

APRIL 86

6



UK EINSTEIN USER GROUP  
NEWSLETTER

Issue Number Six April 1986

EDITORIAL

No apologies for being late ,you've heard them all before and I can't think of any new ones.

Just to prove that these editorials are read and acted upon a parcel arrived on my doorstep with three disks to be reviewed,these are AGROVATOR,Dr.FRANKIE and SCREEN PLUS all from Syntaxsoft,our thanks to them for their support.

The first two are reviewed later in this issue and SCREEN PLUS will be reviewed in next months.

At the same time another parcel appeared,this contained the long awaited SPECULATOR ,this will be reviewed in a later issue ( hopefully next month ) as time needs to be spent on it,but one problem has been found,some of the Spectrum games appear to be out of date and are hard to find in the chain stores,we suggest that you find a back street computer shop who will probably have stock which may even be on offer to clear.

We have also recieved inserts for four new disks from SCREENS,but no review copies ( hope your reading this Nigel ),details of which are reproduced on another page.

I have kept the best for last .Tatung ( UK ) have woken up at last and are currently producing another 10,000 + EINSTEIN'S , advertising starts on May 6th ,prices are not known yet but some of us are going to be annoyed (again ).Lets hope that the Einstein gets the support that it deserves this time around.

Now about this months issue.

The reviews are as above, plus our Adventure column,various programs ,articles and the start of our Machine Code Tutorial.

Also you should find with this newsletter a label with my address on it ,this is for all those ideas ,articles,programs,reviews , complaints etc that you have laying around just looking for a home.

MUST SELL

(due to unforeseen circumstances)

Einstein (single disc),  
with colour monitor,  
80 coloumn card and 40/80 switch,  
WS-2000 modem,  
TP100 printer,  
Zip-Stick joystick,  
Original software bundle

++++

Einsoft programmers kit,  
Easidata,  
Konami 4 game pack,  
3 modem books.

£500.00 the lot.

All in A1 condition.

Please contact A.G.Surridge on 0245-81878 (Evenings) (ESSEX)

SCIENCE and SORCERY - THE ADVENTURER'S COLUMN  
 ( DAMMIT I'VE DIED AGAIN!!)

ZORK I  
 =====

Zork I is the first in Infocom's Zork trilogy, the other two being Zork II and Zork III (amazing huh?). The three games are linked to each other, Zork I leads on to Zork II, which leads to the beginning of Zork III, but each of the games can easily be played alone.

You start Zork I outside a dilapidated old house. Surrounding the house is a forest in which you can find an ornate and irritating egg, a grating which you can't open without a key and a path to the canyon. Entering the house will lead you to the Great Underground Empire of Zork, populated by such pleasant characters as a murderous troll and a tricky thief. This world is pretty much the same in Zorks II and III, but there are quite fundamental differences between each of these games (these will be explained in later reviews). This world is also the basis for a later Infocom game, Sorcerer - which is reputed to be one of the best magical adventure games ever written.

Unfortunately, as far as I am aware this game is not yet available for the Einstein, but perhaps a little hassling of Softsel will result in Sorcerer being released for the Einstein.

The object of the Zork 1 is to explore the underground world, collecting various treasures and returning them to a trophy case back in the house. You must find and utilise various magical devices created by the famous company Frobozz. There are twenty treasures to collect, and doing so will earn you maximum points. But this isn't necessarily the end of the game.

Zork I is a pretty old game now, but even so it still compares very favourably with most other games around at the present time. The problems presented to you as you as you explore the world of Zork are difficult, and often quite funny (although the humour doesn't compare with Hitchhikers Guide to the Galaxy - also from Infocom). Unlike more recent Infocom games, Zork I has very little character interaction - something I consider very important in adventure games. However, it is a very well designed game, with very impressive location descriptions. It's well worth the price of about twenty five pounds, and should provide you with endless hours of fun. I know somebody who plays one of the earlier implementations of this game on a DEC VAX mini computer, and he has not finished it in over two years of lunchtime playing (of course he wouldn't consider playing it during work hours!) Look out for a map of Zork I in future issues of the newsletters.

A one line program

```
10 FORT=0T0360STEP9:X=15*COS(RAD(T));Y=15*SIN(RAD(T));
DRAW99,96T099+X1,96+Y1,1:DRAW99,96T099+X,96+Y:X1=X:Y1=Y:NEXT:GOTO10
```

MACHINE CODE PROGRAMMING FOR THE BEGINNER.  
 (Using the BBCBASIC Assembler)

Introduction.

Due to my wealth of experience in machine code programming (6 months) and the vast amount of material I have had published (1 article in the mag) I have been asked many times to sort out machine code programs and problems by many of our subscribers. I am also frequently asked to give tuition by those subscribers in close contact. The upshot of this is that since a machine code for beginners page is desperately needed and the best person to write such a series is a beginner, here I am.

My first forays into machine code (I think I'll abbreviate this to MC in future) on the Einstein were fraught with heartache and problems because of the way the screen, keyboard, P.S.G, and disc controller are linked to the Z80 processor. Usually these items are "memory mapped". That is part of the 64K of memory is used for those functions. Anyway, once I found out how to access these there was no stopping me. The first thing I wanted to do was to put something to the screen. Then I wanted to be able to read the keyboard. If you can do these then you can do almost anything. That's what this series of articles is about. First we will put some characters to the screen and then we will read the keyboard. When put together all sorts of things can be done, QUICKLY.

Now I will try not to get too technical but some digging around into the inner sanctum of the operations of processors at their most basic level are needed. The Z80 processor has several registers that it can use for data manipulation. One of these is called the accumulator, which is shortened to 'A'. This is the one we will start off with.

In BBCBASIC there are 149 reserved words including the operating system commands. In the Z80 instruction set (same as reserved words) there are about 50 commands 15 of which are rarely used, even by experienced programmers. This leaves about 35 to get to grips with. They are called MNEMONICS, said with a silent leading M.

The first of these is LD - this simply means 'load'. The format is LD this,that. (there must be a space between LD and 'this'). All it does is to store the value in 'that' to the location or register in 'this'. 'That' can be a register or a location and 'this' can be a register or a location that can be a number (also a literal). The only restriction we are worried about at this stage is that one of them must be a register. Some examples are:-

```
LD A,E           ;load the 'A' register with the value
                  ;in the 'E' register.
LD (location),A  ;loads the location in brackets with
                  ;the value in the 'A' register'
LD A,(location) ;loads the 'A' register with the
                  ;value in the location pointed to.
LD A,num         ;loads the 'A' register with the
                  ;number
LD A,'k'         ;loads the 'A' register with lower
                  ;case k (the character between quotes)
```

You will have noticed that the location is in brackets. This indicates that the value IS a location. This will become clearer later.

The next instruction we want to deal with is very sophisticated. It is 'RST 8'. There are actually several RST instructions starting at 'RST 0' and working up through 'RST 8', RST 16, RST 24, RST 32, RST 40, RST 48, to RST 56. They are actually a short way of doing a GOSUB to each of the named locations. i.e. 'RST 8' is a short way of saying GOSUB 8. (Actually in MC GOSUB is replaced with CALL).

'RST 8' is the only one that we are interested in because the Einstein uses it to do lots of things. When an 'RST 8' is done the Einstein then looks at the value following the instruction to see which one of about 50 subroutines to actually go to. For instance if the value after the 'RST 8' is 158 then the subroutine will put a character to the screen! That's one that we want. If the value

following the 'RST 8' is 156 then the subroutine will read the keyboard! Hey there's another. So far so good.

The one I want to deal with this time is the 'RST 8' followed by 158 so that we can put some characters to the screen. In BASIC if we want to put something to the screen then we can use 'PRINT Q\$', in MC we have to load a register with the character to be printed and then use the appropriate subroutine.

Now to talk about the assembler for a moment. The assembler I will be using for these articles is the BBCBASIC assembler. (Read the book for detailed information). It will support all the functions that are required of an assembler but will also try to assemble some MNEMONICS that are not valid. Do not worry about this as I will not be using any of these.

Back to the tutorial.

Any valid ASCII character can be put to the screen as well as the graphics characters. If in BASIC we use PRINT CHR\$(65); it will print an 'A' to the screen. The MC equivalent is

```
LD A,65          ;chr no. to print is loaded into 'A' register
RST 8           ;MC subroutine
DEFB 156        ;function to perform i.e. put it to screen
RET            ;and this will return you to BASIC when the
               ;routine is called from there.
```

So now we know what we are doing let's try a program.

( The [ and ] are produced by pressing the left and right arrow keys )  
Type in the following listing.

```
10 HIMEM=32768:REM RESERVE MEMORY FOR PROGRAM
20 PROC_ass(0):REM FIRST PASS OF ASSEMBLER
30 PROC_ass(3):REM Second pass of assembler, with listing and errors
40 PRINT"PRESS SPACE BAR TO CONTINUE"
50 key=GET:IFkey<>32 THEN GOTO50 :REM Waits for you to press the
space bar before going on. You start the MC program.
60 CALL 32768:STOP:REM Executes the MC program and stops here if all
works OK.
70 DEFPROC_ass(opt)
80 PZ=32768
90 [OPT opt
100 ;MACHINE CODE LINES INSERTED HERE
110 .start LD A,68; puts the value of the character 'D' into
the register
120 RST 8 ; Machine code call
130 DEFB 158; to do this particular function, which in this case is
print the character in the accumulator or 'A' register to the screen
1260 RET; returns to BBCBASIC
1270 ]
1280 ENDFROC
```

Line 10 reserves memory for the MC program. We have used location 32768 because it is convenient. It could have been anything between 16000 and 540000 but the higher and lower you go the more likely it is to clash with what is already in memory.

Lines 20 and 30 assemble the MC program and because it needs to be done twice to calculate all the backward jumps lines 20 and 30 both do the same thing. Line 30 also gives a listing of the assembled program.

Lines 40 and 50 allow a pause for looking at the program listing before running.

Line 60 actually executes the MC program, it is because we run

the MC program like this from BASIC that we have to use a RET instruction to get back to BASIC.

Line 80 tells the assembler where to put the MC program in memory.

Lines 110 to 1260 are the source code for the MC program.

Below is a listing of the above program as the assembler would list it.

Run the program and see what happens. You should get an upper case D put to the screen directly after the PRESS SPACE BAR TO CONTINUE message so far so good. Try replacing the 68 in line 110 with 12. Go on, try it! Now did it clear the screen? Good. We are really getting somewhere. Think about it, if we put 12 to the screen then it clears then we can put as many characters to the screen as we want, like MESSAGES! But if we do it all like that it could be very long winded. We will have to set up a FOR NEXT loop to print messages. To do this we will have to use some more registers. But first some more about registers. A single register will hold a number up to 255. This is actually FF in hex but I'm not going into things like that just yet. The Einstein has a 64K memory, or has 65535 memory locations in which to store data. Hey! you say remembering the example quoted at the bottom of the first page if a register only holds 255 how can I look at a location over 255. Use two registers I reply quickly. If we use two registers together we can actually hold numbers up to 65535. Only certain registers can be used together, the 'H' and 'L' registers, the 'B' and 'C' registers, and the 'D' and 'E' registers. These are normally written as HL or BC or DE if they are being used together. Use the 'B' register to hold the number of characters to be printed and the 'HL' register to tell us where the message is in memory.

OK on with the show.

```

Use lines 10 to 100 as in the previous listing and then
110 .start LD B,35; number of characters to print
120 LD HL, message; this is where the message is in memory
130 .loop LD A,(HL); get a character
140 RST 8; MC call
150 DEFB 158; print it
160 INC HL; advance to next character
170 DEC B; reduce count
180 RET Z; back to BASIC if all done
190 JP loop; otherwise do it again
200 .message DEFB 12; to clear the screen and home cursor
210 DEFM 'This is the message'
220 DEFB 13; carriage return
230 DEFB 10; line feed
240 DEFM 'Good isnt it?'
    
```

and finish off with lines 1260 and 1280 as before

Run it, press the space bar when requested, if all is well the screen will clear your message will be printed and you will return to BASIC.

Excercise for next months lesson, Write a MC routine to put a MENU to the screen.

Best of luck,Chris.

Another one line program

With this one make a note of the number at the top leftand, if it produces a pattern you like replace  $V=0.8+RND(1)*3$  with that number.

```

10 CLS;V=0.8+RND(1)*3:PRINTV:FORT=0TO200*VSTEPV:R=T/V:X=R*COS(T):
Y=R*SIN(T):DRAW99,98TO99+X,98+Y:NEXTT:FORP=1TO999:NEXT:GOTO10
    
```

A PROGRAM TO CALCULATE INCOME TAX.  
John Luther, Shrewsbury

This program was written simply as an exercise but it works out the tax to be collected from the occupational pension covering that due on the DHSS retirement pension interest etc. The tax shown as due is that to be deducted from the principal pension. It will calculate in advance on the weekly figures or check at year end on the payments actually received during the year. Being based on pension it works out the age allowance based on the appropriate income.

Perhaps there are not many members who are concerned with age allowances but it could be useful if experienced members would criticise it constructively so that others, and there must be some, who, like me, have lately come to programming and having no available teachers would be pleased to learn from other EINSTEIN users. And, dare I say it, it could be of benefit to them to formulate means of improving the program.

\*\* TAX CALCULATION PROGRAM \*\*

```

10 RST:G=0:REM WHEN G=1 VALUES OF VARIABLES ARE REVEALED
20 TCOL8:PRINT@6,1;"++A PROGRAM TO CALCULATE++"
30 print@6,2;"++ Income Tax ++":TCOL15
40 REM
50 REM *****
60 REM * Based on Information *
70 REM * March 1986 * J M Luther *
80 REM *****
90 REM * Correct 85/6 Tax year for *
100 REM * Taxable Incomes to HR 40% *
110 REM *****
120 GOSUB 4000:REM Change data
130 RESTORE
140 for I=1 to 9:READ T (I):NEXT
150 DATA 30,40,4255,8800,33,19,3455
160 DATA 16200,19200
170 T=0:PRINT:FMT5,2:CLS
180 TCOL3:PRINT " *** Calculate Pensions *** ":TCOL15
190 PRINT:GOSUB 3000:REM Pensions Options
200 CLS:PRINT "Do you wish to add Miscellaneous Earnings? (Y/N)":
:Y$=Inch$:PRINT Y$:PRINT
210 IF Y$="n" OR Y$="N" THEN 230
220 GOSUB 5000:REM Misc Earnings
230 T=C+X+AS:FOR I=1TO200:NEXT:CLS
240 PRINT "TOTAL PENSIONS & EARNINGS £";T
500 REM ++ CALCULATION OF AGE ALLOWANCE++
510 PRINT
520 TCOL3: PRINT " ***Calculate Age Allowance*** ":PRINT:TCOL15
530 CD=0:D=0:RA=0:B=0
540 S=0:A=0:V=0
550 MA=0:MI=0
560 PRINT "BANK & BS INTEREST"
570 INPUT "ENTER INTEREST (UNTAXED AT SOURCE) ";B
580 INPUT "ENTER INTEREST RECEIVED NETT OF TAX ";S
590 A = S*100/(100-T(1)):IFG=1THEN PRINT "A= ";A
600 INPUT COVENANT GROSS ";V
610 INPUT "MORTGAGE INTEREST PAID ";M
620 TI = T+B+ A-(V+M)
630 PRINT "TOTAL INCOME FOR AGE ALLCE IS £;TI
    
```

```

640 IF TI<=19200 THEN 670
650 CLS:TCOL6
660 PRINT "TOTAL INCOME IS GREATER THAN '";T(9)"SO I CAN'T COPE WITH
    IT";GOTO 2120
670 PRINT "CURRENT MAX AGE ALLCE '";T(3)
680 PRINT "MAX INCOME FOR AGE ALLCE'";T(4)
690 IF TI >T(4) GOTO 710ELSE PRINT "FULL AGE ALLOWENCE IS GIVEN ";T(3)
700 RA=T(3):GOTO 1000
710 RA =T(3)-((TI-T(4))*2/3):RA=INT(RA+0.5)
720 IF RA>=T(7)THEN 730:ELSE RA=T(7):PRINT "MARRIED ALLOWANCE IS '";RA
730 PRINT "REDUCED AGE ALLOWANCE DUE IS '";RA
1000 REM CALCULATE TAX DUE:PRINT
1010 TCOL3: PRINT "   ***Calculate Allowance against Pay***":TCOL15
1020 IFG=1 THEN PRINT "RA= ";RA;"E= ";X;"X= ";X
1030 IF RA<=(B+X+AS) THEN XD=(B+X+AS)-RA:PRINT"RP+BANK INT+EARNINGS
    exceed all ces,so":GOTO1050:CD=0
1040 CD = RA-E-AS:CD=INT(CD+.5)
1050 PRINT "ALLOWANCES AGAINST PAY = £;CD
2000 REM ** CALCULATE TAX DUE ON PRINCIPAL PENSION **
2010 D=(XD+C-CD)*T(1)/100:IF D<1 THEN 2080
2020 IF G=1 THEN PRINT "RA=";RA;"XD=";XD;"CD=";CD
2030 TCOL3:PRINT "   ***Calculate Tax Due on Pension***"
2040 IF C+XD-CD>T(8) THEN PRINT "HIGHER RATE OF TAX DUE":H=(C+XD-
    CD)-T(8)*T(2)/100+T(8)*T(1)/100:H=D:PRINT
2050 FOR I =1 TO 15
2060 TCOL5:PRINT@5,22;"TAX DUE FOR YEAR IS 3";D:TCOL8
2070 PRINT@5,22;"TAX DUE FOR YEAR IS '";D ;NEXT I:GOTO 2110
2080 FOR J = 1 TO 15
2090 TCOL10:PRINT@8,22;"NO TAX IS DUE"
2100 TCOL8:PRINT@8,22;"NO TAX IS DUE":NEXT
2110 IFG=1 THEN PRINT "C= ";C;"D= ";D;"T=";T
2120 TCOL15:PRINT:END
3000 PRINT@2,7;"Options for Pensions/Salary calculations"
3010 PRINT@4,9;"1 Exit Program"
3020 PRINT@4,11;"2 Enter weekly pensions"
3030 PRINT@4,13;"3 Enter year end totals"
3050 PRINT@4,17;"Which?";N$=INCH$:PRINT N$:PRINT
3060 N=VAL(N$):IF N<0 OR N>4 THEN BEEP5:CLS: GOTO 3000
3070 ON N GOSUB 3080,3120,3260
3075 GOTO 200
3080 CLS:PRINT@8,2;"GOODBYE":BEEP5:END
3090 RETURN
3100 RST
3110 REM ** WEEKLY PAID PENSIONS **
3120 CLS:PRINT "   Weekly rates of pension   ":PRINT
3130 PRINT "C S PENSION"
3140 INPUT "Weekly Rate   '";CSP
3150 C=CSP*T(5)+((CSP+(CSP*7/100))*T(6))
3160 PRINT "ANNUAL C.S. PENSION =   '";C:
3170 PRINT "NIR Pension"
3180 INPUT "Weekly Rate(W)   '";W
3190 INPUT "Weekly Rate(J)   '";J
3200 PRINT "TOTAL WEEKLY RP IS           £";W+J
3210 RP=W+J
3220 X=RP*T(5)+((RP*7/100))*T(6))
3230 PRINT "TOTAL RETIREMENT PENSION   £";X
2340 RETURN
3250 REM ** ANNUAL TOTAL OF PENSIONS **
3260 CLS: PRINT "Actual Pensions for year"
3270 PRINT "C S PENSION"
3280 INPUT "TOTAL CSP   '";C
3290 PRINT "NIR Pension"

```

```

3300 INPUT "TOTAL RP      ";X
3310 RETURN
4000 PRINT@2,3;"Do you wish to amend the basic data?";
      :Y$=INCH$;PRINT Y$;PRINT
4010 IF Y$="Y"OR Y$="y" THEN GOTO 4030
4020 CLS: GOTO 130
4030 FOR I=1 TO 9:READ X(I):NEXT
4040 PRINT@4,5;"BASIC RATE %=";X(1)
4050 PRINT@4,6;"HIGHER RATE%=";X(2)
4060 PRINT@4,7;"FULL AGE ALLOWANCE=";X(3)
4070 PRINT@4,8;MAX INCOME FOR AGE ALLCE=";X(4)
4080 PRINT@4,9;"WEEKS TO CHANGE OF RATE=";X(5)
4090 PRINT@4,10;"WEEKS AFTER CHANGE=2;X(6)
4100 PRINT@4,11;"MARRIED BASIC ALLCE=";X(7)
4110 PRINT@4,12;"40% BAND=";X(8)
4120 PRINT@4,13; "45% BAND="X(9)
4130 PRINT: FOR I=1 TO 15:TCOL6
4140 PRINT@5,15;"
4150 PRINT@5,15; "When line is shown press ESC":TCOL15
4160 NEXT I:PRINT
4170 LIST 150,2:TCOL9;PRINT@9,20;"Make changes NOW":TCOL15
4180 PRINT @7,21+"Then RE-RUN the program"
4190 GOTO 2120
4200 CLS:RETURN
5000 CLS:PRINT "* Miscellaneous Earnings *"
5020 INPUT "MISC EARNINGS ";AS
5030 RETURN

```

TRIG HELPER  
by  
Dave Harvey

Trig helper is a program I put together some time ago to help me with some engineering drawings I was doing .All you do is enter the info as requested, but it must be entered in proportion with the displayed triangle .Continue to enter what info you have and when the program has enough it will stop taking inputs and calculate the complete triangle .The program will tell you if you have entered impossible or bad info and invite you to start again .The way the calculations are done may be worth remembering,each formula is preceded by a test to see if the data required is non zero if not the formula is passed and the next one tested until a formula is used and one more part to the puzzle is found enabling previously unusable formuli to be used,and,so it goes on until everything is found .If on the first pass all the answers aren't found a second pass is done .I have used this program on many occasions and have found no bugs I hope someone finds it usful.

```

10 REM*****
11 REM**                                     **
12 REM**          TRIG HELPER                **
13 REM**    by David Harvey                  **
14 REM**          April 1986                 **
15 REM**                                     **
16 REM*****
20 REM*****
25 ON ERROR GOTO 6000:REM REM THE WHOLE OF THIS LINE BEFORE TRYING
   TO DEBUG THE PROGRAM.WHEN FULLY RUNNING TAKE OUT REM
35 COLOUR 6
36 COLOUR 128
37 CLS
40 PRINTTAB(15,10),"TRIG HELPER"

```

```

50 PRINTTAB(2,12),"RIGHT ANGLE TRIANGLE (Y/N)";
60 G$=GET$
70 IF G$="Y"THEN 4040
80 IF G$="N"THEN 4170
90 GOTO60
100 REM*****
110 DEF PROCSETSQR
120 CLS
130 GCOL15,6
140 MOVE100,300
150 DRAW600,700
160 DRAW600,300
170 DRAW 100,300
180 MOVE 600,320
190 DRAW 580,320
200 DRAW 580,300
210 COLOUR 4
220 PRINTTAB(13,6),"HYP"
230 PRINTTAB(13,15),"ADJ"
240 PRINTTAB(26,9),"OPP"
250 PRINTTAB(8,13),"A1"
260 PRINTTAB(21,5),"A2"
270 PRINTTAB(3,20),"ENTER THE INFORMATION IF YOU HAVE IT
    ELSE ENTER 0"
280 ENDPROC
290 REM*****
300 DEF PROCSETOBT
310 CLS
320 GCOL15,6
330 MOVE 100,300
340 DRAW 600,700
350 DRAW 800,300
360 DRAW 100,300
370 MOVE 900,700
380 GCOL 15,3
390 DRAW 900,300
400 PLOT21,800,300
410 MOVE 900,700
420 PLOT 21,600,700
430 MOVE 600,700
440 PLOT 21,600,200
450 MOVE 800,300
460 PLOT 21,800,200
470 MOVE 100,300
480 PLOT21,100,200
490 DRAW 800,200
500 DRAW 780,180
510 MOVE 800,200
520 DRAW 780,220
530 MOVE 100,200
540 DRAW 120,220
550 MOVE 100,200
560 DRAW 120,180
570 MOVE 580,180
580 DRAW 620,220
590 MOVE 620,180
600 DRAW 580,220
610 MOVE900,700
620 DRAW 880,680
630 MOVE 900,700
640 DRAW 920,680
650 MOVE 900,300

```

```

660 DRAW 880,320
670 MOVE 900,300
680 DRAW 920,320
690 MOVE 450,750
700 DRAW 580,640
710 PLOT37,5,5
720 MOVE 615,640
730 PLOT 37,5,5
740 DRAW 700,750
750 COLOUR 4
760 PRINTTAB(13,6),"S1"
770 PRINTTAB(18,15),"S2"
780 PRINTTAB(31,9),"S3"
790 PRINTTAB(8,13),"A1"
800 PRINTTAB(24,5),"A2"
810 PRINTTAB(29,13),"A3"
820 PRINTTAB(37,7),"H"
830 PRINTTAB(15,17),"F1"
840 PRINTTAB(29,17),"F2"
850 PRINTTAB(17,0),"FA1"
860 PRINTTAB(28,0),"FA2"
870 PRINTTAB(3,20),"ENTER THE INFORMATION IF YOU HAVE IT
    ELSE ENTER 0"
880 ENDPROC
890 REM*****
900 DEF PROCNUMERIC(PROMPT$)
910 PRINTTAB(2,22),"";
920 PRINT"
930 PRINTTAB(2,22),"";
940 PRINT PROMPT$;
950 INPUT;INVAL$
960 IF LEN(INVAL$)> 10 THEN INVAL$=LEFT$(INVAL$,10)
970 FLAG1=0
980 FOR LOOP = 1 TO LEN(INVAL$)
990 IF MID$(INVAL$,LOOP,1)<>"." AND MID$(INVAL$,LOOP,1)<"0"
    OR MID$(INVAL$,LOOP,1)>"9" THEN FLAG1=1
1000 NEXT LOOP
1010 IF FLAG1=1 THEN 910
1020 INVAL=EVAL(INVAL$)
1040 ENDPROC
1050 REM*****
1060 DEF PROCANGULAR(PROMPT$)
1070 PRINTTAB(2,22)"";
1080 PRINT"
1090 PRINTTAB(2,22)"";
1100 PRINTPROMPT$;" DEGREES";
1110 INPUT;INDEG$;
1120 FLAG1=0
1130 FOR LOOP=1 TO LEN(INDEG$)
1140 IF MID$(INDEG$,LOOP,1)<"0" OR MID$(INDEG$,LOOP,1)>"9" THEN FLAG=1
1150 NEXT LOOP
1160 IF FLAG1=1 THEN 1070
1170 IF VAL(INDEG$)<0 THEN 1070
1180 INDEG=EVAL(INDEG$)
1200 PRINTTAB(25,22);" MINUTES";
1210 INPUT;INMIN$;
1220 FLAG1=0
1230 FOR LOOP=1 TO LEN(INMIN$)
1240 IF MID$(INMIN$,LOOP,1)<"0" OR MID$(INMIN$,LOOP,1)>"9"
    THEN FLAG1=1
1250 NEXT LOOP
1260 IF FLAG1=1 THEN 1070

```

```

1270 IF VAL(INMIN$)<0 THEN 1070
1280 INMIN=EVAL(INMIN$)
1290 IF INMIN<0 OR INMIN>59 THEN 1070
1300 INMIN=INMIN/60
1310 INDEG=INDEG+INMIN
1320 INANG=RAD(INDEG)
1330 ENDPROC
1350 REM*****
1360 DEF PROCGETSQR
1370 OPP=0:ADJ=0:HYP=0:A1=0:A2=0
1380 PROCNUMERIC("SIDE OPP")
1390 OPP=INVAL:IF OPP>0 THEN CO=CO+1
1400 PROCNUMERIC("SIDE ADJ")
1410 ADJ=INVAL:IF ADJ>0 THEN CO=CO+1:IF CO=2 THEN 1440
1420 PROCNUMERIC("SIDE HYP")
1430 HYP=INVAL:IF HYP>0 THEN CO=CO+1
1440 IF CO=0 THEN PRINTTAB(1,1)"AT LEAST ONE SIDE IS REQUIRED":
    GOTO 1600
1450 IF HYP>0 AND ADJ>0 AND HYP<ADJ THEN PRINTTAB(1,1)"BAD
    PROPORTION":GOTO 1600
1460 IF HYP>0 AND OPP>0 AND HYP<OPP THEN PRINTTAB(1,1)"BAD
    PROPORTION":GOTO 1600
1470 IF ADJ>0 AND OPP>0 AND ADJ<OPP THEN PRINTTAB(1,1)"BAD
    PROPORTION":GOTO 1600
1480 IF HYP>0 AND ADJ>0 AND HYP=ADJ THEN PRINTTAB(1,1)"HYP AND ADJ
    CANNOT BE EQUAL":GOTO1600
1490 IF HYP>0 AND OPP>0 AND OPP=HYP THENPRINTTAB(1,1)"HYP AND OPP
    CANNOT BE EQUAL":GOTO1600
1500 IF CO=2 THEN 1580
1510 PROCANGULAR("ACUTE ANGLE A1")
1520 A1=INANG:IF A1>0 THEN CO=CO+1
1530 IF DEG(A1)>45 OR DEG(A1)<0 THEN PRINTTAB(1,1)"BAD PROPORTION":
    GOTO 1600
1540 IF CO=2 THEN 1580
1550 PROCANGULAR("OBTUSE ANGLE A2")
1560 A2=INANG:IF A2>0 THEN CO=CO+1
1570 IF A2<>0 AND DEG(A2)>90 OR DEG(A2)<45 THEN PRINTTAB(1,1)"BAD
    PROPORTION":GOTO1600
1580 EROR=0
1590 ENDPROC
1600 EROR=1:PROCSOUND: FOR DELAY=1TO2500:NEXT:ENDPROC
1610 REM***** GET OBT *****
1620 DEF PROCGETOBT
1630 S1=0:S2=0:S3=0:A1=0:A2=0:A3=0:PA1=0:PA2=0:P1=0:P2=0:H=0:AREA=0
1640 PROCNUMERIC("BASE S2")
1650 S2=INVAL:IF S2>0 THEN CO=CO+1
1660 PROCNUMERIC("LONG SIDE S1")
1670 S1=INVAL:IF S1>0 THEN CO=CO+1
1680 PROCNUMERIC("SHORT SIDE S3")
1690 S3=INVAL:IF S3>0 THEN CO=CO+1
1700 IF S1>0 AND S3>0 AND S3>S1 THEN PRINTTAB(9,9)"BAD PROPORTION":
    GOTO 2160
1710 IF S1>0 AND S2>0 AND S1>S2 THEN PRINTTAB(9,9)"BAD PROPORTION":
    GOTO 2160
1720 IF S2>0 AND S3>0 AND S3>S2 THEN PRINTTAB(9,9)"BAD PROPORTION":
    GOTO 2160
1725 IF S1>0 AND S2>0 AND S3>0 AND S1+S3=<S2 THEN PRINTTAB(9,9)"BAD
    PROPORTION":GOTO2160
1730 IF CO=3 THEN 2170
1740 PROCNUMERIC("BASE PORTION P2")
1750 P2=INVAL:IF P2>0 THEN CO=CO+1
1760 IF P2>0 AND S2>0 AND P2=>S2 THEN PRINTTAB(9,9)"BAD PROPORTION":

```

```

GOTO 2160
1765 PROCBALANCE
1770 IF CO=3 THEN 2170
1780 PROCNUMERIC("BASE PORTION P1")
1790 P1=INVAL:IF P1>0 THENCO=CO+1
1800 IF P2>0 AND P1>0 AND P2>P1 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO 2160
1810 IF S2>0 AND P1>0 AND P1=>S2 THENPRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
1820 IF S2>0 AND P1>0 AND P2>0 AND P1+P2<>S2 THEN PRINTTAB(9,9)"BAD
ADDITION":GOTO2160
1830 IF CO=3 AND S2>0 AND P1>0 AND P2>0 THEN CO=CO-1
1835 PROCBALANCE
1840 IF CO=3 THEN 2170
1850 PROCNUMERIC("HEIGHT H")
1860 H=INVAL:IF H>0 THEN CO=CO+1
1870 IF H>0 AND S1>0 AND H=>S1 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
1880 IF H>0 AND S3>0 AND H=>S3 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
1890 IF H>0 AND S2>0 AND H>S2 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO 2160
1895 PROCBALANCE
1900 IF CO=3 THEN 2170
1910 PROCANGULAR("ANGLE A1")
1920 A1=INANG:IF A1>0 THENCO=CO+1
1930 IF A1>0 AND DEG(A1)>60 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
1934 IF CO=3 AND A1>0 AND S1=0 AND H=0 AND P1=0 AND P2=0 THEN CO=CO-1
1935 PROCBALANCE
1940 IF CO=3 THEN 2170
1950 PROCANGULAR("ANGLE A3")
1960 A3=INANG:IF A3>0 THEN CO=CO+1
1970 IF A3>0 AND DEG(A3)>90 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO 2160
1980 IF A1>0 AND A3>0 AND A1>A3 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO 2160
1984 IF CO=3 AND A3>0 AND A1=0 AND S3=0 AND H=0 AND P1=0 AND P2=0
THEN CO=CO-1
1985 PROCBALANCE
1990 IF CO=3 THEN2170
2000 PROCANGULAR("TOTAL ANGLE A2")
2010 A2=INANG:IF A2>0 THENCO=CO+1
2020 IF A2>0 AND DEG(A2)>180 THEN PRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
2030 IF A3>0 AND A2>0 AND A3>A2 THENPRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
2040 IF A2>0 AND A1>0 AND A1>A2 THENPRINTTAB(9,9)"BAD PROPORTION":
GOTO2160
2045 PROCBALANCE
2050 IF CO=3 THEN 2170
2060 PROCANGULAR("PART ANGLE PA1")
2070 PA1=INANG:IF PA1>0 THEN CO=CO+1
2080 IF PA1>0 AND A2>0 AND PA1>A2 THEN PRINTTAB(9,9)"PA1 CANNOT BE
>A2":GOTO 2160
2084 IF CO=3 AND PA1>0 AND P1=0 AND P2=0 AND S1=0 AND H=0 AND A3=0
AND A2=0 THEN CO=CO-1
2085 PROCBALANCE
2090 IF CO=3 THEN2170
2100 PROCANGULAR("PART ANGLE PA2")
2110 PA2=INANG:IF PA2>0 THEN CO=CO+1
2120 IF PA2>0 AND A2>0 AND PA2>A2 THENPRINTTAB(9,9)"PA2 CANNOT BE

```

```

>A2":GOTO 2160
2130 IF PA2>0 AND PA1>0 AND PA1<PA2 THENPRINTTAB(9,9)"BAD PROPORTION"
      :GOTO2160
2140 IF PA1>0 AND PA2>0 AND A2>0 AND PA1+PA2<>A2 THEN PRINTTAB(9,9)
      "BAD ADDITION":GOTO2160
2150 GOTO 2170
2160 EROR=1:PROCSOUND: FOR J=1TO2500:NEXT:ENDPROC
2170 EROR=0:ENDPROC
2180 REM*****
2190 DEF PROCCALSQR
2200 REM##### A1 #####
2210 IF A1>0 THEN 2310
2220 IF OPP=0 OR HYP=0 THEN 2240
2230 A1=ASN(OPP/HYP):GOTO2310
2240 IF ADJ=0 OR HYP=0 THEN 2260
2250 A1=ACS(ADJ/HYP):GOTO 2310
2260 IF OPP=0 OR ADJ=0 THEN 2280
2270 A1=ATN(OPP/ADJ):GOTO 2310
2280 IF A2=0 THEN 2310
2290 A1 = RAD(90)-A2
2300 REM##### A2 #####
2310 IF A2>0 THEN 2410
2320 IF HYP=0 OR ADJ=0 THEN 2340
2330 A2=ASN(ADJ/HYP):GOTO2410
2340 IF OPP=0 OR HYP=0 THEN 2360
2350 A2=ACS(OPP/HYP):GOTO 2410
2360 IF ADJ=0 OR OPP=0 THEN 2380
2370 A2=ATN(ADJ/OPP):GOTO2410
2380 IF A1=0 THEN 2410
2390 A2=RAD(90)-A1:GOTO2410
2400 REM##### HYP #####
2410 IF HYP>0 THEN 2520
2420 IF OPP = 0 OR ADJ=0 THEN 2440
2430 HYP = SQR(OPP^2+ADJ^2):GOTO2520
2440 IF A1=0 OR OPP=0 THEN 2460
2450 HYP=OPP/SIN(A1):GOTO2520
2460 IF A1=0 OR ADJ =0 THEN 2480
2470 HYP=ADJ/COS(A1):GOTO2520
2480 IF A2=0 OR OPP=0 THEN 2500
2490 HYP=ADJ/SIN(A2):GOTO2520
2500 IF A2=0 OR A2=0 THEN 2520
2510 HYP=OPP/COS(A2):GOTO2520
2520 REM##### OPP #####
2530 IF OPP>0 THEN 2640
2540 IF HYP=0 OR ADJ=0 THEN 2560
2550 OPP=SQR(HYP^2-ADJ^2):GOTO 2640
2560 IF A1=0 OR HYP=0 THEN 2580
2570 OPP=HYP*SIN(A1):GOTO 2640
2580 IF A1=0 OR ADJ=0THEN 2600
2590 OPP=ADJ*TAN(A1):GOTO 2640
2600 IF A2=0 OR ADJ=0 THEN 2620
2610 OPP=ADJ/TAN(A2):GOTO 2640
2620 IF A2=0 OR HYP=0 THEN 2640
2630 OPP=HYP*COS(A2):GOTO 2640
2640 REM##### ADJ #####
2650 IF ADJ>0 THEN 2760
2660 IF HYP=0 OR OPP=0 THEN 2680
2670 ADJ=SQR(HYP^2-OPP^2):GOTO 2760
2680 IF A1=0 OR HYP=0 THEN 2700
2690 ADJ=HYP*COS(A1):GOTO 2760
2700 IF A2=0 OR HYP=0 THEN 2720
2710 ADJ=HYP*SIN(A2):GOTO 2760

```

```

2720 IF OPP=0 OR A2=0 THEN 2740
2730 ADJ=OPP*TAN(A2):GOTO 2760
2740 IF OPP=0 OR A1=0 THEN 2760
2750 ADJ=OPP*(1/TAN(A1)):GOTO2760
2760 REMEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
2770 IF A1=0 OR A2 =0 OR ADJ=0 OR OPP=0 OR HYP=0 THEN 2200
2780 AREA=(OPP*ADJ)/2
2790 ENDPROC
2800 REMEEEEEEEEEE CALC OBLIQUE EEEEEEEEE
2810 DEF PROCCALOBT
2820 REPEAT
2830 IFA1>0ORA3=0 OR A2=0 THEN 2850
2840 A1=RAD(180)-(A2+A3)
2850 IFA2>0 OR A1=0 OR A3 =0 THEN 2870
2860 A2=RAD(180)-(A1+A3)
2870 IF A3>0 OR A2=0 OR A1=0 THEN 2890
2880 A3=RAD(180)-(A1+A2)
2890 IF S3>0 OR S2=0 OR A1=0 OR A2 =0 THEN 2910
2900 S3=(S2*SIN(A1))/SIN(A2)
2910 IF A1>0 OR S1=0 OR S3=0 OR A2=0 THEN 2930
2920 A1=ATN((S3*SIN(A2))/(S1-(S3*COS(A2))))
2930 IF S1>0 OR S2=0 OR A2=0 OR A3=0 THEN 2950
2940 S1=(S2*SIN(A3))/SIN(A2)
2950 IF A2>0 OR S1=0 OR A3=0 OR S3=0 THEN 2970
2960 A2=ATN((S1*SIN(A3))/(S3-(S1*COS(A3))))
2970 IF A1>0 OR S3=0 OR A2=0 OR S2=0 THEN 2990
2980 A1=ASN((S3*SIN(A2))/S2)
2990 IF A3>0 OR S2=0 OR A2=0 OR S1=0 THEN 3010
3000 A3=ASN((S1*SIN(A2))/S2)
3010 IF S2>0 OR P1=0 OR P2=0 THEN 3030
3020 S2=P1+P2
3030 IF PA1>0 OR A1=0 THEN 3050
3040 PA1=RAD(90)-A1
3050 IF PA2>0 OR A3=0 THEN 3070
3060 PA2=RAD(90)-A3
3070 IF A2>0 OR PA1=0 OR PA2=0 THEN 3090
3080 A2=PA1+PA2
3090 IF H>0 OR A3=0 OR S3=0 THEN 3110
3100 H=S3*SIN(A3)
3110 IF H>0 OR A1=0 OR S1=0 THEN 3130
3120 H=S1*SIN(A1)
3130 IF H>0 OR S1=0 OR P1=0 THEN 3150
3140 H=SQR(S1^2-P1^2)
3150 IF H>0 OR S3=0 OR P2=0 THEN 3170
3160 H=SQR(S3^2-P2^2)
3170 IF A1>0 OR PA1=0 THEN 3190
3180 A1=RAD(90)-PA1
3190 IF A3>0 OR PA2=0 THEN 3210
3200 A3=RAD(90)-PA2
3210 IF P1>0 OR PA1=0 OR S1=0 THEN 3230
3220 P1=S1*SIN(PA1)
3230 IF P2>0 OR PA2=0 OR S3=0 THEN 3250
3240 P2=S3*SIN(PA2)
3250 IF PA1>0 OR A2=0 OR PA2=0 THEN 3270
3260 PA1=A2-PA2
3270 IF PA2>0 OR A2=0 OR PA1=0 THEN 3290
3280 PA2=A2-PA1
3290 IF A1>0 OR S1=0 OR H=0 THEN 3310
3300 A1=ASN(H/S1)
3310 IF A3>0 OR H=0 OR S3=0 THEN 3330
3320 A3=ASN(H/S3)
3330 IF S3>0 OR H=0 OR P2=0 THEN 3350

```

```

3340 S3=SQR(H^2+P2^2)
3350 IF S1>0 OR H=0 OR P1=0 THEN 3370
3360 S1=SQR(H^2+P1^2)
3370 IF S1>0 OR A1=0 OR H=0 THEN 3390
3380 S1=H/SIN(A1)
3390 IF S1>0 OR A1=0 OR P1=0 THEN 3410
3400 S1=P1/COS(A1)
3410 IF S3>0 OR A3=0 OR H=0 THEN 3430
3420 S3=H/SIN(A3)
3430 IF S3>0 OR A3=0 OR P2=0 THEN 3450
3440 S3=P2/COS(A3)
3460 IF A1>0 OR P1=0 OR H=0 THEN 3480
3470 A1=ATN(H/P1)
3480 IF A3>0 OR P2=0 OR H=0 THEN 3500
3490 A3=ATN(H/P2)
3500 IF A2>0 OR S3=0 OR S1=0 OR S2=0 THEN 3520
3510 A2=ACS(((S1^2+S3^2)-S2^2)/(2*(S1*S3)))
3520 IF P1>0 OR P2=0 OR S2=0 THEN 3540
3530 P1=S2-P2
3540 IF P2>0 OR P1=0 OR S2=0 THEN 3560
3550 P2=S2-P1
3560 IF P1>0 OR H=0 OR S1=0 THEN 3580
3570 P1=SQR(S1^2-H^2)
3580 IF P2>0 OR S3=0 OR H=0 THEN 3600
3590 P2=SQR(S3^2-H^2)
3600 IF H>0 OR S1=0 OR PA1=0 THEN 3620
3610 H=S1/COS(PA1)
3620 IF H>0 OR S3=0 OR PA2=0 THEN 3640
3630 H=S3/COS(PA2)
3640 IF H>0 OR PA1=0 OR P1=0 THEN 3660
3650 H=P1/TAN(PA1)
3660 IF H>0 OR P2=0 OR PA2=0 THEN 3680
3670 H=P2/TAN(PA2)
3680 IF H>0 OR A1=0 OR P1=0 THEN 3700
3690 H=P1*TAN(A1)
3700 IF H>0 OR A3=0 OR P2=0 THEN 3720
3710 H=P2*TAN(A3)
3720 IF P1>0 OR PA1=0 OR H=0 THEN 3740
3730 P1=H*TAN(PA1)
3740 IF P2>0 OR PA2=0 OR H=0 THEN 3760
3750 P2=H*TAN(PA2)
3760 IF P1>0 OR A1=0 OR S1=0 THEN 3780
3770 P1=S1*COS(A1)
3780 IF P2>0 OR A3=0 OR S3=0 THEN 3800
3790 P2=S3*COS(A3)
3800 IF S1>0 OR P1=0 OR PA1=0 THEN 3820
3810 S1=P1/SIN(PA1)
3820 IF S3>0 OR P2=0 OR PA2=0 THEN 3840
3830 S3=P2/SIN(PA2)
3840 IF S1>0 OR H=0 OR PA1=0 THEN 3860
3850 S1=H/COS(PA1)
3860 IF S3>0 OR H=0 OR PA2=0 THEN 3880
3870 S3=H/COS(PA2)
3880 IF PA1>0 OR S1=0 OR P1=0 THEN 3900
3890 PA1=ASN(P1/S1)
3900 IF PA2>0 OR S3=0 OR P2=0 THEN 3920
3910 PA2=ASN(P2/S3)
3920 IF PA1>0 OR H=0 OR P1=0 THEN 3940
3930 PA1=ATN(P1/H)
3940 IF PA2>0 OR H=0 OR P2=0 THEN 3960
3950 PA2=ATN(P2/H)
3960 IF AREA>0 OR S2=0 OR S1=0 OR A1=0 THEN 3962

```

```

3961 AREA=(S2*S1*SIN(A1))/2
4000 CO=CO+1
4010 UNTIL CO=4
4020 ENDPROC
4030 REM##### TEST #####
4040 PROCSETSQR
4050 EROR=0:CO=0
4060 PROCGETSQR
4070 IF EROR =1 THEN 4030
4080 PROCCALSQR
4090 PRINTTAB(0,20)STRING$(120," ")
4100 PRINTTAB(0,19)" HYP=";HYP
4110 PRINTTAB(14,19)" ADJ=";ADJ
4120 PRINTTAB(27,19)" OFF=";OFF
4125 A1=DEG(A1);PROCROUND(A1);A1=AX
4130 PRINTTAB(0,20)" A1=";INT(A1);" D ";INT(60*(A1-INT(A1))); " M "
4135 A2=DEG(A2);PROCROUND(A2);A2=AX
4140 PRINTTAB(14,20)" A2=";INT(A2);" D ";INT(60*(A2-INT(A2))); " M "
4150 PRINTTAB(27,20)" AREA=";AREA;
4160 GOTO 5000
4170 REM#####
4180 PROCSETOBT
4190 EROR=0:CO=0
4200 PROCGETOBT
4210 IF EROR=1 THEN 4170
4220 CO=0
4230 PROCCALOBT
4240 PRINTTAB(1,18)STRING$(120," ")
4250 PRINTTAB(0,21)STRING$(80," ");;
4260 PRINTTAB(0,18)" S1=";S1
4270 PRINTTAB(14,18)" S2=";S2
4280 PRINTTAB(27,18)" S3=";S3
4290 PRINTTAB(0,19)" P1=";P1
4300 PRINTTAB(14,19)" P2=";P2
4310 PRINTTAB(27,19)" H=";H
4315 A1=DEG(A1);PROCROUND(A1);A1=AX
4320 PRINTTAB(0,20)" A1=";INT(A1);" D ";INT(60*(A1-INT(A1))); " M "
4325 A2=DEG(A2);PROCROUND(A2);A2=AX
4330 PRINTTAB(14,20)" A2=";INT(A2);" D ";INT(60*(A2-INT(A2))); " M "
4335 A3=DEG(A3);PROCROUND(A3);A3=AX
4340 PRINTTAB(27,20)" A3=";INT(A3);" D ";INT(60*(A3-INT(A3))); " M "
4345 PA1=DEG(PA1);PROCROUND(PA1);PA1=AX
4350 PRINTTAB(0,21)" PA1=";INT(PA1);" D ";
    INT(60*(PA1-INT(PA1))); " M "
4355 PA2=DEG(PA2);PROCROUND(PA2);PA2=AX
4360 PRINTTAB(14,21)" PA2=";INT(PA2);" D ";
    INT(60*(PA2-INT(PA2))); " M "
4361 PRINTTAB(27,21)" AREA=";AREA;
4370 GOTO 5000
4380 REM
4390 DEF PROCBALANCE
4400 REM## TEST FOR ONE SIDED INPUTS£
4410 IF CO=3 AND A3=0 AND PA2=0 AND S3=0 AND P2=0 AND A2=0 THEN
    CO=CO-1
4420 IF CO=3 AND A1=0 AND PA1=0 AND S1=0 AND P1=0 AND A2=0 THEN
    CO=CO-1
4430 ENDPROC
4440 DEF PROC SOUND
4441 X=1
4442 FOR LOOP=150 TO 0 STEP-20
4450 SOUND 1,-9,LOOP,X
4460 FOR DELAY=1TO20:NEXT

```



colours and until they return to their original colour, blue, you may eat them. So altogether, I think it is an excellent game but I do have one complaint, you cannot use any joystick which is rather disappointing as I always thought that pakman and a joystick went together.

LETTERS

Thanks for the EUG., Iss 4: any chance of Issues 1-3???. My first perusal of the interior prompts me to write the following observations:-

1. Sorry I missed the competition, I got the mag too late. (Good job, that, I already have FORTH - haven't a clue how (or on what) to use it).
2. I've got Hitchhiker. How do I move away from the flying brick which kills me?? - or stop the bulldozer. The towel will not wrap around the head - its killing, I tell you. (By the way, I had a sector error on a disc with this on; the CP/M routine FINDBAD worked; but I wish someone would re-write it for the Einstein: offers, anywhere??).
3. Bigsums. What a great idea. I'll do this with my grandson - or maybe his Dad. Seriously, though: What a quaint layout of columns!!.
4. Paul Burgess's SETxxBD is very good, as one might expect from this prolific and skillful user. However; Please would you ask some knowledgeable type to explain just what gives in the world of Assemblers, etc. I have the DevPack, but do not know what to do with it, or how, for that matter. The instruction book might just as well be written in Chinese. What does a Linker do, (L80). Or a Macro Assembler (M80). Why do my futile attempts to use C or Pascal, (or for that matter Microsoft Basic), result in disaster for lack of a something which everybody else takes for granted. What is meant by a .REL, or .PRN file, etc., etc., etc.,??.
5. Patterns of Chaos. Not understood at all, (I am tempted to ask who or what is Julia Mandelbrot, but I suspect that it is perfectly valid - for something; I guess I'll try it and see what it does - at least it is in some form of BASIC!!
6. The World. As I missed Pt 1, I'll skip over it for the moment.
7. System 5. Now I may be able to do something with this. You see, I have one, and therefore offer my experience so far:-

Programs written under DOS 1.11, (WDPRO, v 2.4 or 2.61) will not work well, (something to do with interrupts and quaint, non-standard calls, I understand). The Cracker (CR80), suffers a similar fate. Wordstar, (the version for Einstein under DOS 1.31) will not even load properly under DOS 2.02. (In fact, you get thrown back to the MOS register table, and it is cold re-light time again). The TATUNG Sales Ledger/Invoice Ledger will not work under the new basic 5.05, either. Crystal are aware of the problem, and have said to me that they are contacting the firms concerned. How much help they get from Tatung will depend a lot on whether Dave Bell is busy. (He is a very helpful guy, surrounded by an apparent indifference that is almost tangible). Whether Kuma (WDPRO), or Soft-tech (CR80) will spend much time on the problem is also debatable - we live in a commercial world.

However, my new 40 track DS drive, (Epson SD321) works well under the new regime, giving 390k indicated available space. And at least one version of Microsoft runs; v5.21. So does the compiler, (what little sense I can make of it. It's all to do with L80 again!). Several of the public domain CP/M software works, (except FINDBAD), more's the pity, and I have found no problem, so far, with the InfoCom games, (apart from the aforementioned brick). As I do not write great wads of LDA or DEFFh, I will pass on XED. I tried it and it is quite easy to write a READ.ME file with it. There was mention of a problem with DOS 2.02 and FORTH. As I do not use it, I'll pass on that, too.

XSM and such like assemblers are a closed book to me, even if I do have ZEN, DevPack, ASM and several others: (you may wonder why I have them; I understand them to be quite useful - so I live in hopes that one day....).

8. Reviews - great. And thanks for explaining about the BRAIN. I did not understand the "framework"/"build" article, but then, I do not make a habit of translating Locomotive to Xtal.

9. VRAM scroll. I'd like to use this one, but do not understand what is written in. Is it in Assembler, Pascal, C, (or for that matter AFL),?. It looks like both to me. Cursor addressing did not emerge until DOS 1.2/1.21 became available. I doubt that earlier version/s of the DOS used the code.

10. Thanks for the Hints and Tips. Very useful. It is a handy way of experimenting with IN/Out codes - without the need for LEDs hanging out from the back.

11. Letters. Can I have a copy of the routines for England/Ireland map? I may have a use for them, (Amateur Radio). I may have some information for Steve Carter about linking Assemble code programs in Basic, but will have to check my facts first. I will send you the draft when it is finished.

12. Lastly, (phew)What are the pretty pictures at the back - more Julia??

Yours sincerely  
(signed) D.R.Coomber,1104.

#### ANSWERS

Taking your points one by one:

Back issues available at '1.25

1)Yes I've got a copy of FORTH too and I can't get the hang of it! ANY OF YOU OUT THERE CARE TO WRITE A COLUMN ON FORTH FOR US ALL TO UNDERSTAND!!!!

2)Over to you Tony. See our Adventurers column.

3)Whoops! Yes this was a departure from the normal layout, but we promise we won't do it again.

4)Watch this space, or more precisely this mag. We are publishing a Machine Code Tutorial series for the beginner and hopefully if it is supported by yourselves we will get around to Macro assemblers and Linkers and .REL and .PRN files.

5)We don't understand Patterns of Chaos but they do produce pretty patterns and ther alot of you out there for whom they are an all consuming hobby. (By the way if you saw the last of the MICRO LIVE programs you will know what I am getting at.

6)See the top line re back issues.

7)Given time you will sort out the wheat from the chaff and get lots of use out of what you do have, hang on in there.

8)Due to the large number of users of all sorts of machines there are some terrific utilities being written on other micros and sometimes the quickest way to get them on your own machine is to convert.

9)Video Ram Scroll. Written in a combination of PASCAL and Machine Code. Cursor addressing actually became available with MOS 1.2 not DOS 1.2 .

10)It's a pleasure, thanks to the guy who asked the question in the first place.

11)The back issues are available as previously stated

12)Phew! YES And thanks for a super letter, i'ts great to know that our efforts are being taken seriously.

# PEACH COMPUTERS LTD.

Co. Reg. No: 73767  
Vat Reg. No: 406 5533 66

192 Greenock Road,  
Largs,  
Ayrshire,  
Scotland, U.K.  
Tel: 0475-673766

KEITH STOKES (EUG)  
HILLCROFT CODMORE HILL  
PULBOROUGH  
WEST SUSSEX RH20 1BQ

28/04/86

## DIFFICULTY IN INSERTING AND EXTRACTING A DISC

Dear Keith,

We have recently sorted out a strange problem which appeared in one of our Einstein disc drives and we thought that the problem and its solution might be of interest to other EUG readers.

What we found was that it was becoming more and more difficult to insert and remove discs for the (second) disc drive in one of our Einsteins. The problem got steadily worse over a period of a few months until eventually we really had to do something about it as it looked as if it would soon be impossible to get a disc out even if we did manage to get it in.

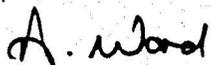
So we opened up the case and removed the drive unit from its metal can. Our immediate impression was very favourable as the drive seems to have been very well designed and put together as a precision unit. We therefore proceeded with some caution and it was about an hour before we figured out what was wrong. The trouble lay in the length of one of a pair of small alignment pins which pass through the two holes at the far away end of each disc. As we pushed the disc in the edge of the disc was butting against the side of one of the pins instead of passing over it prior to final alignment when the disc drops over the two pins.

It seems to be the case that the pin is actually screwed into its slightly wider base stud and had managed to unscrew itself very slightly over a long period - probably as a result of torques due to the spin of the disc.

After a certain amount of cautious fiddling around we managed to get a grip of the smooth shiny pin and screw it back into its base stud. As it was screwed in we could see that its height became smaller by about 2 mm and eventually looked the same as the other pin of the pair. When we put it all back together again we were pleased to find that we could now insert and remove discs normally. Incidentally the first sign of trouble (many months ago) was that the disc did not jump out smartly when the release was pushed. We hope this description is of use to anyone who encounters this problem.

With kind regards we remain

Yours sincerely



Agnes and Alastair Ward

# SCREENSOFT



## FLIGHT SIMULATION

FOR USE WITH EINSTEIN MICROCOMPUTER

SCREENSOFT REAL TIME SIMULATION WITH 3D GRAPHICS.

USES MASSIVE 64,000 X 64,000 LONGITUDE AND LATITUDE FLYING AREA, MAKING EACH FLIGHT COMPLETELY DIFFERENT. DEVELOPED UNDER PILOT INSTRUCTION TO GIVE REALISTIC FLIGHT EFFECT. VIEW THROUGH COCKPIT GIVES MOVING 3D GRAPHICS.  
COMPREHENSIVE INSTRUMENT PANEL WITH MOVING NEEDLE METERS AND DIGITAL DISPLAYS.  
3D LANDMARKS YOU CAN FLY AROUND.  
THREE RUNWAYS AVAILABLE FOR REFUELLING, TAKE OFF AND LANDING....GROUND AND LAND-MARK ORIENTATION, CORRECT WITH ALL FLYING ATTITUDES (ROLLS ETC.)  
HOURS OF ABSORBING SKILLFUL INVOLVEMENT WITH ITS SUPERB SIMULATION.... COMPLETE WITH FULLY DETAILED CHART OF LANDMARKS AND AIRFIELDS.

KEYBOARD OR JOYSTICK OPERATION

1986

COPYRIGHT © MYRIDDIN SOFTWARE LTD. Swindon, Wilts SN5 8BG  
COPYRIGHT © SCREENSOFT LTD. Northwood, Middlesex  
The Program on this disk is subject to Copyright and it is illegal to reproduce them in any form without the express prior written consent of Myriddin & Screensoft Software.

1st, P.O. Box 61, Swindon, Wilts SN5 8BG  
SCREENSOFT LIMITED, Northwood, Middlesex.

# SCREENSOFT



## HELLMATTS GRIDDO FUNCTION MACHINE

FOR USE WITH EINSTEIN MICROCOMPUTER

AN EXCITING COMPENDIUM OF INTERACTIVE EDUCATIONAL PROGRAMS FOR THE EINSTEIN COMPUTER FROM SCREENSOFT.

All these programs are suitable for children up to the age of Thirteen years. Simple cursor control excludes the giving of ambiguous answers and therefore makes these programs ideal for younger children as well.  
The programs cover the four disciplines of basic arithmetic enabling the programs to be used time and time again as the child progresses.  
Help will be given if the child answers incorrectly on two successive occasions.  
All instructions are given within the programs.  
TO LOAD  
(1) PUT YOUR SYSTEM DISK IN DRIVE 0  
(2) LOAD XBAS  
(3) REPLACE SYSTEM DISK WITH HELLMATTS DISK AND TYPE "N MENU" AND THEN HIT ENTER KEY.

1986

COPYRIGHT © KERIAN (U.K.) LTD Gibburn Road, Hestle HU13  
The Programs on this disk are subject to Copyright and it is illegal to reproduce them in any form without the express prior written consent of KERIAN(U.K.)LTD.

SCREENSOFT LIMITED, Main Avenue, Moor Park, Northwood, Middlesex.

# SCREENSOFT



## TURBO CHESS & DRAUGHTS

FOR USE WITH EINSTEIN MICROCOMPUTER

SCREENSOFT TURBO CHESS & DRAUGHTS

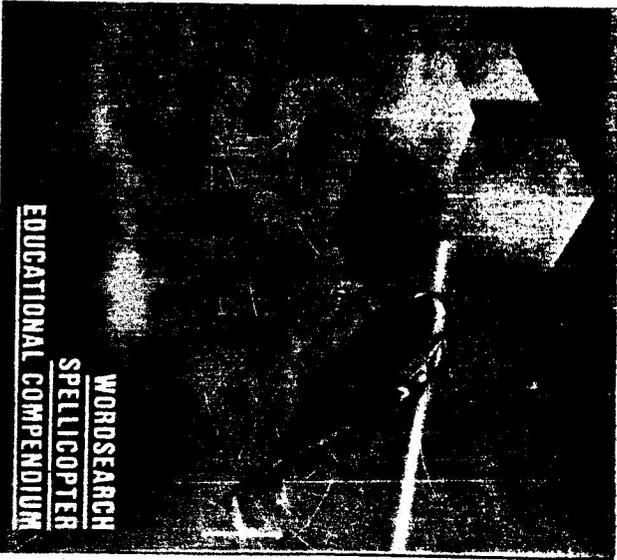
Turbo Chess represents a 'state-of-the-art' chess program for the Einstein Computer. Features of this fine program include:  
\* INFINITE LEVELS OF PLAY  
\* COMPREHENSIVE ANALYSE ROUTINE  
\* LOAD A PREVIOUSLY SAVED GAME  
\* SELECTABLE COLOUR OPTIONS FOR BOARD AND PIECES  
\* PRINTER DUMP INCLUDED  
\* COMPUTER RECOMMENDED MOVE OPTION  
\* RECALL ALL MOVES TO DATE ROUTINE  
In line with screensoft's policy to give maximum value for money, Kerian's popular board game Draughts is included on side 'B' of the program disk.  
FULL LOADING AND PLAYING INSTRUCTIONS FOR THESE EXCITING GAMES ARE ON THE REVERSE OF THIS CARD.

1986

COPYRIGHT © KERIAN (U.K.) LTD Gibburn Road, Hestle HU13 9HZ  
The Programs on this disk are subject to Copyright and it is illegal to reproduce them in any form without the express prior written consent of KERIAN(U.K.)LTD.

SCREENSOFT, Main Avenue, Moor Park, Northwood, Middlesex, England.

# SCREENSOFT



## WORDSEARCH SPELLICOPTER

FOR USE WITH EINSTEIN MICROCOMPUTER

AN EXCITING COMPENDIUM OF INTERACTIVE EDUCATIONAL PROGRAMS FOR THE EINSTEIN COMPUTER FROM SCREENSOFT.

All three programs are suitable for children up to the age of Thirteen years. Simple cursor control excludes the giving of ambiguous answers and therefore makes these programs ideal for younger children as well.  
The programs contain a comprehensive word set, from three letter words through to advanced spelling techniques. Enabling the programs to be used time and time again as the child progresses.  
Help will be given if the child answers incorrectly on two successive occasions.  
All instructions are given within the programs.  
TO LOAD  
(1) PUT YOUR SYSTEM DISK IN DRIVE 0  
(2) LOAD XBAS  
(3) REPLACE SYSTEM DISK WITH SPELL 'N' BUILD DISK AND TYPE "W MENU" AND THEN HIT ENTER KEY.

1986

COPYRIGHT © KERIAN (U.K.) LTD Gibburn Road, Hestle HU13 9HZ  
The Programs on this disk are subject to Copyright and it is illegal to reproduce them in any form without the express prior written consent of KERIAN(U.K.)LTD.

SCREENSOFT LIMITED, Main Avenue, Moor Park, Northwood, Middlesex.