, although why we would possibly want to do this is beyond me!!!!

It is absolutely essential that any register used within the interrupt routine is saved. This INCLUDES the F register.

Finally, A Few Ideas

Interrupts can be used effectively in games programs e.g. spaceship control could be put in the interrupt routine while the main program deals with the screen and collisions.

The Real-Time clock program is fairly accurate over periods of time up to two or three hours and could therefore be used to good effect in adventure or arcade games.

As mentioned at the start event occurring interrupts can be easily generated for egg-timer programs etc.

Well I hope that this article inspires some ideas and I look forward to seeing some programs using these ideas in future issues of the Sharpsoft User Notes.

COPYRIGHT © William Howard 18/12/84



MZ-80K NOTES & LISTINGS

```

650 IFG$="R"THENMUSIC"_C1"0":GOSUB300:GOTO540
660 IFG$="D"THEN680
670 GOTO640
680 PRINT"C_____":MUSIC"_C1"0"
690 INPUT"How many hours have you decided to          employ each week? ";FL
691 FL=INT(FL)
692 IFFL>1200THENPRINT"_You cannot employ more than 1200 hours> per week."
694 IFFL>1200GOSUB 30
695 IFFL>1200GOTO 680
696 IFFL<100THENPRINT"_You will need a lot more than that!"
697 IFFL<100GOSUB30
698 IFFL<100THEN680
699 OL=INT(FL/2):IFOL+FL>1200THENOL=1200-FL
700 IFCH=14THENRETURN
701 ZB=1
720 GOSUB70
730 PRINT"C_____Now you must decide how much to pay yourworkers per hour."
740 PRINT"_Remember.  If you pay too much your hatswill cost too much to ";
750 PRINT"make.  If you pay too little you may suffer from poor ";
760 PRINT"industrial relations.":PRINT"_You may choose between £1.50 and ";
770 PRINT"£2.50 per hour which is the normal range for the industry."
780 PRINT"_____Press R to see Production Information."
790 PRINT"Press D if you are ready to decide."
800 GETG#
810 IFG$="R"THENMUSIC"_C1"0":GOSUB300:GOTO730
820 IFG$="D"THENMUSIC"_C1"0":GOTO840
830 GOTO800
840 PRINT"C_____How much do you want to pay per hour?"
850 INPUT"£";L
855 L=(INT(L*100))/100
860 IFL<1.5THEN890
870 IFL>2.5THEN890
880 GOTO895
890 MUSIC"_C4":PRINT"_You must pay between £1.50 and £2.50":GOSUB 30:GOTO840
895 RA=INT(100*(2.5-L)/2.5)
900 GOSUB70
910 PRINT"CYou will need some capital to start yourbusiness;";
915 PRINT" the following are an absoluteminimum:"
920 PRINT">>>>Factory";TAB(29);"35 000"
930 PRINT">>>>Machinery etc>>>>>>>>>>20 000"
935 IFZC=1THENRETURN
940 PRINT">>>>One week's wages>>>>";INT(FL*L);:GOSUB 50
945 PRINT">>>>One week's overheads>>>>>>>>2>000"
950 PRINT"__How much additional cash do you want? >You can buy raw ";
960 PRINT"materials on credit.  However, your customers will also ";
970 PRINT"expectcredit from you.  Your bank manager will probably ";
980 PRINT"allow you some overdraft facility but you should not ";
990 PRINT"rely too heavily on this at first."
1000 PRINT"_____Press R to see Production Information."
1010 PRINT"Press D if you are ready to decide."
1020 GETG#
1030 IFG$="R"THENMUSIC"_C1"0":GOSUB300:GOTO910
1040 IFG$="D"THENMUSIC"_C1"0":GOTO1060
1050 GOTO1020
1060 PRINT"C_____How much additional cash do you want?"
1070 INPUT"£";C:C=INT(C+(FL*L)+2000)
1080 PRINT"C_____Your total capital requirement is now asfollows:"
1090 ZC=1:GOSUB920
1100 PRINT">>>>Cash";TAB(25);C;:GOSUB50
1110 PRINTTAB(28);"_____"
1120 PRINTTAB(25);C+35000+20000;:GOSUB50
1130 PRINTTAB(28);"_____"
1140 GOSUB30
1150 GOSUB70

```


MS-DOS 5.0 NOTES & LISTINGS

```

1650 POKES9555,1
1660 NEXTW
1670 PRINT"CInformation available":PRINT"-----"
1680 PRINT" 1 Interest rates"
1690 PRINT" 2 Weather forecast"
1700 PRINT" 3 Production information"
1710 PRINT" 4 Production, sales and order records"
1720 PRINT" 5 Profit and loss account"
1730 PRINT" 6 Balance sheet"
1740 PRINT" 7 Breakdown of stocks"
1745 IF RB>2 THEN PRINT"__ 8 Start a new game":FORW=1TO5:S(W,WK)=0:NEXTW
1747 IFRB=4THEN PRINT" ? See final result again"
1748 IFRB>2THENGOTO1912
1750 IF MS=1 THEN PRINT" 8 Message"
1760 PRINT" _Actions available":PRINT"-----"
1770 PRINT"10 Pay creditors"
1780 PRINT"11 Chase debtors"
1790 PRINT"12 Bank deposit transaction"
1800 PRINT"13 Purchase raw materials"
1810 PRINT"14 Change labour force"
1820 PRINT"15 Advertise"
1830 PRINT"16 Sell surplus stocks at a discount"
1840 PRINT"17 Fix production levels and selling prices for week";WK+1
1845 IFMS>1THEN1910
1850 FORW=1TO10
1860 IFMS=1THEN PRINT"H_____>>>> <<<<<<<<";
1870 FORWW=1TO100:NEXTWW
1880 PRINT"Message:"
1890 FORWW=1TO100:NEXTWW
1900 NEXTW
1910 MUSIC"_C1R"EBR_C"
1912 PRINT"H_____ Choose your next move.:"
1915 INPUTCH
1920 MUSIC"D1EF"
1930 IF (RB>2)*(CH=8) THENRUN
1940 IF (RB=4)*(CH=9) THENPRINT"C":GOTO22040
1950 IFMS=0THENIFCH=8THEN1670
1960 ONCHGOSUB2000,3000,300,5000,6000,7000,8000,9000
1962 IFCH=10THENGOSUB11000
1964 IFCH=11THENGOSUB12000
1966 IFCH=12THENGOSUB13000
1970 IFCH=13THENGOSUB14000
1972 IFCH=14THENGOSUB15000
1974 IFCH=15THENGOSUB16000
1976 IFCH=16THENGOSUB17000
1977 IFMS=0THEN1980
1978 MS=0:IF (CH=0)+(CH=9)+(CH>17) THENMS=1
1980 IFCH=17THENWK=WK+1:RA=RA+5:GOTO18000
1990 GOTO1670
2000 PRINT"C__Interest rates":PRINT"-----"
2010 PRINT" _Payable on loan capital>>>>>>";CR;"%"
2015 IF CR<10 THEN PRINT "^";TAB(31);CR;"%"
2020 PRINT" _Payable on overdrafts>>>>>>";BR;"%"
2025 IF BR<10 THEN PRINT "^";TAB(31);BR;"%"
2030 PRINT" _Receivable on bank deposits>>>>";DR;"%"
2033 IFDR<10THENPRINT"^";TAB(31);DR;"%"
2035 IFZJ=1THENRETURN
2040 PRINT"__All interest is paid weekly."
2050 GOSUB30
2060 RETURN
3000 PRINT"C__Seven Day Weather Forecast"
3010 PRINT"-----"
3020 IFWR=1THENPRINT"Fine"
3030 IFWR=2THENPRINT"Mainly fine"

```

```

3040 IFWR=3THENPRINT"Occasional showers"
3050 IFWR=4THENPRINT"Showers"
3060 IFWR=5THENPRINT"Rainy"
3065 PRINT
3070 IFWT=1THENPRINT"Hot"
3080 IFWT=2THENPRINT"Warm"
3090 IFWT=3THENPRINT"Mild"
3100 IFWT=4THENPRINT"Cool for the time of year"
3110 IFWT=5THENPRINT"Cold for the time of year"
3115 PRINT
3120 IFWS=1THENPRINT"Sunny"
3130 IFWS=2THENPRINT"Some cloud"
3140 IFWS=3THENPRINT"Cloudy"
3150 GOSUB30
3160 RETURN
5000 IFWK=0THENPRINT"C____There are no records yet.":GOSUB30:RETURN
5002 PRINT"C__Press 1 to see order records or 2 to see"
5003 PRINT"^production and sales records."
5004 GETG:IFG=0THEN5004
5005 ON G60T05920,5030
5030 PRINT"C__For which product do you wish to see  production and sales";
5035 PRINT" records?"
5040 PRINT"___";H$(1);">>>->>>press 1_"
5050 PRINTH$(2);">>>>>->>>press 2_"
5060 PRINTH$(3);">>>>>->>>press 3_"
5070 PRINTH$(4);">>>->>>press 4_"
5080 PRINTH$(5);">>>>>->>>press 5_"
5090 GETG
5095 IFG=0THEN5090
5100 IFG<6THENGOSUB5200
5105 IFG>5THENPRINT"C":G0T05030
5110 PRINT"C_____If you wish to see information on other products";
5120 PRINT" press 1."
5130 PRINT" _Otherwise press 2."
5140 GETG
5150 IFG=0THEN5140
5160 IFG=1THENPRINT"C":G0T05030
5170 IFG=2THENRETURN
5180 G0T05110
5200 PRINT"C";
5205 ONWK60T05290,5285,5280,5275,5270,5265,5260,5255,5250,5245,5240,5235
5235 ZE=12:V=30:Y=18:GOSUB5300
5240 ZE=11:V=22:Y=18:GOSUB5300
5245 ZE=10:V=13:Y=18:GOSUB5300
5250 ZE=9:V=4:Y=18:GOSUB5300
5255 ZE=8:V=30:Y=10:GOSUB5300
5260 ZE=7:V=22:Y=10:GOSUB5300
5265 ZE=6:V=13:Y=10:GOSUB5300
5270 ZE=5:V=4:Y=10:GOSUB5300
5275 ZE=4:V=30:Y=2:GOSUB5300
5280 ZE=3:V=22:Y=2:GOSUB5300
5285 ZE=2:V=13:Y=2:GOSUB5300
5290 ZE=1:V=4:Y=2:GOSUB5300
5295 GOSUB5700
5297 GOSUB30
5299 RETURN
5300 POKE4465,V:POKE4466,Y:PRINTO(G,ZE):GOSUB 50
5310 POKE4465,V:POKE4466,Y+1:PRINTP(G,ZE):GOSUB50
5315 S(G,ZE)=INT(S(G,ZE))
5320 POKE4465,V:POKE4466,Y+2:PRINTS(G,ZE):GOSUB50
5330 POKE4465,V:POKE4466,Y+3:PRINTO(G,(ZE+1)):GOSUB50
5350 RETURN
5700 PRINT"HWeek>>>>>>>1>>>>>>>2>>>>>>>3>>>>>>>4"
5710 PRINT"_____";

```

```

5720 PRINT"Op Stk"
5730 PRINT"Made":PRINT"Sold"
5740 PRINT"C1 Stk"
5760 PRINT"_____";
5770 PRINT"_Week>>>>>>>5>>>>>>>6>>>>>>>7>>>>>>>8"
5780 PRINT"_____";
5790 PRINT"Op Stk"
5800 PRINT"Made":PRINT"Sold"
5810 PRINT"C1 Stk"
5830 PRINT"_____";
5840 PRINT"_Week>>>>>>>9>>>>>>>10>>>>>>>11>>>>>>>12"
5850 PRINT"_____";
5860 PRINT"Op Stk"
5870 PRINT"Made":PRINT"Sold"
5880 PRINT"C1 Stk"
5900 PRINT"_____";
5910 RETURN
5920 IFWK=0THENPRINT"C___There have been no orders yet.":GOSUB30:RETURN
5921 PRINT"COrders Received":PRINT"_____";
5922 PRINT"Wk Sports Boater Rain Novelty Sun"
5924 FORW=1TO40:PRINT"-";:NEXTW:PRINT
5926 FORG=1TOWK:FORGG=5TO1STEP-1:PRINTTAB(6*GG-(7-GG));"^-";OP(GG,G):;GOSUB50
5928 NEXTGG:PRINT"^-";G;"_":NEXTG
5930 FORW=1TO39:PRINT"-";:NEXTW:GOSUB30:RETURN
6000 PRINT"C"
6010 IFWK=0THENPRINT"_____There is no profit and loss account yet";
6020 IFWK=0THENPRINT" since you have not started trading.":GOSUB30:RETURN
6030 IFWK=1THENPRINT"Profit and Loss Account for first week"
6040 IFWK=1THENPRINT"_____":GOTO6070
6050 PRINT"Profit and Loss Account for";WK;" weeks"
6060 PRINT"_____";:IFWK>9THENPRINT"---"
6070 PRINT"_":PRINT"Sales";TAB(24);S;:GOSUB50
6075 PC=INT(PC)
6080 PRINT"Cost of Production>>>>>";PC;:GOSUB50
6090 PRINTTAB(26);"_____"
6100 PRINT"Trading Profit>>>>>>>";(S-PC);:GOSUB50
6105 PRINTTAB(26);"_____"
6110 PRINT"Fixed Overheads>>>>>>>";FT;:GOSUB50
6120 IFIT>0THENPRINT"Interest on capital>>>>>";IT;:GOSUB50
6130 IFIB>0THENPRINT"Interest on overdraft>>>";IB;:GOSUB50
6140 IFAV>0THENPRINT"Advertising";TAB(24);AV;:GOSUB50
6150 IFBD>0THENPRINT"Bad Debts Written Off>>>";BD;:GOSUB50
6160 IFLE>0THENPRINT"Legal Expenses>>>>>>>";LE;:GOSUB50
6170 IFUF>0THENPRINT"Uninsured Fire Losses>>>";UF;:GOSUB50
6175 IFFC>0THENPRINT"Factor's Fees>>>>>>>>>>";FC;:GOSUB50
6176 UL=INT(UL)
6177 IFUL>0THENPRINT"Unused Labour>>>>>>>>>>";UL;:GOSUB50
6180 PRINTTAB(26);"_____"
6185 OC=FT+IT+IB+AV+BD+LE+UF+FC+UL
6190 PRINTTAB(24);OC;:GOSUB50
6192 PRINTTAB(26);"_____"
6193 IFID=0THEN6205
6195 IFID>0THENPRINT"Interest on Deposit>>>>>";ID;:GOSUB50
6200 PRINTTAB(26);"_____"
6205 NP=INT((S-PC)-OC+ID)
6210 PRINT"Net Profit";TAB(24);NP;:GOSUB50
6220 PRINTTAB(26);"_____"
6230 GOSUB30
6240 RETURN
7000 PRINT"C>>>>>>>>>>Headgear Limited"
7002 OC=FT+IT+IB+AV+BD+LE+UF+FC+UL
7003 NP=INT(S-(PC+OC)+ID);PF=NP

```

```

7005 IFWK=0THENPRINT"Balance Sheet before start of trading"
7006 IFWK=0THENPRINT"_____":GOTO7030
7010 PRINT"Balance Sheet as at the end of week";WK
7020 PRINT"_____";:IFWK>9THENPRINT "-";
7030 PRINT:PRINT"Assets_<<<<<<_____
7040 PRINT"Freehold Property>>>>>>>>";35000;:GOSUB50
7050 PRINT"Plant and Machinery>>>>>>>";20000;:GOSUB50
7055 RM=INT(RM):FG=INT(FG):C=INT(C)
7060 IFRM>0THENPRINT"Stocks of Raw Materials>>>";RM;:GOSUB50
7070 IFFG>0THENPRINT"Stocks of Finished Goods>>";FG;:GOSUB50
7075 IFDB>0THENPRINT"Debtors";TAB(26);DB;:GOSUB50
7080 IFD>0THENPRINT"Bank Deposit";TAB(26);D;:GOSUB50
7085 C=INT(((SC+PF+LC+CD)-(RM+FG+DB+D+C+55000)))+C
7086 IFINT(((SC+PF+LC+CD)-(RM+FG+DB+D+C+55000))<>0THEN7085
7090 IFC<>0THENPRINT"Cash";TAB(26);C;:GOSUB50
7100 PRINTTAB(25);"_____"
7110 PRINTTAB(26);55000+RM+FG+DB+D+C;:GOSUB 50
7120 PRINTTAB(25);"_____"
7130 PRINT"Liabilities_<<<<<<<<<_____
7140 PRINT"Share Capital";TAB(26);SC;:GOSUB50
7145 IFFP>0THENPRINT"Accumulated Profit>>>>>>>";PF;:GOSUB50
7147 IFFP<0THENPRINT"Accumulated loss>>>>>>>>";PF;:GOSUB50
7150 IFLC>0THENPRINT"Loan Capital";TAB(26);LC;:GOSUB50
7160 IFCD>0THEN PRINT "Creditors";TAB(26);CD;:GOSUB50
7170 PRINTTAB(25);"_____"
7180 PRINTTAB(26);SC+LC+CD+PF;:GOSUB50
7190 PRINT TAB(25);"_____"
7200 GOSUB30
7210 RETURN
8000 PRINT"C"
8010 IFFG=0THENIFRM=0THENPRINT"____You do not have any stocks at present."
8020 IFFG=0THENIFRM=0THENGOSUB 30:RETURN
8030 PRINT"____>>>>Stocks Held at the end of Week";WK
8040 PRINT">>>>_____";:IFWK>9THENPRINT"^-";
8050 PRINT:PRINTTAB(12);"Units of Raw>>Units of "
8060 PRINTTAB(12);"Materials>>>>Finished Goods"
8070 PRINT"^-_____
8073 PRINT"^-";
8075 FORZG=1TO5:RM(ZG)=INT(RM(ZG)):PRINTTAB(10);RM(ZG);:GOSUB50
8077 PRINT"^-";TAB(24);FG(ZG);:GOSUB50
8079 NEXTZG
8080 PRINT"^^^^^";H$(1)
8090 FORW=2TO5:PRINTH$(W):NEXTW
8130 PRINT"_____
8140 GOSUB30
8150 RETURN
9000 FORW=1TO4
9001 PRINT"C_____":FORWW=1TO15:PRINT"Message ";:MUSIC"EA1":NEXTWW
9002 FORWW=1TO100:NEXTWW
9003 PRINT"C_____":FORWW=1TO15:PRINTSPC(8);:NEXTWW
9004 NEXTW
9005 DNMTGOSUB9100,9200,9300,9400,9500,9600,9700,9800,9900
9007 IFMT=21THENGOSUB9950
9010 IFMT>1THENGOSUB30
9020 RETURN
9100 PRINT"C_Weather prospects have changed._"
9110 PRINT"The seven day forecast is now ___"
9120 GOSUB 3020
9140 RETURN
9200 PRINT"C_Interest rates have gone";
9210 IFZI>0THENPRINT" up."
9220 IFZI<0THENPRINT" down."
9230 PRINT"The rates are now as follows:"
9240 ZJ=1:GOSUB 2010:ZJ=0:RETURN

```

MZ-80K NOTES & LISTINGS

```

9300 PRINT"C_____A major trade union has started a      recruiting";
9310 PRINT" drive amongst your labour      force."
9320 RETURN
9400 PRINT"C_____Raw material costs for ";
9410 ONZLGOSUB9421,9422,9423,9424,9425:GOTO9430
9421 PRINTH$(1)::GOTO9427
9422 PRINTH$(2)::GOTO9427
9423 PRINTH$(3)::GOTO9427
9424 PRINTH$(4)::GOTO9427
9425 PRINTH$(5);
9427 PRINT"s":RETURN
9430 IFMT=5THENRETURN
9432 PRINT"have gone up by about";ZK; "%."
9440 PRINT" _The raw material cost>per unit is      now £";MA(ZL)
9450 MS=1:RETURN
9500 PRINT"C_____Variable overheads for ";
9510 GOSUB9410
9530 PRINT"have gone up by about";ZK; "%."
9540 PRINT" _The variable overhead per unit is      now £";PO(ZL)
9550 MS=1:RETURN
9600 PRINT"C_____Weekly fixed overheads have gone up"
9605 PRINT"by about";ZM; "% to £";FO
9610 RETURN
9700 ONZSGOTO9710,9730,9750
9710 PRINT"C_____A successful pop music band has appearedon BBC tv's";
9720 PRINT" Top of the Pops to perform their latest number one hit ";
9725 PRINT"wearing      novelty hats made by Headgear Ltd."
9727 RETURN
9730 PRINT"C_____An influential fashion writer has "
9735 PRINT"published an article in a national"
9737 PRINT"nespaper under the headline"
9740 PRINT"__>>>>>>THE BOATER IS BACK!"
9745 RETURN
9750 PRINT"C_____The serious press has given considerable";
9755 PRINT"publicity to a claim by a medical      researcher";
9760 PRINT" that the wearing of hats      during the summer months";
9765 PRINT" encourages      the growth of head lice."
9770 RETURN
9800 ONYAGOTO9810,9840,9870
9810 IFYB=1THEN9890
9815 PRINT"C_The FT share index is at its lowest      point for three years."
9830 YB=1:RETURN
9840 IFYC=1THEN9890
9845 PRINT"C_____Unemployment amongst unskilled workers is starting to fall."
9860 YC=1:RETURN
9870 IFYD=1THEN9890
9875 PRINT"C_____A general election is to be held soon."
9887 YD=1:RETURN
9890 PRINT"C_____ "
9891 FORW=1TO10
9892 PRINT" If you want to get ahead, get a hat!"
9893 MUSIC" _fD0R0_E"
9894 NEXTW
9895 RETURN
9900 PRINT"C_____A customer has gone bankrupt owing      you £";YH; "."
9910 PRINT" _This amount has been written off as a      bad debt."
9920 RETURN
9950 PRINT"C_____As a result of your solicitor's actions overdue debts ";
9960 PRINT"of £";YG; " have been paid."
9970 RETURN
10400 ZK=INT(1+RND(4)*20)
11000 PRINT"C"

```

```

11010 IFCD=0THENPRINT"_____You do not have any creditors.":GOSUB30:RETURN
11020 PRINT"C_____You owe £";CD;" to creditors."
11030 PRINT"_____Press 1 if you wish to see your balancsheet."
11040 PRINT"_Press 2 if you are ready to make your payment."
11050 GETG
11060 IFG=0THEN11050
11070 IFG=1THENGOSUB7000:GOTO11020
11080 IFG=2THEN11100
11090 GOTO11020
11100 PRINT"C_____How much do you want to pay"
11105 PRINT"to creditors ?"
11110 INPUT"£";YE
11115 YE=INT(YE)
11120 IFYE>CDTHENPRINT"_You only owe £";CD
11130 IFYE>CDTHENGOSUB30:GOTO11100
11140 C=C-YE;CD=CD-YE
11150 GOSUB 30:RETURN
12000 PRINT"C"
12010 IFFE=0THEN12040
12020 PRINT"_____You have already employed a factor so there is ";
12030 PRINT"no further action you can take against debtors.":GOSUB30:RETURN
12040 IFDB>0THEN12070
12050 PRINT"_____You do not have any debtors at present."
12060 GOSUB30:RETURN
12070 PRINT"_____There are two ways you can try to improve the flow of "
12080 PRINT"cash from your debtors. One is to instruct a solicitor to";
12090 PRINT"write to debtors demanding payment. The other is to ";
12100 PRINT"employ a factor."
12110 GOSUB30
12120 PRINT"C_____It is pointless to ask a solicitor to take action ";
12130 PRINT"over debts which are less than six weeks old._"
12140 OD=INT(DB-((S/WK)*6))
12143 IFOD<0THENOD=0
12145 IFWK<7THENOD=0
12150 PRINT"At present you have";
12160 IFOD<1THENPRINT" no ";:GOTO12180
12170 PRINT" £";OD;" in ";
12180 PRINT"debts":PRINT"six weeks old or more.";
12190 IFOD<1THEN12210
12200 PRINT" These are owed by";INT(OD/100)+1;" different debtors.";
12205 PRINT" Each":PRINT"solicitor's letter costs £25."
12210 GOSUB30
12220 PRINT"C___You can sell your debtors to a factor."
12230 PRINT"He will give you immediate cash for themless 5%";
12240 PRINT" of their value which is his fee. Once you have sold them ";
12250 PRINT"the factortakes over all the work of collection and ";
12260 PRINT"carries the risk of non-payment._"
12270 PRINT"A factor will only buy your debtors if you agree to sell ";
12280 PRINT"all of them to him forat least the next 12 weeks."
12290 GOSUB 30
12300 PRINT"C_____Remember. Action against slow paying customers may ";
12310 PRINT"discourage them from buying from you in future."
12320 GOSUB30
12330 PRINT"C_____Do you now want to take action against debtors ? ____"
12340 PRINT"Press 1 if you do. _"
12350 PRINT"Press 2 if you don't. _"
12360 PRINT"Press 3 if you want to see your"
12365 PRINT">>>>>>balance sheet before":PRINT">>>>>>deciding."
12370 GETG
12380 IFG=0THEN12370
12390 IFG=1THENMUSIC"G0_A1":GOTO12420
12400 IFG=2THENMUSIC"_G0_A1":RETURN
12410 IFG=3THENMUSIC"_A0A_A":GOSUB7000:GOTO12330
12415 GOTO12330

```

MZ-80K NOTES & LISTINGS

```

12420 PRINT"C_____Press 1 to instruct your solicitor._"
12430 PRINT"Press 2 to employ a factor."
12440 GET G
12450 IF G=0 THEN 12440
12460 IFG=1THENMUSIC"£A0A1":GOTO12500
12470 IFG=2THENMUSIC"£A0-A1":GOTO12700
12480 GOTO12420
12500 PRINT"C"
12510 IFOD>0THEN12560
12520 PRINT"_____You are not owed any debts which are      over six weeks";
12530 PRINT" old so there is no point in instructing your solicitor."
12540 GOSUB30:RETURN
12560 PRINT"___Overdue debts amount to f";OD
12570 PRINT"owed by";INT(OD/100)+1;" different customers._"
12580 PRINT"How many of these customers do you want your solicitor ";
12590 PRINT"to write to ?_"
12600 INPUTYF:IFYF=0THEN12330
12610 IFYF>INT(OD/100)+1THEN12500
12620 YG=INT(((OD/(INT(OD/100)+1))*YF)*RND(1))
12625 LE=LE+25*YF:C=C-25*YF
12630 PRINT"C_____You have instructed your solicitor."
12640 FORW=1TOS:FF(W)=FF(W)-5:NEXTW
12650 GOSUB30:RETURN
12700 PRINT"C_____You have employed a factor."
12710 FE=1
12720 C=C+INT(DB*.95)
12730 FC=FC+INT(DB*.05)+1
12740 DB=0:IFCH<>11THENRETURN
12750 GOSUB 30:RETURN
13000 PRINT"C_____Press 1 to make a deposit."
13010 PRINT"__Press 2 to make a withdrawal."
13015 PRINT"_____Remember that cash on deposit is subjectto one week's";
13017 PRINT" notice of withdrawal."
13020 GET G:IFG<1THEN13020
13030 IFG=1THEN13100
13040 IFG=2THEN13200
13050 GOTO13000
13100 IFC<1THENPRINT"C_____You do not have any cash to deposit.":GOSUB30:RETURN
13110 INPUT "C_____How much do you wish to deposit ? f";YJ:YJ=INT(YJ)
13120 IF YJ>C THEN PRINT"__You do not have this much cash.":GOSUB30:GOTO13110
13130 C=C-YJ: D=D+YJ
13140 GOSUB 30: RETURN
13200 IFD<1THENPRINT"C_____You do not have any cash on deposit.":GOSUB30:RETURN
13210 INPUT"C_____How much do you wish to withdraw ? f";YJ:YJ=INT(YJ)
13220 IFYJ>DTHENPRINT"__You do not have this much on deposit."
13230 IFYJ>DTHEN GOSUB 30: GOTO 13210
13235 PRINT"C_____f";YJ;" will be available next week."
13240 YI=1:GOSUB30:RETURN
14000 PRINT"C_____For which product do you wish to buy      materials ?"
14010 PRINT"_____1      ";H$(1)
14020 PRINT"_____2      ";H$(2)
14030 PRINT"_____3      ";H$(3)
14040 PRINT"_____4      ";H$(4)
14050 PRINT"_____5      ";H$(5):IFCH<>13 THEN RETURN
14055 PRINT"__Press 6 to see current stocks."
14060 GETG:IFG=0THEN14060
14080 IFG>6THEN14000
14090 MUSIC"£A0-B-A1"
14095 IFG=6THENGOSUB8000:GOTO14000
14097 ONGGOSUB14300,14310,14320,14330,14340
14100 PRINT"_____The present cost of materials per hat isf";MA(G);"."
14105 PRINT"Your present stock is";RM(G);" units.____"
14110 INPUT"__How many hats do you want to buy      material for ? ";YK

```


MZ-80K NOTES & LISTINGS

```

17170 MUSIC"AOEA BC":RETURN
18000 IFZT>0THENFF(4)=FF(4)-10:ZT=ZT-1
18010 IFYI=1THENC=C+YJ:D=D-YJ:YI=0
18020 YM=0:LW=0
18030 FORW=1TO5:IFFF(W)>120THENFF(W)=INT(FF(W)-((FF(W)-120)*.6))
18040 NEXTW
18044 DIMYS(5)
18055 IF(YT>0)*(YT<6)THENFF(YT)=INT(FF(YT)*.75):YT=0
18056 IFYT=6THENFORW=1TO5:FF(W)=INT(FF(W)*.9):YT=0
18060 IFRA<10THENRL=0
18070 IFRL=0THEN18100
18080 PRINT"C___Your workforce has gone on strike for   ";
18090 PRINT"more pay. This week there will be no   ";
18094 PRINT"production.   ";IFFG=0THEN18097
18096 PRINT"You can now fix prices for the sale of any hats in stock."
18097 IFFG=0THENPRINT"There are no finished goods in stock.":GOSUB30:GOTO20000
18098 GOTO18120
18100 PRINT"C_____You must now decide production levels and selling   ";
18110 PRINT"prices for week";WK:PRINT"for each of your five products."
18120 GOSUB30
18130 FORG=1TO5
18135 PRINT"C";
18140 FORW=1TO24:PRINTH$(G):MUSIC"EGOG1":NEXT W
18150 IFRL=1THEN18500
18160 PRINT"H_____";TAB(15);"How many   ";H$(G);"s"
18170 PRINTTAB(15);"do you plan to make in"
18180 PRINTTAB(15);"week";WK;" ?_"
18190 INPUT">>>>>>>>>>>>>>>>";P(G,WK)
18192 P(G,WK)=INT(P(G,WK))
18195 IF P(G,WK)<RM(G) THEN 18255
18220 PRINTTAB(15);"You only have materials "
18230 PRINTTAB(15);"for";RM(G);">";H$(G);"s."
18240 FORW=1TO5000:NEXTW
18250 GOTO18135
18255 IFP(G,WK)*L(G)<=FL+OL-YPTHEN18300
18280 PRINTTAB(15);"You only have hours"
18290 PRINTTAB(15);"for";INT((FL+OL-YP)/L(G))-1;">";H$(G);"s.":GOTO18240
18300 RM(G)=RM(G)-P(G,WK):RM=INT(RM-(P(G,WK)*MA(G))):IF RM<0 THEN RM=0
18310 O(G,WK+1)=O(G,WK)+P(G,WK)
18320 FG(G)=FG(G)+P(G,WK)
18330 YP=YP+P(G,WK)*L(G)
18340 IFYP<FLTHENPC(G)=PC(G)+INT(P(G,WK)*L(G)*L):C=C-INT(P(G,WK)*L(G)*L)
18350 IFYP<FLTHEN18420
18360 IF(YP-(P(G,WK)*L(G)))>=FLTHENPC(G)=PC(G)+INT(P(G,WK)*L(G)*L*1.5)
18370 IF(YP-(P(G,WK)*L(G)))>=FLTHENC=C-INT(P(G,WK)*L(G)*L*1.5)
18380 IF(YP-(P(G,WK)*L(G)))>=FLTHEN18420
18390 PC(G)=PC(G)+(FL-(YP-P(G,WK)*L(G)))*L
18400 C=C-(FL-(YP-P(G,WK)*L(G)))*L
18410 PC(G)=PC(G)+(YP-FL)*L*1.5:C=C-(YP-FL)*L*1.5
18420 PC(G)=PC(G)+INT(MA(G)*P(G,WK))
18430 PC(G)=PC(G)+INT(PO(G)*P(G,WK)):C=C-INT(PO(G)*P(G,WK))
18455 IF FG(G)<1 THEN 18600
18460 PRINT"H";
18470 FOR W=1 TO 24
18480 PRINTTAB(15);SPC(24)
18490 NEXT W
18500 IF FG(G)=0THEN18600
18505 PRINT"H";TAB(15);"_____What price will you ask"
18510 PRINTTAB(15);"for   ";H$(G);"s"
18520 PRINTTAB(15);"this week ?"
18530 INPUT"__>>>>>>>>>>>>>>>>f";YR
18540 IF(YR>.05)*(YR<5.99)THEN18555
18550 PRINTTAB(15);"_Be sensible !":FOR W =1TO 2000:NEXT W:GOTO18460
18555 IFYR=Q(G)THEN18600
18556 YU=Q(G):G(G)=YR

```

```

18560 IFWK=1THENFF (G)=FF (G)-60*(Q (G)-(PO (G)+MA (G)+L*L (G)+.5)):GOTO18600
18562 FF (G)=FF (G)-60*(Q (G)-YU)
18600 NEXT G
18602 IFRL=0THENIFYP<FLTHENUL=UL+(FL-YP)*L:C=C-(FL-YP)*L
18604 YF=0
20000 IFYF<>0THEN20950
20005 MT=INT (1+RND (1)*20)
20010 ONMTGOSUB20100,20200,20300,20400,20500,20600,20700,20800,20900
20020 GOTO21000
20100 ZH=INT (1+RND (5)*4):ONZHGOSUB20110,20120,20130,20140
20105 IFWR>3THENWS=3
20106 GOTO20145
20110 WR=WR+INT (1+RND (2)*2):IFWR>5THENWR=1
20115 GOTO20145
20120 WR=WR-INT (1+RND (2)*2):IFWR<1THENWR=5
20125 GOTO20145
20130 WT=WT+INT (1+RND (2)*2):IFWT>5THENWT=1
20135 GOTO20145
20140 WT=WT-INT (1+RND (2)*2):IFWT<1THENWT=5
20145 MS=1:GOSUB1560:RETURN
20200 ZI=INT (-3+RND (7)*6):IFZI=0THEN20200
20210 MS=1
20220 CR=CR+ZI:BR=BR+ZI:DR=DR+ZI
20230 RETURN
20300 IFZN=1THENRETURN
20305 RA=RA+INT ((2.5-L)*20)
20310 MS=1:ZN=1:RETURN
20400 IFZO=1THENRETURN
20405 ZK=INT (5+RND (9)*15)
20410 ZL=INT (1+RND (5)*5)
20420 MA (ZL)=(INT (100*(MA (ZL)*((100+ZK)/100)))/100
20430 MS=1:ZO=1:RETURN
20500 IFZP=1THENRETURN
20505 ZK=INT (10+RND (9)*10)
20510 ZL=INT (1+RND (5)*5)
20520 PO (ZL)=(INT (100*(PO (ZL)*((100+ZK)/100)))/100
20530 MS=1:ZP=1:RETURN
20600 IFZO=1THENRETURN
20605 ZM=INT (5+RND (6)*15)
20610 FO=INT (FO+(FO*(ZM/100)))
20620 MS=1:ZO=1:RETURN
20700 IFZR=1THENRETURN
20710 ZS=INT (1+RND (7)*3)
20720 IFZS=1THENFF (4)=FF (4)+40:ZT=4
20730 IFZS=2THENFF (2)=FF (2)+INT (10+RND (1)*30)
20740 IFZS=3THENFORW=1TO5:FF (W)=FF (W)-5:NEXT W
20760 MS=1:ZR=1:RETURN
20800 YA=INT (1+RND (8)*3)
20810 MS=1:RETURN
20900 IF (DB<100)+(BD>0) THENRETURN
20910 YH=INT (DB/(10+RND (1)*20))
20920 DB=DB-YH:BD=BD+YH
20930 MS=1:RETURN
20950 MT=21:MS=1
20960 DB=DB-YG
20970 C=C+YG
20980 LF=LF+YF*25
20990 C=C-YF*25
20995 YF=0
21000 IF (WK<6)+(CD<2000) THEN21100
21010 W=INT (1+RND (1)*5):IFW<5THEN21100
21015 GOSUB90

```

MZ-80K NOTES & LISTINGS

```

21020 PRINT"H____A creditor has sued you for £";INT(CD*.2);"."
21025 PRINT"You have settled out of court and paid costs of £";INT(CD*.025)
21030 C=C-INT(CD*.2)
21035 C=C-INT(CD*.025):LE=LE+INT(CD*.025)
21037 CD=CD-INT(CD*.2)
21040 GOSUB 30
21100 IF(C=0)+(C>-1000)+(CD=0)THEN21200
21105 IF(C+DB)/CD>.8THEN 21200
21110 W=INT(1+RND(1)*3):IFW<2THEN 21200
21115 GOSUB90
21120 ONRBGOTO21140,21150
21130 PRINT"C__Your bank manager has asked you to      reduce your overdraft."
21135 RB=1:GOSUB30: GOTO21200
21140 PRINT"C____Your bank manager has warned you that he will have to ";
21142 PRINT"appoint a receiver to      liquidate your business if your ";
21144 PRINT"cash      position does not improve."
21146 RB=2:GOSUB30:GOTO21200
21150 RB=3
21151 FORW=1TO40:MUSIC"__C0__C0":NEXTW
21152 PRINT"C____The bank has put in a receiver to"
21153 PRINT"liquidate your business."
21154 PRINT"__The game is over."
21156 PRINT"____Press 1 to review your final position. Press 2 to start ";
21158 PRINT"a new game."
21160 GETG:IFG=0THEN21160
21162 IFG=1THEN1670
21164 IFG=2THEN10
21166 GOTO 21156
21200 IFRA<10THEN 21300
21210 IFRL=1THEN21260
21220 W=INT(1+RND(1)*100):IFW>RATHEN21300
21225 GOSUB90
21230 PRINT"H____Your workers have asked for their pay to be increased ";
21235 PRINT"to £";(L+L*.20);" per hour."
21240 PRINT"__Press 1 to agree, 2 to offer a lower      increase or 3 to refuse."
21242 GETG:IFG=0THEN21242
21243 IFG>3THEN21240
21244 IF G=1THENRA=-100:L=(L*1.20):RL=0 :GOTO21300
21246 IFG=2THENINPUT"__What hourly rate do you wish to offer? £";W
21247 IFG=3THENRL=1:GOTO21300
21248 IFW<LTHEN "That is a decrease!":GOTO21246
21250 IFW>L*1.1THENPRINT"Accepted.":RL=0:RA=-30:L=W:GOSUB30:GOTO21300
21251 PRINT"Your offer is refused.":RL=1:GOSUB30:GOTO21300
21260 GOSUB90
21261 PRINT"H____Your workers are on strike for more pay. "
21262 PRINT"They are asking for £";L+L*.2;" per hour."
21264 GOTO21240
21300 IFWK<5THEN21400
21310 W=INT(DB*.1+RND(1)*DB*.2)
21315 IFW>DBTHENW=0
21320 DB=DB-W:C=C+W
21400 W=INT(1+RND(1)*100)
21410 IFW<>50THEN21500
21415 IFRM=0THEN21500
21420 GOSUB90
21430 PRINT"H____A fire has destroyed all stocks of raw materials. _"
21440 FORW=1TO10:MUSIC"__C7f__G5R1":NEXTW
21445 W=INT(RM*.1+RND(1)*RM*.9)
21450 PRINT"The insurance company pays you ";
21455 IFW=0THENPRINT"nothing.":GOTO21460
21457 PRINT"£";W;"."
21458 UF=UF+RM-W
21460 RM=0:C=C+W
21470 FORW=1TO5:RM(W)=0:NEXTW

```

```

21480 GOSUB30
21500 GOSUB90
21512 PRINT"Results of week";WK:PRINT"-----";:IFWK>9THENPRINT"--";
21514 PRINT"_:":PRINTTAB(18);"Orders";TAB(30);"Stocks"
21515 DIMOP(5,12)
21516 PRINTTAB(18);"Received";TAB(30);"Sold"
21518 FORWW=1TO39:PRINT"--":NEXTWW
21519 PRINT"H"
21520 FORG=1TO5
21522 IFWK<12THENS(G,WK+1)=S(G,WK)+INT(RND(1)*30-15)
21525 S(G,WK)=INT(S(G,WK)*((100+FF(G))/100)):IFS(G,WK)<0THENS(G,WK)=0
21527 IFFH(G)>0THENFF(G)=FF(G)-FH(G):FH(G)=0
21530 OP(G,WK)=S(G,WK)
21532 FORWW=1TO4+G:PRINT:NEXTWW
21533 PRINTH$(G);TAB(14);S(G,WK);:GOSUB50
21534 PRINT"H":FORWW=1TO4+G:PRINT:NEXTWW
21535 IFS(G,WK)>=FG(G)THEN21540
21536 PC=PC+INT(S(G,WK)*(PC(G)/FG(G)))
21537 Q(G,WK+1)=FG(G)-S(G,WK)
21538 PC(G)=PC(G)-INT(S(G,WK)*(PC(G)/FG(G)))
21539 FG(G)=FG(G)-S(G,WK):GOTO21546
21540 S(G,WK)=FG(G)
21542 PC=PC+PC(G):O(G,WK+1)=O
21544 PC(G)=0:FG(G)=0
21546 PRINTTAB(26);S(G,WK);:GOSUB50
21560 S=S+INT(S(G,WK)*Q(G))
21570 DB=DB+INT(S(G,WK)*Q(G))
21575 IFFE=1THENGOSUB12720
21580 PRINT"H":NEXTG
21583 FG=PC(1)+PC(2)+PC(3)+PC(4)+PC(5)
21585 FORW=1TO10:PRINT:NEXTW
21590 FORW=1TO39:PRINT"--":NEXTW
21600 FT=FT+FO:C=C-FO
21610 IT=IT+INT((LC*CR/52)/100):C=C-INT((LC*CR/52)/100)
21620 IFC<0THENIB=IB+INT((-C*BR/52)/100)+1:C=C-INT((C*BR/52)/100)-1
21630 IFD>0THENID=ID+INT((D*DR/52)/100):C=C-INT((D*DR/52)/100)
21640 OC=FT+IT+IB+AV+BD+LE+UF+FC+UL
21650 NP=INT(S-(PC+OC)+ID):PF=NP
21660 GOSUB30
21665 IFWK=12THEN22000
21700 GOTO1600
22000 PRINT"C":FORW=1TO5
22010 MUSIC" _C_ _C_ D_ _D_ E_ _E_ F_ _F_ G_ _A_ _A_ BCDEFGAFCAB C _EC_ D_ _ED_ E_ _EF_ G_ "
22020 MUSIC" _EG_ A_ _EA_ B_ C_ "
22030 NEXTW
22040 PRINT"----- You have now completed twelve weeks of ";
22045 IFNP<0THENPRINT"trading. Your net profit is nil.":GOTO22130
22047 IFNP<0THENPRINT"trading. You have lost £";-(NP);".":GOTO22130
22050 PRINT"trading. Your return on share capital ";
22060 PRINT" in twelve weeks amounts to";INT(((NP/12)*52)/SC)*10000/100;
22070 PRINT"%":PRINT"over a whole year. That is ";
22080 IFINT(((NP/12)*52)/SC)*100<CR THEN PRINT"less than":GOTO22110
22090 IFINT(((NP/12)*52)/SC)*100=INT(CR) THEN PRINT"the same as":GOTO22110
22100 IF(((NP/12)*52)/SC)*100>CR THEN PRINT"more than"
22110 PRINT"the current interest rate on loan "
22120 PRINT"capital of";CR;"%."
22125 IFCR<(((NP/12)*52)/SC)*100THENPRINT"_ Well done !"
22130 RB=4:GOTO21154
23000 END

```

```

1 REM      Music Composer
2 REM
3 REM By Simon Jones 1984
4 REM
5 REM
6 POKE $2C4C,$00
7 POKE $2CAB,$00
8 POKE 7017,24
9 DIM M$(255),ED$(254)
10 GOTO 1500
11 PRINT "███":TEMPOS
15 MU$="+BOR+BOR+BOR+BO":TEMPO 7
20 PRINT "      Music Composer"
30 PRINT ""
40 PRINT "█"
50 PRINT TAB(9);"1.Read music from tape"
60 PRINT
70 PRINT TAB(9);"2.Write music to tape"
80 PRINT
90 PRINT TAB(9);"3.Create a passage"
100 PRINT
110 PRINT TAB(9);"4.Play music"
120 PRINT
130 PRINT TAB(9);"5.Edit a line"
140 PRINT
145 PRINT TAB(9);"6.End program"
146 PRINT
147 PRINT TAB(9);"7.Instructions"
150 PRINT "█"
160 PRINT TAB(9);"PLEASE SELECT OPTION";
170 USR($09B3)
180 GET T
190 ON T GOTO 200,400,600,800,970,1200,1630
195 GOTO 170
200 REM READ MUSIC FROM TAPE
210 PRINT "█"
220 PRINT "      OPTION No. 1"
225 PRINT ""
230 PRINT:PRINT
240 PRINT "> ENTER FILE NAME";
250 INPUT F$
252 IF F$="" THEN F$=""
255 IF LEN(F$)>16 GOTO 210
260 PRINT "█"
270 PRINT "> PRESS ANY KEY WHEN TAPE IS READY";
280 USR($09B3):PRINT "█"
290 OPEN/T F$
300 FOR A=1 TO 255
310 INPUT/T M$(A)
320 IF M$(A)="END" THEN 340
330 NEXT
340 CLOSE/T
350 GOTO 550
400 REM WRITE MUSIC TO TAPE
410 PRINT "█"
420 PRINT "      OPTION No. 2"
430 PRINT ""
440 PRINT:PRINT
445 IF M$(1)="" THEN PRINT "* No music to write":GOTO 740
450 PRINT "> PLEASE INPUT FILE NAME";
460 INPUT F$
465 IF LEN(F$)>16 GOTO 410
470 PRINT "█"

```

```

480 PRINT"> PRESS ANY KEY WHEN TAPE IS READY";
490 USR($09B3)
500 WOPEN/T F$
510 FOR A=1 TO 255
520 PRINT/T M$(A)
530 IF M$(A)="END" THEN 550
540 NEXT
550 CLOSE/T
555 GOTO 11
570 USR($09B3)
580 GOTO 11
600 PRINT"@"
610 PRINT"          OPTION No. 3"
620 PRINT"          "
630 PRINT:PRINT
640 INPUT"> HOW MANY LINES OF MUSIC ARE THERE";L$:IF L$=""THEN GOTO600
641 FOR K=1 TO LEN(L$)
642 F$=LEFT$(L$,K)
643 IF ASC(F$)<48 THEN MUSIC MU$:GOTO 600
644 IF ASC(F$)>57 THEN MUSIC MU$:GOTO 600
645 NEXT
646 L=VAL(L$)
650 IF L>255 THEN PRINT"Too many!";MUSIC MU$:GOTO 640
655 IF L<1 THEN GOTO 11
660 PRINT"@"
670 FOR A=4 TO 22
680 CURSOR 0,A:PRINT STRING$(" ",38)
690 NEXT:PRINT"*****"
700 FOR A=1 TO L
705 PRINT"> INPUT LINE ";A;
706 INPUT M$(A)
707 IF M$(A)="" THEN M$(A)=" "
710 NEXT A:M$(A)="END"
740 PRINT"Press any key when ready";
745 MUSIC MU$
750 USR($09B3)
760 GOTO 11
800 REM PLAY A PASSAGE
810 PRINT"@"
820 PRINT"          OPTION No. 4"
830 PRINT"          "
840 PRINT:PRINT
845 IF M$(1)="" THEN PRINT"* You have not yet created a passage":GOTO 740
850 PRINT"> PLEASE INPUT TEMPO (1-7)";
860 USR($09B3)
870 GET T
880 IF (T>7)+(T<1) THEN 860
885 CURSOR 0,7:PRINT"Press any key to play passage ";:USR($09B3)
890 TEMPO T
900 FOR A=1 TO 255
910 IF M$(A)="END" THEN 940
920 MUSIC M$(A)
930 NEXT
940 CURSOR 0,9:PRINT"> PRESS ANY KEY WHEN READY ";
950 USR($09B3)
960 GOTO 11
970 REM EDIT LINE
980 PRINT"@"
990 PRINT"          OPTION No. 5"
1000 PRINT"          "
1010 PRINT:PRINT
1020 IF M$(1)="" THEN PRINT" No music to edit":GOTO 740
1030 INPUT"> WHICH LINE DO YOU WISH TO EDIT";L$:IF L$="" THEN 980

```

```

1033 FOR K=1 TO LEN(L$)
1034 D$=LEFT$(L$,K)
1035 IF ASC(D$)<48 THEN MUSIC MU$:GOTO 980
1036 IF ASC(D$)>57 THEN MUSIC MU$:GOTO 980
1037 NEXT
1038 L=VAL(L$)
1039 L=INT(L)
1040 IF L>254 THEN PRINT"███ There's no such line███":MUSIC MU$:GOTO 1030
1041 IF L<1 THEN 11
1045 IF M$(L)="" THEN PRINT"███ There's no such line███":GOTO 1030
1050 PRINT"███";" LINE ";L;M$(L)
1060 PRINT"LINE ";L;
1062 INPUT ED$(L)
1063 IF ED$(L)="" THEN GOTO 980
1064 M$(L)=ED$(L)
1070 REM SET DATA MARKER IN ARRAY.
1080 FOR A=L TO 255
1090 IF M$(A)="" THEN M$(A)="END":GOTO 1120
1100 NEXT
1120 GOTO 740
1200 PRINT"█":CURSOR 0,22:PRINT"Exit program-(are you sure)";
1210 USR($09B3)
1220 GET T$
1230 IF T$="Y" THEN END
1240 IF T$="N" THEN 11
1250 GOTO 1210
1302 IF G$<>A$ THEN G$="!"
1305 NEXT G:RESTORE
1306 IF M$(A)="END" THEN END
1500 REM
1510 REM ** INSTRUCTIONS **
1520 REM
1530 PRINT"█":POKE 57346,0
1535 PRINT"███";TAB(11);"MUSIC COMPOSER"
1536 PRINT TAB(11);"-----"
1537 PRINT"███"
1540 PRINT TAB(2);"|"
1542 PRINT TAB(2);"|"
1544 PRINT TAB(2);"|" Would you like some instructions
1546 PRINT TAB(2);"|"
1548 PRINT TAB(2);"|" on this program?
1550 PRINT TAB(2);"|"
1552 PRINT TAB(2);"|"
1554 PRINT TAB(2);"|" PLEASE PRESS Y OR N ONLY.
1556 PRINT TAB(2);"|"
1558 PRINT TAB(2);"|"
1559 CURSOR 10,23:PRINT"(C) 1984 SIMON JONES"
1588 POKE 57346,1
1589 CURSOR 33,17:USR($09B3)
1590 GET T$
1600 IF T$="Y" THEN 1630
1610 IF T$="N" THEN 11
1620 TEMP07:MUSIC"+D3":GOTO 1589
1630 PRINT"█"
1640 PRINT TAB(10);"MUSIC COMPOSER"
1650 PRINT TAB(10);"-----"
1660 PRINT"███"
1670 PRINT" This program is about creating and"
1680 PRINT"█ editing music. The music you create"
1690 PRINT"███ can then be stored on tape so that"
1700 PRINT"███ you can play it back after this is"
1710 PRINT"███ turned off. You have 6 options and"
1720 PRINT"███ each one will be explained to you."
    
```

```

1730 PRINT"█"
1740 GOSUB 3500
1750 PRINT"█"
1760 PRINT"1. READ MUSIC FROM TAPE:--"
1765 PRINT"█"
1770 PRINT"█"
1780 PRINT"This option allows you to read in the"
1790 PRINT"music from tape that has already been"
1800 PRINT"written to tape. When you select this"
1810 PRINT"option, the music on tape will replace"
1820 PRINT"the present one.█ You will be asked for"
1830 PRINT"a file-name, this is the name that you"
1840 PRINT"gave to the file when it was written"
1850 PRINT"to tape. If you do not enter one and"
1860 PRINT"press CR or ENT then the computer will"
1870 PRINT"read in the first file it finds."
1880 PRINT"When the file has been read in, you"
1890 PRINT"will return to the menu."
1900 PRINT"█"
1910 GOSUB 3500
1920 PRINT"█"
1930 PRINT"2. WRITE MUSIC TO TAPE:--"
1940 PRINT"█"
1950 PRINT"█"
1960 PRINT"This option allows you to write the"
1970 PRINT"music that you have already created."
1980 PRINT"You will be asked for a file name, if"
1990 PRINT"you do not give one by pressing CR or"
2000 PRINT"ENT then that file will have no"
2010 PRINT"name.█ Once the computer has written"
2020 PRINT"the music to tape then you will return"
2030 PRINT"to the menu."
2040 PRINT"█"
2050 GOSUB 3500
2060 PRINT"█"
2070 PRINT"3. CREATE A PASSAGE:--"
2080 PRINT"█"
2090 PRINT"█"
2100 PRINT"This option allows you to create a"
2110 PRINT"passage of music in a series of lines."
2120 PRINT"Each line may contain up to 255"
2130 PRINT"characters. A character may be one of"
2140 PRINT"the following:--"
2150 PRINT"█1. Note (A,B,C,D,E,F,G)"
2160 PRINT"█2. Duration (1-9)"
2170 PRINT"█3. Type of note:█"
2180 PRINT" ( " or a + is a high note"
2190 PRINT" _ or a - is a low note )"
2200 PRINT"█"
2210 PRINT"If you press CR or ENT only then that"
2220 PRINT"line will be blank. If your line is"
2230 PRINT"over 255 characters, then that line"
2240 PRINT"will be shortened. So don't enter a "
2250 PRINT"line above 255 characters."
2260 PRINT"█"
2270 GOSUB 3500
2280 PRINT"█"

```

```

2290 PRINT"4. PLAY MUSIC-:"
2300 PRINT"
2310 PRINT"
2320 PRINT"This option allows you to play the music"
2330 PRINT"that has either been created or has been"
2340 PRINT"read in from tape. You will be asked to"
2350 PRINT"input a tempo ranging from 1-7."
2360 PRINT"Tempo 1 is the slowest and tempo 7 is"
2370 PRINT"the fastest. After you have selected"
2380 PRINT"the correct tempo you will be asked"
2390 PRINT"to press any key before it plays the"
2400 PRINT"passage."
2410 PRINT"
2420 GOSUB 3500
2430 PRINT"
2440 PRINT"5. EDIT A LINE-:"
2450 PRINT"
2460 PRINT"
2470 PRINT"This option will allow you to edit a"
2480 PRINT"line, if the music is not quite correct."
2490 PRINT"You will be asked for the line you wish"
2500 PRINT"to edit. If you ask for a line that"
2510 PRINT"does not exist then the computer will"
2520 PRINT"tell you so. When you have asked for"
2530 PRINT"valid line, the old line will appear"
2540 PRINT",and a blank one will appear below that"
2550 PRINT"where you will be able to enter the"
2560 PRINT"correct line,character by character."
2561 PRINT"If you press CR or ENT then that line"
2562 PRINT"will become blank."
2570 PRINT"
2580 GOSUB 3500
2590 PRINT"
2600 PRINT"6. END PROGRAM-:"
2610 PRINT"
2630 PRINT"This option should speak for itself."
2640 PRINT"However, there is a point worth"
2650 PRINT"remembering. If you should end the"
2660 PRINT"program and then run the program,"
2670 PRINT"then the music will be lost unless"
2680 PRINT"it has been written to tape."
2690 PRINT"So make sure you have written the"
2700 PRINT"music (if you still want it)"
2710 PRINT"before you exit the program."
2730 PRINT"If you have any problems, it might"
2740 PRINT"be a good idea to look at the user"
2750 PRINT>manual and try typing in some of the"
2760 PRINT"music in using the CREATE option."
2770 PRINT"If you are at anytime unsure of any"
2780 PRINT"thing, then you can look at these"
2790 PRINT"instructions again by selecting option"
2800 PRINT"No. 7"
2810 GOSUB 3500
2820 GOTO 11
3500 REM SUB
3510 PRINT TAB(6);"PRESS ANY KEY TO CONTINUE";
3520 USR($09B3)
3530 RETURN

```

```

10 REM
30 REM
35 REM  SUICIDE RUN
40 REM
45 REM  [C] SIMON JONES
50 REM
55 REM  16th APRIL 84
60 REM
65 REM
70 REM
75 REM
76 LIMIT 49152:TEMPO 6
80 DIM SA(255),HS(10),HS$(10)
81 POKE $2C4C,$00
82 POKE $2CAB,$00
85 FOR L=1 TO 10:HS$(L)="?????":HS(L)=1000:NEXT
90 A1$="8":A2$="2":A3$="4":A4$="6"
94 REM
95 REM =====
96 REM MACHINE CODE ROUTINES
97 REM =====
98 REM
100 DATA 62,166,17,40,208,6,40,18,19,16,252,17,191,211,6,41,18,19,16,252,17
110 DATA 40,208,6,23,14,40,13,19,32,252,18,16,247,17,39,208,6,23,14,40,19,13
120 DATA 32,252,18,16,247,201,46,20,62,255,38,14,17,0,208,6,40,18,19,16,252
125 DATA 17,192,211,6,40,18,19,16,252,17,39,208,6,23,14,40,19,13,32,252,18,16
127 DATA 247,17,0,208,6,23,14,40,19,13,32,252,18,16,247,61,37,32,206,45,32
129 DATA 199,201,-1
130 READ A
140 IF A=-1 THEN 176
150 POKE 49152+X,A
160 X=X+1
175 GOTO 130
176 GOTO 2000
177 REM =====
178 REM SET UP SCREEN DISPLAY
179 REM =====
180 PRINT "█":UP=0:LI=3:SC=0:TEMPO7
185 PRINT "█"
190 PRINT "█"
191 PRINT "█SCORE";SC;TAB(12);"HI-Score";HS;TAB(28);"00 SECONDS"
200USR(49152)
205POKE 53328,161:POKE 54207,162
210CURSOR 1,8:PRINT STRING$("█",20)
220CURSOR 20,17:PRINT STRING$("█",20)
230MAN=53370:M=207
240POKE MAN,M:R=202
250FOR K=1 TO T*10
260SA=INT(RND(1)*800+53400)
270IF PEEK(SA)<>0 THEN 260
290MN=MN+1:SA(MN)=SA:POKE SA,R
300NEXT:POKE 54206,0:MN=1
310FOR K=1 TO 3
320IF PEEK($E001)<>126 THEN MUSIC"+D3"
330POKE MA,0
340IF PEEK($E001)<>126 THEN MUSIC"R4"
350POKE MA,M
360NEXT
370IF PEEK($E001)<>126 THEN MUSIC"+A3"
375M=207:TI$="000000"
380POKE MA,M:POKE 4514,15:POKE 4513,200:USR(68):USR(71)
385IF TI$>"000015" THEN IF M=207 THEN GOSUB 1091:GOTO 680
386IF TI$>"000015" THEN GOSUB 1035:GOTO 680
387CURSOR 28,0:PRINT RIGHT$(TI$,2);" SECONDS"

```

```

390 GET T$:IF T$="" THEN 380
395 D=1:DE=0
396 REM =====
397 REM MAIN LOOP
398 REM =====
399 REM
400 GET T$
405 IF TI$>"000015" THEN IF M=207 THEN GOSUB 1091:GOTO 680
406 IF TI$>"000015" THEN GOSUB 1035:GOTO 680
407 CURSOR 28,0:PRINT RIGHT$(TI$,2)
420 IF T$=A1$ THEN D=-40
430 IF T$=A2$ THEN D=40
440 IF T$=A3$ THEN D=-1
450 IF T$=A4$ THEN D=1
455 IF PEEK($E001)=126 THEN 1580
460 IF PEEK(MA+D)>0 THEN 500
470 MA=MA+D:POKE MA,M:POKE MA-D,0
480 USR(68):USR(71)
490 GOTO 400
500 IF PEEK(MA+D)=161 THEN 570
510 IF PEEK(MA+D)=162 THEN 590
511 IF M=206 THEN GOSUB 1035:GOTO 621
515 GOTO 621
516 REM
517 REM =====
518 REM INCREASE SCORE
519 REM =====
520 FOR S1=1 TO 10:FOR S2=255 TO 0 STEP-55
530 POKE 4514,S1:POKE 4513,S2:USR(68)
535 SC=SC+T$2:PRINT "SCORE":SC
536 IF UP>10 THEN UP=0:GOTO 1100
540 NEXT:NEXT:USR(71)
550 POKE MAN,M:UP=UP+1
560 TI$=B$:GOTO 380
561 REM
562 REM =====
563 REM CHANGE FACE
564 REM =====
565 REM
570 IF PEEK(MA)=206 THEN M=207:TI$="000000":B$=TI$:GOTO520
580 GOTO 621
590 IF PEEK(MA)=207 THEN M=206:B$=TI$:GOTO520
595 IF PEEK(MA)=206 THEN GOSUB 1035:GOTO 621
610 REM
611 REM =====
612 REM HIT AN OBSTACLE
613 REM =====
614 REM
621 IF AG=1 THEN GOTO 680
625 FOR RE=1 TO 50
630 RN=255-INT(RND(1)*15)
640 POKE MA-40,RN:POKE MA+40,RN:POKE MA-1,RN
642 POKE MA+1,RN:POKE MA-41,RN:POKE MA+41,RN
645 POKE MA-39,RN:POKE MA+39,RN:POKE MA,RN
650 POKE 4514,INT(RND(1)*255):USR(68)
670 USR(68):NEXT:USR(71)
680 AG=0:FOR P=2 TO 23
690 CURSOR 1,P:PRINT STRING$( " ",38)
700 NEXT:LI=LI-1:IF LI<0 THEN 800
710 CURSOR 17,9:PRINT"LEFT:";LI
730 FOR K=1 TO 13
740 CURSOR 18,11:PRINT"READY"
750 FOR D1=1 TO 100:NEXT D1

```



```

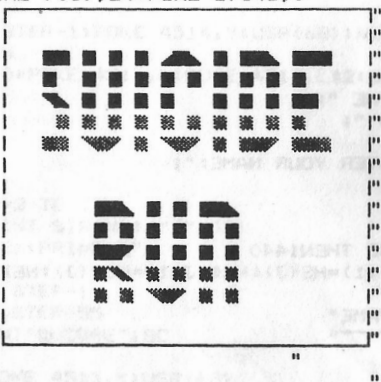
1200 NEXT: NEXT:USR(71)
1210 FOR J=1 TO 20:FOR K=255 TO 0 STEP-55:POKE 4514,J:POKE 4513,K:USR(68)
1215 SC=SC+T*2:PRINT"SCORE";SC
1216 USR(71)
1220 NEXT: NEXT:USR(71)
1225 IF T>12 THEN 1259
1230 S$="Now try level"
1231 FOR K1=1 TO LEN(S$)
1232 FOR B=3 TO 0 STEP-1:POKE 4514,B:USR(71):USR(68):NEXT
1233 CURSOR 13,17:PRINT LEFT$(S$,K1);
1234 NEXT K1:T=T+1:PRINT T
1235 FOR J=1 TO 12
1236 CURSOR 13,17:PRINT S$;T
1237 FOR KJ=1 TO 70:NEXT KJ
1238 CURSOR 13,17:PRINT"
1239 FOR KJ=1 TO 70:NEXT KJ
1240 NEXT J
1249 FOR P=2 TO 20
1250 CURSOR 2,P:PRINT STRING$( " ",35)
1251 NEXT:GOTO 190
1252 REM
1253 REM =====
1254 REM SAVED TOWN
1255 REM =====
1256 REM
1259 PRINT"
1260 TEMPO 6
1270 USR(49152)
1275 CURSOR 12,8:PRINT"CONGRATULATIONS!"
1280 CURSOR 8,10:PRINT"WELL DONE, YOU HAVE DEFUSED"
1290 CURSOR 9,12:PRINT"ALL THE BOMBS AND SAVED"
1300 CURSOR 14,14:PRINT" THE TOWN."
1301 M1$="+D3+E3+C3^A5^B3^G5^D3^E3^C3A5B3G5"
1302 M2$="D3E3C3_A5_B3_A3_#G3_G5R5G0B0^D0^G5D3#D3"
1303 M3$="E3^C5E3^C5E3^C7+C3+D3+#D3+E3+C3+D3+E5^B3+D5+C8"
1304 M4$="D3#D3E3^C5E3^C5E3^C8^A3^G3^#F3^A3+C3+E5+D3+C3^A3"
1305 M5$="+D8D3#D3E3^C5E3^C5E3^C8+C3+D3+#D3+E3"
1306 M6$="+C3+D3+E5^B3+D5+C8R0"
1307 M7$="+C3+D3+E3+C3+D3+E5+C3+D3+C3+E3+C3+D3+E5+C3+D3+C3"
1308 M8$="+E3+C3+D3+E5^B3+D5+C4"
1309 MUSIC M1$,M2$,M3$,M4$,M5$,M6$,M7$,M8$
1319 REM
1320 REM =====
1330 REM HALL OF FAME
1340 REM =====
1341 REM
1350 PRINT"
1351 PRINT"
1355 PRINTTAB(7);"YOU HAVE ";
1356 IF SC>HS(1) THEN PRINT"THE ";
1357 IF SC<HS(1) THEN PRINT"A ";
1358 PRINT"HIGH-SCORE!!"
1360 PRINT TAB(5);"PLEASE ENTER YOUR NAME:";
1370 INPUT N$
1375 IF N$="" THEN 1351
1380 N$=LEFT$(N$,8)
1390 FOR I=1 TO 10:IF HS(I)>SC THEN 1440
1395 FOR J=9 TO I STEP-1:HS(J+1)=HS(J):HS$(J+1)=HS$(J):NEXT:HS(I)=SC:HS$(I)=N$
1400 PRINT"POKE 57346,0"
1405 PRINT TAB(13);"HALL OF FAME"
1410 PRINT TAB(13);"
1411 PRINT"
1420 FOR I=1 TO 9.

```

```

1430 PRINT TAB(11);I;". ";TAB(13);HS(I);TAB(21);HS*(I)
1435 PRINT
1440 NEXT I:PRINTTAB(10);10;". ";TAB(13);HS(10);TAB(21);HS*(10):POKE 57346,1
1455 HS=HS(1)
1465 FOR DELAY=1 TO 1000:NEXT
1470 FOR DELAY=1 TO 2000
1475 GET T$:IF T$<>" " THEN 2340
1480 NEXT
1485 GOTO 2340
1500 REM
1501 REM =====
1502 REM BONUS TITLE
1503 REM =====
1504 REM
1509 PRINT "#####"
1510 PRINTTAB(5);" "
1515 PRINTTAB(5);" "
1520 PRINTTAB(5);" "
1525 PRINTTAB(5);" "
1530 PRINTTAB(5);" "
1535 PRINTTAB(5);" "
1540 PRINTTAB(5);" "
1545 PRINTTAB(5);" "
1550 PRINTTAB(5);" "
1560 GOTO 1170
1570 END
1575 REM
1576 REM =====
1577 REM BREAK KEY PRESSED
1578 REM =====
1579 REM
1580 PRINT "#####BREAK"
1590 PRINT"RESTART?";
1600 USR($09B3)
1610 GET T$
1620 IF T$="Y" THEN 2000
1630 IF T$="N" THEN PRINT"R":END
1640 GOTO 1600
1700 END
1995 REM
1996 REM =====
1997 REM OPENING TITLE
1998 REM =====
1999 REM
2000 PRINT"#####":POKE 7017,24:POKE 57346,0
2005 PRINT"
2010 PRINT"
2020 PRINT"
2030 PRINT"
2040 PRINT"
2050 PRINT"
2060 PRINT"
2070 PRINT"
2080 PRINT"
2090 PRINT"
2100 PRINT"
2110 PRINT"
2120 PRINT"
2130 PRINT"
2140 PRINT"
2150 PRINT"
2160 PRINT"
2170 PRINT"

```



```

2175 CURSOR 0,24:PRINT"#####":R=50
2180 POKE 57346,1
2190 FOR K=1 TO 35
2200 FOR J=1 TO 30:NEXT J
2210 PRINT"##"
2220 NEXT
2230 S$="By Simon Jones 1984 "
2240 FOR K=1 TO LEN(S$)+9
2250 CURSOR 40-K,21:PRINT LEFT$(S$,K)
2260 R=R-1:POKE 4514,R:USR(68)
2270 NEXT:USR(71)
2280 USR(49202)
2290 CURSOR 0,23
2300 FOR K=1 TO 35
2310 FOR L=1 TO 30:NEXT L
2320 PRINT
2330 NEXT
2335 REM
2336 REM ----
2337 REM MENU
2338 REM ----
2339 REM
2340 PRINT"#####":TI$="000000"
2350 PRINT TAB(13);"Suicide Run"
2360 PRINT TAB(13);"-----"
2370 PRINT"##"
2380 PRINT TAB(12);"1..Instructions##"
2390 PRINT TAB(12);"2..Define keys##"
2400 PRINT TAB(12);"3..Start game##"
2410 PRINT TAB(12);"SELECT OPTION " ;
2420 GET T
2430 ON T GOTO 2460,2850,960
2440 IF TI$>"000030" THEN 1400
2450 GOTO 2430
2456 REM
2457 REM -----
2458 REM INSTRUCTIONS
2459 REM -----
2460 REM
2465 PRINT"##"
2470 PRINT TAB(13);"Suicide Run"
2480 PRINT TAB(13);"-----"
2490 PRINT"##Your aim to to dispose the bombs which"
2505 PRINT"terrorise's the town. You must first"
2510 PRINT"collect a bomb from the unit at the"
2520 PRINT"bottom of the screen, which looks like"
2525 PRINT"'|'. Then you must take it back up to"
2530 PRINT"the top of the screen and go into the"
2540 PRINT"unit which looks like this '2'."
2550 PRINT"However, you have only 16 seconds to"
2555 PRINT"do this, before the bomb explodes.If"
2560 PRINT"the bomb explodes or you hit anything"
2565 PRINT"then you will lose a life. Bonus's"
2600 PRINT"are awarded throughout the game."
2610 PRINT"##"
2620 PRINT TAB(7);"HIT SPACE BAR WHEN READY"
2630 GET T$:IF T$=" " THEN 2630
2640 PRINT"##"
2650 PRINT" Normal keys are as follows-:"
2660 PRINT"##"

```

2670 PRINT TAB(10);

2680 PRINT TAB(10);

2690 PRINT TAB(10);

2700 PRINT TAB(10);

2710 PRINT TAB(10);

2720 PRINT TAB(10);

2730 PRINT TAB(10);

2740 PRINT TAB(10);

2750 PRINT TAB(10);

2760 PRINT TAB(10);

2770 PRINT TAB(10);

2780 PRINT"##";

2790 PRINT"### These keys will give you";

2800 USR(49202);

2810 GET T\$;

2820 IF T\$="Y" THEN GOTO 2830;

2830 IF T\$="N" THEN GOTO 2840;

2840 GOTO 2800;

2845 REM

2848 REM -----

2849 REM CONTROL KEY DEFINER

2848 REM -----

2849 REM

2850 PRINT"#####";

2850 PRINT

2850 PRINT

2850 PRINT

2850 CURSOR 10,4

2850 PRINT"ENTER UP KEY";USR(49202);

2850 CURSOR 10,11

2850 PRINT"ENTER DOWN KEY";USR(49202);

2850 CURSOR 10,12

2850 PRINT"ENTER LEFT KEY";USR(49202);

2850 CURSOR 10,13

2850 PRINT"ENTER RIGHT KEY";USR(49202);

2850 GOTO 2840

```

2670 PRINT TAB(10);"
2680 PRINT TAB(10);"
2690 PRINT TAB(10);"
2700 PRINT TAB(10);"
2710 PRINT TAB(10);"
2720 PRINT TAB(10);"
2730 PRINT TAB(10);"
2740 PRINT TAB(10);"
2750 PRINT TAB(10);"
2760 PRINT TAB(10);"
2770 PRINT TAB(10);"
2780 PRINT"  "
2790 PRINT"Are these keys alright with you?";
2800 USR($09B3)
2810 GET T$
2820 IF T$="Y" THEN A1$="B":A2$="2":A3$="4":A4$="6":GOTO 2340
2830 IF T$="N" THEN 2850
2840 GOTO 2800
2845 REM
2846 REM =====
2847 REM CONTROL KEY DEFINER
2848 REM =====
2849 REM
2850 PRINT"  "
2860 PRINT"          KEY DEFINER"
2870 PRINT"
2880 PRINT"  "
2885 CURSOR 10,9
2890 PRINT"ENTER UP KEY";:USR($09B3):GET A1$:PRINT A1$:MUSIC"-D2"
2905 CURSOR 10,11
2910 PRINT"ENTER DOWN KEY";:USR($09B3):GET A2$:PRINT A2$:MUSIC"-D2"
2915 CURSOR 10,13
3000 PRINT"ENTER LEFT KEY";:USR($09B3):GET A3$:PRINT A3$:MUSIC"-D2"
3005 CURSOR 10,15
3010 PRINT"ENTER RIGHT KEY";:USR($09B3):GET A4$:PRINT A4$:MUSIC"-D2"
3020 GOTO 2340

```



```

1 PRINT"█"
2 PRINT"██████"
3 PRINT"██████"
4 PRINT"██████"
5 PRINT"██████"          FLYING
6 PRINT"██████"
7 PRINT"██████"          MISSION
8 PRINT"██████"
9 PRINT"██████"
10 PRINT"██████"        Copyright
11 PRINT"██████"
12 PRINT"██████"        by
13 PRINT"██████"
14 PRINT"██████"        Letizia Bizzarri
15 PRINT"██████"
16 PRINT"██████"        Via Lago Isoletta 31
17 PRINT"██████"
18 PRINT"██████"        65100 Pescara
19 PRINT"██████"
20 PRINT"██████"        ITALY
21 PRINT"██████"
22 M$="C+D+E+F+G+A+BCDEFGAB-C-D-E-F-G-A--":MUSICM$,M$
23 PRINT"██████████████████Do you want instruction (Y/N)"
24 GETR$:IFR$=""THEN24
25 IFR$="Y"THENGOSUB10000
30 REM
40 PRINT"█"
50 GOSUBB000
60 FORA=0TO20
70 FORB=0TO31
80 CURSORB,A:PRINT"███"
90 IFCHARACTER$(B+3,A)<>" "THEN GOTO270
100 GETA$:IFA$="F"THENGOSUB160
110 FORP=0TO9:NEXTP
120 NEXT B:CURSORB,A:PRINT"  ":NEXT A
140 CURSOR11,31:PRINT"Press S for another fly"
150 GETR$:IFR$=""THEN150
151 IFR$="S"THEN30
152 PRINT"██████████████████OK, bye !!!██████████████████"
155 END
160 B1=B+3:IFB1>31THENB1=0
170 FORX=A+1TO20
180 CURSORB1,X-1:PRINT"  "
190 CURSORB1,X:PRINT"$"
200 B=B+1
210 IFB>31THENCURSORB,A:PRINT"  ":A=A+1:B=0
220 CURSORB,A:PRINT"███":IFCHARACTER$(B+3,A)<>" "THENGOTO270
230 POKE4514,X:USR(68)
240 NEXT X
250 CURSORB1,X-1:PRINT"  "
260 USR(71):RETURN
270 FORC=AT019:CURSORB+1,C:PRINT"  "
280 CURSORB+1,C+1:PRINT"███"
290 POKE4514,C:USR(68):NEXTC :USR(71)
300 CURSOR11,1 :PRINT"Destroyed!!!":CURSOR11,5 :PRINT"Points":(A*10)+B
310 GOTO140
8000 FORK=1TO31
8010 R=INT(6*RND(1))+2:J=INT(10*RND(1))+10
8020 CURSORK,J:PRINT"█"
8030 FORJ=J+1TO20
8040 CURSORK,J:PRINT"□"
8050 NEXT J:NEXTK
8060 FORA=0TO31

```

```

1070 - B070 CURSORA,21:PRINT"";:NEXT:RETURN
1080 - 10000 PRINT"█ FLYING MISSION
1090 - 10010 PRINT:PRINT
1100 10020 PRINT"Now on the screen you'll see the skyline
1110 10030 PRINT"█of a city.
1120 10040 PRINT"You are the pilot of the aircraft that
1130 10050 PRINT"has to destroy the enemy city.
1140 10060 PRINT"You'll fly rapidly over the city, and
1150 10070 PRINT"each time your aircraft will fly at
1160 10080 PRINT"a lower level. You have to destroy
1170 10090 PRINT"the skyscrapes, otherwise your aircraft
1180 10100 PRINT"will crash against them.
1190 10110 PRINT"To shoot, press < F >
1200 10120 PRINT:PRINT:PRINT
1210 10130 PRINT"ANY KEY TO GO
1220 10140 GETR$:IFR$=""THEN10140
1230 10150 RETURN
1240
1250
1260
1270
1280
1290
1300
1310
1320
1330
1340
1350
1360
1370
1380
1390
1400
1410
1420
1430
1440
1450
1460
1470
1480
1490
1500
1510
1520
1530
1540
1550
1560
1570
1580
1590
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740
1750
1760
1770
1780
1790
1800
1810
1820
1830
1840
1850
1860
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000
2010
2020
2030
2040
2050
2060
2070
2080
2090
2100
2110
2120
2130
2140
2150
2160
2170
2180
2190
2200
2210
2220
2230
2240
2250
2260
2270
2280
2290
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2400
2410
2420
2430
2440
2450
2460
2470
2480
2490
2500
2510
2520
2530
2540
2550
2560
2570
2580
2590
2600
2610
2620
2630
2640
2650
2660
2670
2680
2690
2700
2710
2720
2730
2740
2750
2760
2770
2780
2790
2800
2810
2820
2830
2840
2850
2860
2870
2880
2890
2900
2910
2920
2930
2940
2950
2960
2970
2980
2990
3000
3010
3020
3030
3040
3050
3060
3070
3080
3090
3100
3110
3120
3130
3140
3150
3160
3170
3180
3190
3200
3210
3220
3230
3240
3250
3260
3270
3280
3290
3300
3310
3320
3330
3340
3350
3360
3370
3380
3390
3400
3410
3420
3430
3440
3450
3460
3470
3480
3490
3500
3510
3520
3530
3540
3550
3560
3570
3580
3590
3600
3610
3620
3630
3640
3650
3660
3670
3680
3690
3700
3710
3720
3730
3740
3750
3760
3770
3780
3790
3800
3810
3820
3830
3840
3850
3860
3870
3880
3890
3900
3910
3920
3930
3940
3950
3960
3970
3980
3990
4000
4010
4020
4030
4040
4050
4060
4070
4080
4090
4100
4110
4120
4130
4140
4150
4160
4170
4180
4190
4200
4210
4220
4230
4240
4250
4260
4270
4280
4290
4300
4310
4320
4330
4340
4350
4360
4370
4380
4390
4400
4410
4420
4430
4440
4450
4460
4470
4480
4490
4500
4510
4520
4530
4540
4550
4560
4570
4580
4590
4600
4610
4620
4630
4640
4650
4660
4670
4680
4690
4700
4710
4720
4730
4740
4750
4760
4770
4780
4790
4800
4810
4820
4830
4840
4850
4860
4870
4880
4890
4900
4910
4920
4930
4940
4950
4960
4970
4980
4990
5000
5010
5020
5030
5040
5050
5060
5070
5080
5090
5100
5110
5120
5130
5140
5150
5160
5170
5180
5190
5200
5210
5220
5230
5240
5250
5260
5270
5280
5290
5300
5310
5320
5330
5340
5350
5360
5370
5380
5390
5400
5410
5420
5430
5440
5450
5460
5470
5480
5490
5500
5510
5520
5530
5540
5550
5560
5570
5580
5590
5600
5610
5620
5630
5640
5650
5660
5670
5680
5690
5700
5710
5720
5730
5740
5750
5760
5770
5780
5790
5800
5810
5820
5830
5840
5850
5860
5870
5880
5890
5900
5910
5920
5930
5940
5950
5960
5970
5980
5990
6000
6010
6020
6030
6040
6050
6060
6070
6080
6090
6100
6110
6120
6130
6140
6150
6160
6170
6180
6190
6200
6210
6220
6230
6240
6250
6260
6270
6280
6290
6300
6310
6320
6330
6340
6350
6360
6370
6380
6390
6400
6410
6420
6430
6440
6450
6460
6470
6480
6490
6500
6510
6520
6530
6540
6550
6560
6570
6580
6590
6600
6610
6620
6630
6640
6650
6660
6670
6680
6690
6700
6710
6720
6730
6740
6750
6760
6770
6780
6790
6800
6810
6820
6830
6840
6850
6860
6870
6880
6890
6900
6910
6920
6930
6940
6950
6960
6970
6980
6990
7000
7010
7020
7030
7040
7050
7060
7070
7080
7090
7100
7110
7120
7130
7140
7150
7160
7170
7180
7190
7200
7210
7220
7230
7240
7250
7260
7270
7280
7290
7300
7310
7320
7330
7340
7350
7360
7370
7380
7390
7400
7410
7420
7430
7440
7450
7460
7470
7480
7490
7500
7510
7520
7530
7540
7550
7560
7570
7580
7590
7600
7610
7620
7630
7640
7650
7660
7670
7680
7690
7700
7710
7720
7730
7740
7750
7760
7770
7780
7790
7800
7810
7820
7830
7840
7850
7860
7870
7880
7890
7900
7910
7920
7930
7940
7950
7960
7970
7980
7990
8000
8010
8020
8030
8040
8050
8060
8070
8080
8090
8100
8110
8120
8130
8140
8150
8160
8170
8180
8190
8200
8210
8220
8230
8240
8250
8260
8270
8280
8290
8300
8310
8320
8330
8340
8350
8360
8370
8380
8390
8400
8410
8420
8430
8440
8450
8460
8470
8480
8490
8500
8510
8520
8530
8540
8550
8560
8570
8580
8590
8600
8610
8620
8630
8640
8650
8660
8670
8680
8690
8700
8710
8720
8730
8740
8750
8760
8770
8780
8790
8800
8810
8820
8830
8840
8850
8860
8870
8880
8890
8900
8910
8920
8930
8940
8950
8960
8970
8980
8990
9000
9010
9020
9030
9040
9050
9060
9070
9080
9090
9100
9110
9120
9130
9140
9150
9160
9170
9180
9190
9200
9210
9220
9230
9240
9250
9260
9270
9280
9290
9300
9310
9320
9330
9340
9350
9360
9370
9380
9390
9400
9410
9420
9430
9440
9450
9460
9470
9480
9490
9500
9510
9520
9530
9540
9550
9560
9570
9580
9590
9600
9610
9620
9630
9640
9650
9660
9670
9680
9690
9700
9710
9720
9730
9740
9750
9760
9770
9780
9790
9800
9810
9820
9830
9840
9850
9860
9870
9880
9890
9900
9910
9920
9930
9940
9950
9960
9970
9980
9990

```


TAPE LOADER from B.L. Page

Here is a listing of a BASIC Program Loader; it controls the tape fast-forward time. My printer cannot cope with Sharp graphics characters so I have used:-

CS\$ for clear screen HM\$ for home UP\$ for up cursor

```

1000 REM TaPe Loader. B.L.Page JAN 85
1001 REM OP-5925
1500 GOSUB 9000
2000 REM Main Program
2010 PRINT CS$;"Files on TaPe -----"
2020 PRINT
2030 GOSUB 3000
2040 PRINT
2050 USR(BEEP)
2060 INPUT "Select ";I
2070 IF (I<0)+(I>P) GOTO 2060
2080 USR(BEEP)
2090 PRINT CS$;"Please Press fast forward"
2100 GOSUB 3400
2110 D=DY(I):REM Set delay
2115 PRINT CS$;"Please wait";D;" seconds"
2120 GOSUB 3200
2130 GOSUB 4400
2140 PRINT
2150 PRINT"LOAD":
2160 GOSUB 3600
2170 PRINT FI$(I);
2180 GOSUB 3600
2190 USR(BEEP)
2200 PRINT HM$;"Press stop and then return"
2210 END
3000 REM Get time in seconds modulo 10 minutes
3010 TM#=TI#
3020 TI#=60*VAL(MID$(TM#,4,1))+VAL(RIGHT$(TM#,2))
3030 RETURN
3200 REM Delay for D seconds
3210 GOSUB 3000
3220 FT=TI+D
3230 IF FT>=699 THEN FT=FT-699
3240 GOSUB 3000
3250 IF TI<>FT GOTO 3240
3260 RETURN
3300 REM Sense taPe motor
3310 C=INT(PEEK(57346)/16):R=C-2*INT(C/2)
3320 RETURN
3400 REM Wait for taPe motor to start
3410 GOSUB 3300
3420 IF R=0 GOTO 3410
3460 RETURN
3600 REM Print double quote
3610 POKE 6350,0
3620 PRINT CHR$(34);
3630 POKE 6350,34
3640 RETURN
3800 REM Read Data
3810 RESTORE

```

MZ-80B NOTES & LISTINGS

```

3820 P=-1
3830 P=P+1
3835 READ FI#
3840 IF FI#="*" THEN RETURN
3850 FI#(P)=FI#
3860 READ DY(P),ST,ED
3870 PRINT P;TAB(4);FI#;TAB(20);ST;TAB(30);ED
3880 GOTO 3830
3999 RETURN
4000 REM Resave updated loader
4010 PRINT CS#;"Looking for SYNC";

```

```

4020 GOSUB 3300
4030 IF R=0 THEN PRINT", Please press PLAY"
4040 GOSUB 3400
4050 OPEN"SYNC":CLOSE
4060 PRINT
4090 PRINT"POKE AR,1:SAVE";
4090 GOSUB 3600
4100 PRINT"LOADER.SB ";
4110 GOSUB 3600
4120 PRINT":POKE AR,0";
4130 PRINT UP#;UP#
4140 END
4200 REM Write marker for new tape
4210 WOPEN"SYNC":CLOSE
4220 END
4400 REM To 991e cassette motor
4410 POKE 57347,6:POKE 57347,7
4420 RETURN
9000 REM Initialise
9010 CS#="C":NM#="C":UP#="C"
9020 DIM FI#(25),DY(25)
9030 BEEP=62
9040 AR=10682 REM Address of auto run flag
9050 RETURN
9500 REM Data
9505 REM Format - filename,delay,start,end
9506 REM Delay is fast forward time in seconds
9507 REM GOTO 4000 to resave
9508 REM GOTO 4200 to write marker
9510 DATA FILE1,10,10,20
9520 DATA FILE2,20,30,40
9530 DATA FILE3,30,50,60
9540 DATA FILE4,40,70,80
9550 DATA FILE5,50,90,100
9999 DATA *

```

```

10 REM STEREOSCOPIC SLICING (L+R FRAMES ON SEPARATE SCREENS)
20 REM Arens, 5 Heidepark, Wageningen Hoog, NL-6705 AB Holland
30 DIMA1(53,3),A2(101,3),A3(101,3),X3(2,3),AA(3)
40 DIMB1(101),E2(50,2),B4(3,3),B5(3,2),B8(7,2),B9(10)
50 DATA -1,-1,5,-1,-1,5,1,2,3,2,2,18,1,2,1,3,2,3,1,2,1
60 CONSOLEC80,S0,24,N:GRAPH00,I1,C,12,C
70 GOSUB1580
80 READA1(1,1),A1(1,2),A1(2,1),A1(2,2),A1(3,1),A1(3,2)
90 READA2(1,1),A2(1,2),A2(1,3),A3(1,1),A3(1,2),A3(1,3)
100 READB5(1,1),B5(1,2),B5(2,1),B5(2,2),B5(3,1),B5(3,2),P,E,G
110 V1=.5/B9(9)
120 FORG=1TO7
130 B8(G,1)=COS(B9(G)*.0174533)
140 B8(G,2)=SIN(B9(G)*.0174533)
150 NEXTG
160 FORG=1TO101
170 B1(G)=G
180 NEXTG
190 N=23
200 REM INPUT DATA
210 FORG=4TON
220 A1(G,1)=RND(1)
230 A1(G,2)=RND(1)
240 A1(G,3)=RND(1)/B9(9)
250 NEXTG
260 REM GENERATE DELAUNAY TRIANGULATION
270 FORG=4TON
280 M=0
290 FORH=1TOP
300 D=A3(H,3)-(A1(G,1)-A3(H,1))^2
310 IFD<OTHEN530
320 D=D+(A1(G,2)-A3(H,2))^2
330 IFD<OTHEN530
340 E=E-1
350 B1(E)=H
360 FORI=1TO3
370 IFM<1THEN490
380 L=M
390 FORJ=1TOL
400 IF(A2(H,B5(I,1))<>B2(J,1))+A2(H,B5(I,2))<>B2(J,2) THEN480
410 M=M-1
420 IFJ>MTHEN520
430 FORK=JТОM
440 B2(K,1)=B2(K+1,1)
450 B2(K,2)=B2(K+1,2)
460 NEXTK
470 GOTOS20
480 NEXTJ
490 M=M+1
500 B2(M,1)=A2(H,B5(I,1))
510 B2(M,2)=A2(H,B5(I,2))
520 NEXTI
530 NEXTH
540 FORI=1ТОM
550 K=B1(E)
560 E=E+1
570 FORJ=1ТО2
580 L=B2(I,J)
590 B4(J,1)=A1(L,1)-A1(G,1)
600 B4(J,2)=A1(L,2)-A1(G,2)
610 B4(J,3)=B4(J,1)*(A1(L,1)+A1(G,1))/2
620 B4(J,3)=B4(J,2)*(A1(L,2)+A1(G,2))/2+B4(J,3)
630 NEXTJ
640 D=B4(1,1)*B4(2,2)-B4(1,2)*B4(2,1)
650 A3(K,1)=(B4(1,3)*B4(2,2)-B4(2,3)*B4(1,2))/D
660 A3(K,2)=(B4(1,1)*B4(2,3)-B4(2,1)*B4(1,3))/D
670 A3(K,3)=(A1(G,1)-A3(K,1))^2+(A1(G,2)-A3(K,2))^2
680 A2(K,1)=B2(1,1)
690 A2(K,2)=B2(1,2)
700 A2(K,3)=G
710 NEXTI
720 P=P+2
730 NEXTG
740 IFC=1THEN830
750 REM ROTATE DATA SET
760 FORG=4TON
770 Z=(A1(G,3)-V1)*B8(G,1)-(A1(G,1)-.5)*B8(G,2)+V1
780 A1(G,1)=(A1(G,1)-.5)*B8(G,1)+(A1(G,3)-V1)*B8(G,2)+.5
790 A1(G,3)=(Z-V1)*B8(G+1,1)+(A1(G,2)-.5)*B8(G+1,2)+V1
800 A1(G,2)=(A1(G,2)-.5)*B8(G+1,1)-(Z-V1)*B8(G+1,2)+.5
810 NEXTG
820 REM SLICE DATA SET
830 FORH=1TOP

```

```

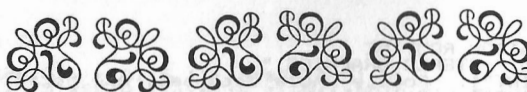
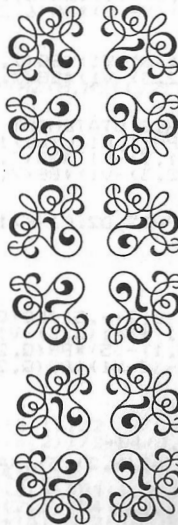
440 IF (A2(H,1)<4)+(A3(H,3)>1) THEN1330
450 GOSUB1450:T=MAX
460 GOSUB1450:S=MIN
470 R=-.866
480 FORI=1TO100
490 R=R+B9(8)
500 IF (T<R)+(S>R) THEN1320
510 Y=1
520 U=0
530 U=U+1
540 F=(R-A1(A2(H,B5(U,1)),3))/(A1(A2(H,B5(U,2)),3)-A1(A2(H,B5(U,1)),3))
550 IF (F<0)+(F>1) THEN990
560 X3(Y,1)=A1(A2(H,B5(U,1)),1)+(A1(A2(H,B5(U,2)),1)-A1(A2(H,B5(U,1)),1))*F
570 X3(Y,2)=A1(A2(H,B5(U,1)),2)+(A1(A2(H,B5(U,2)),2)-A1(A2(H,B5(U,1)),2))*F
580 Y=Y+1
590 IFY<3THEN930
600 X3(1,3)=R
610 X3(2,3)=R
620 IFC=1THEN1210
630 REM REVERSE ROTATE INTERSECTION TRACES
640 FORG=1TO2
650 Z=(X3(G,3)-V1)*B8(Q+1,1)-(X3(G,2)-.5)*B8(Q+1,2)+V1
660 X3(G,2)=(X3(G,2)-.5)*B8(Q+1,1)+(X3(G,3)-V1)*B8(Q+1,2)+.5
670 X3(G,3)=(Z-V1)*B8(Q,1)+(X3(G,1)-.5)*B8(Q,2)+V1
680 X3(G,1)=(X3(G,1)-.5)*B8(Q,1)-(Z-V1)*B8(Q,2)+.5
690 NEXTG
700 REM APPLY PERSPECTIVE AND VIEWPOINT
710 FORG=1TO2
720 X4=X3(G,1)*X3(G,2)*B8(6,2)
730 X3(G,1)=X3(G,1)+X4
740 X3(G,2)=X3(G,2)+X4
750 Y=(X3(G,2)-.5)*B8(5,1)+(X3(G,1)-.5)*B8(5,2)+.5
760 X3(G,1)=(X3(G,1)-.5)*B8(5,1)-(X3(G,2)-.5)*B8(5,2)+.5
770 X3(G,2)=(Y-.5)*B8(6,1)-(X3(G,3)-V1)*B8(6,2)+.5
780 X3(G,3)=(X3(G,3)-V1)*B8(6,1)+(Y-.5)*B8(6,2)+V1
790 NEXTG
800 REM DRAW STEREOGRAM PAIR
810 X1=(X3(1,1)-.5)*B8(7,1)-(X3(1,3)-V1)*B8(7,2)+.5:IFX1<0THENX1=0
820 X2=(X3(2,1)-.5)*B8(7,1)-(X3(2,3)-V1)*B8(7,2)+.5:IFX2<0THENX2=0
830 Y1=X3(1,2):IFY1<0THENY1=0
840 Y2=X3(2,2):IFY2<0THENY2=0
850 ON ERROR GOTO1910 : REMOVE THIS STATEMENT WITH TAPE BASIC
860 CURSOR20,0:PRINT" LEFT ":GRAPH11,01:LINEX1*320,Y1*200+10,X2*320,Y2*200+10
870 X1=(X3(1,1)-.5)*B8(7,1)+(X3(1,3)-V1)*B8(7,2)+.5:IFX1<0THENX1=0
880 X2=(X3(2,1)-.5)*B8(7,1)+(X3(2,3)-V1)*B8(7,2)+.5:IFX2<0THENX2=0
890 Y1=X3(1,2):IFY1<0THENY1=0
900 Y2=X3(2,2):IFY2<0THENY2=0
910 CURSOR20,0:PRINT" RIGHT ":GRAPH12,02:LINEX1*320,Y1*200+10,X2*320,Y2*200+10
920 NEXTI
930 NEXTJ
940 REM REVERSE ROTATE DATA SET
950 IGB>B9(10) THEN1440
960 FORG=1TO3
970 Z=(A1(G,3)-V1)*B8(Q+1,1)-(A1(G,2)-.5)*B8(Q+1,2)+V1
980 A1(G,2)=(A1(G,2)-.5)*B8(Q+1,1)+(A1(G,3)-V1)*B8(Q+1,2)+.5
990 A1(G,3)=(Z-V1)*B8(Q,1)+(A1(G,1)-.5)*B8(Q,2)+V1
1000 A1(G,1)=(A1(G,1)-.5)*B8(Q,1)-(Z-V1)*B8(Q,2)+.5
1010 NEXTG
1020 U=0
1030 GOTO740
1040 GOTO1750
1050 REM SORTING ROUTINE
1060 AA(1)=A1(A2(H,1),3):AA(2)=A1(A2(H,2),3):AA(3)=A1(A2(H,3),3)
1070 FORJ=1TO2
1080 I=AA(J):AA(J)=AA(J+1):AA(J+1)=I
1090 FORK=J+1TO3STEP-1
1100 I=AA(K):AA(K)=AA(K+1):AA(K+1)=I
1110 NEXTK
1120 NEXTJ
1130 MAX=AA(3)
1140 MIN=AA(1)
1150 RETURN
1160 REM INITIALIZATION ROUTINE
1170 CURSOR0,0:PRINT" STEREO GRAPHICS "
1180 CURSOR10,5:PRINT"1. Rotate whole data set, use horizontal slices"
1190 CURSOR10,7:PRINT"2. Slice with an inclined plane plus hill shading"
1200 CURSOR10,9:PRINT"3. Slice with an inclined plane plus eggcrating"
1210 CURSOR14,13:PRINT"Your choice 1,2,3 ?"
1220 GETC:IF (C<1)+(C>3) THEN1640
1230 IFC=1 THENRESTORE1700
1240 IFC=2 THENRESTORE1710
1250 IFC=3 THENRESTORE1720

```

```

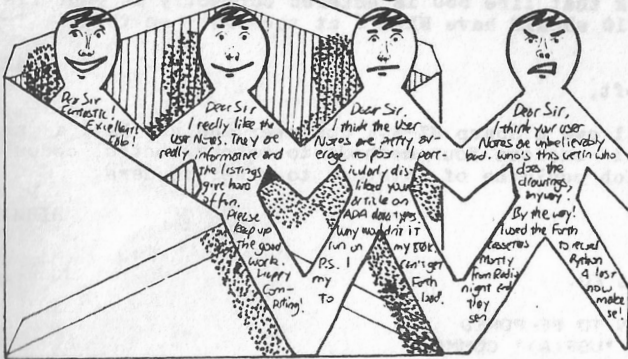
1630 FORT=1T010:READE9(T):NEXTT
1670 RESTORE:PRINTCHR$(6):CURSOR0,0:PRINT" STEREO GRAPHICS ":RETURN
1700 DATA 0.,0.,0.,0.,0.,0.,6.,1.,025,5.,0.
1710 DATA 15.,-15.,0.,0.,15.,-15.,6.,.02,6.,0.
1720 DATA 90.,0.,0.,90.,15.,-15.,6.,.03,4.,2.
1730 REM HARDCOPY ROUTINE
1740 CURSOR30,0:PRINT" HARDCOPY OF LEFT & RIGHT PICTURES (Y/N)? "
1750 GETA$:IFA$=""THEN1750
1760 IFA$="N"THEN1840
1770 IFA$<"Y"THEN1750
1780 PRINT/P"STEREO GRAPHICS: LEFT PICTURE"
1790 COPY/P2
1800 PRINT/P:PRINT/P:PRINT/P:PRINT/P
1810 PRINT/P"STEREO GRAPHICS: RIGHT PICTURE"
1820 COPY/P3
1830 PRINT/P:PRINT/P:PRINT/P:PRINT/P
1840 CURSOR30,0:PRINT" ANOTHER RUN (Y/N)?":SPACE$(49):
1850 GETA$:IFA$=""THEN1850
1860 IFA$="N"THEN1890
1870 IFA$<"Y"THEN1850
1880 RUN
1890 PRINTCHR$(6):CLR:GRAPH00,12,C,11,C,01
1900 END
1910 RESUMENEXT

```



LETTERS page

READERS LETTERS



Dear User Notes,

I've done it, solved the MZ-80PD riddle, I have found out how to copy the SP-6015 Master Disk.

First you format a blank disk, using the program on the Master Disk. Oh! I nearly forgot you need Xtal CP/M v2.21 (that's what I used).

Next you boot CP/M then load the copy program. Place the Master in Drive 1 and the Sharp BASIC format blank disk in drive 2.

When the copy program asks you for a SOURCE DRIVE type A and for a DESTINATION DRIVE type B.

Now it will ask you what part of the disk you want copied, just press W for the whole Disk.

Once this is complete, press the BIG RESET button, the one that says power! Now boot the new copy and in a few seconds you should hear a beep telling you that you have just renewed your Master Disk.

I hope this will cure the WHAT AM I GOIN TO DO WHEN THE MASTER WEARS OUT syndrome suffered by K disk users. If you have an A I can't help I've only got a well used and battered MZ-80K.

A. MARTIN
NEWTOWNABBEY

Dear Sirs,

I have a problem with one of the games for the MZ-80A, the Grand Prix, Author S.G. Dow, in your User Notes, Issue 13 in that I am receiving an error against line 110 - "Next was used without a corresponding For". Looking at the programme with my limited knowledge there would appear to be several lines missing. Could you please confirm and, if so, arrange for me to have a note of the missing lines.

K.F. PORTER
ESSEX

Please check that line 550 is entered correctly in your listing. Also line 6210 should have NEXT J at the end - we think! Try it.

SHARPSOFT

Dear Sharpsoft,

I still own a Sharp MZ-80K computer and now have a printer attached. I enclose four methods to output control codes to a printer, which could be of interest to other readers.

P. CLARK
BIRMINGHAM

```

500 REM METHOD 0 & EG.
501 REM
502 REM B=LOCATION TO BE POKED
504 REM A=USED IN 'USR(A)' COMMAND
506 B=18573:A=18544:PRINT/P
508 POKEB,27:USR(A):POKEB,58:USR(A):POKEB,0:USR(A):POKEB,0:USR(A)
510 POKEB,0:USR(A)
511 END
512 REM
514 REM METHOD 1
516 REM
518 REM X=ASCII TO BE POKED
519 REM
520 B=18573:A=18544:PRINT/P
530 GOTO560
540 POKEB,X:USR(A)
550 RETURN
560 X=27:GOSUB540:X=58:GOSUB540:X=0:GOSUB540:X=0:GOSUB540:X=0:GOSUB540
570 END
580 REM
590 REM METHOD 2
600 REM
610 REM "X=ASCII TO BE POKED:Y=NO OF ITEMS TO BE POKED:Z=LOOP VARIABLE"
620 REM
630 B=18573:A=18544:PRINT/P
640 READ Y:FORZ=1TOY:READX:POKEB,X:USR(A):NEXTZ
650 DATA 5,27,58,0,0,0
660 END
670 REM
680 REM METHOD 3 (USED IF > ONE PRINTER INSTRUCTION NEEDED IN A PROGRAM)
690 REM
700 REM EG. ESC @, ESC 27,58,0,0,0, ESC 17, ESC 18 ETC ALL IN THE SAME PROGRAM
710 REM
720 REM "A(=NO TO BE POKED:Y=NO OF ITEMS TO BE POKED:Z=L=M=LOOP VARIABLE"
730 REM
740 REM IF 800 Z=1 TO Y THEN LEAVE OUT LINE 820
745 REM
750 B=18573:A=18544:PRINT/P
760 READ Y
770 DIMA(Y)
780 FORZ=1TOY:READA(Z):NEXTZ
790 REM
800 FORL=1TO3:POKEB,A(L):USR(A):NEXTL
810 REM
820 FORM=4TOY:POKEB,A(M):USR(A):NEXTM
830 DATA 5,27,58,0,0,0
840 END

```

©SHARPSOFT

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