

JUNE '86

8



UK EINSTEIN USER GROUP  
NEWSLETTER

**EWUG**

Issue Number Eight June 1986

EDITORIAL

I said it would not last, and it didn't, still the quality makes up for any other faults we may have ( I hope).

First of all I would like to thank all who were able to make our open evening at Oxshot. On the basis of the enthusiasm shown at the meeting and on such short notice we would like to announce the Einstein Event of the year.

**The  
UKEUG  
National  
Einstein  
Show**

To be held at the National Motorcycle Museum on Saturday the 8th November.

Entrance is free and we will be there from 10am to 10pm.

Make a note of the date and we hope to see you there.

Further details will be in later issues.

Now on to the competition.

We had only ONE correct answer from Mr T. McBride who even now is working on a masterpiece fit to grace these pages.

Solution to the competition:-

```
10 FOR K=1 TO 48:REM THERE ARE 48 NUMBERS
20 READ J:REM GET EACH NUMBER IN TURN
30 PRINT CHR$(J+K);:REM PRINT THEM SEQUENTIALLY
40 NEXT
50 STOP
60 DATA 82,70,70,79,27,61,72,69,71,59,73,61,71,59,64,62
70 DATA 15,64,50,61,64,51,59,45,58,06,40,44,53,06,09,42,10
80 DATA 41,06,-04,47,41,-07,43,38,34,43,25,-13,27,37,-02
```

You may have noticed that the quality of print is a whole lot better in this issue, I have splashed out and purchased a new printer. So the main review this month will be running throughout the issue demonstrating some of the capabilities of the PANASONIC KX-P1080 available from SCREENS at only £199.95.

Now some software news, Syntaxsoft have a new Prestel access program which gives a full colour display and is compatible with the Tandata or Miracle WS 2000.

Also to be released soon is Highway Encounter which is a game of skill and dexterity with superb graphics.

Syntaxsoft are on 0282 698848.

Coming soon from Screens a Typing Tutor program with graded exercises at £14.95.

And from Tatung a book "The Einstein Primer" by David Bell written for beginners.

~~~~~



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

DEADLINE

=====

Ok, before I start I must point out that I am not really a fan of 'Whodunnits'. Things like Agatha Christie novels and Cluedo usually leave me cold. 'What is he blabbering on about?' do I hear you mutter? Well, Deadline ain't no ordinary adventure, it's the first of the three current 'Mystery Series' by Infocom.

After the untimely demise of a Mr Marshall Robner by 'a self administered overdose', your Chief of Detectives asks you to investigate further into the case. You begin outside the Robner house. Exploring the gardens reveals a couple of unexplained holes in the ground, and an eccentric old gardener. Inside the house you will meet the family and friends of the deceased, and the cook, who is a source of all kinds of interesting gossip. You can ask questions of any of the inhabitants of the house, although some may not exactly be helpful. You should also find lots of interesting clues around the house, such as a telephone call to Mrs Robner, or a mighty suspicious letter which arrives at around 11.00 in the morning.

As well as the normal commands like LOOK, EXAMINE etc, this game understands a number of rather specialised commands. You can ACCUSE suspects, ANALYSE objects and substances, ARREST suspects, FINGERPRINT objects and use many other unusual but useful commands to help you solve your investigation. But you better get it right, otherwise you will end up losing your job. And you better get it quickly, because you only have 24 hours (game time, not real time) to solve the investigation.

Deadline is a fascinating game, the level of interaction with characters encountered in the game is very impressive, and some of the responses are unexpected and funny, for example, try to kiss one of the game characters. Although it is easily up to Infocom's usual professionalism I'm not sure if this game will appeal to many adventure players. If you are a fan of whodunnits then this is probably the game for you. As I've said, I don't really like murder and mystery books, and although the game is unusual and interesting enough, I didn't enjoy playing it as much as other adventure games.

HINT

In the April Newsletter D.R. Coomber wrote in with a problem with bricks and bulldozers. Well D.R., I have some good news and some bad. The good news is that below is an explanation of how to get round your problems (so if any of you want to solve these problems yourselves, skip past the next bit). The bad news is that this is just the start of Hitchhiker, it gets nastier (but funnier too).

Ok, so far you have managed to get out of bed and cure your hangover, only to discover that there is a large and unfriendly bulldozer advancing on your home. What you must do next is lie down in front of the bulldozer (yes, I know that nobody in their right mind would lie in front of a dirty great bulldozer, but nobody in their right mind wrote this game!). This will do the trick. The brick kills you because you let the bulldozer knock your home down, so getting up and allowing the bulldozer to continue will result in the same deadly outcome. Obviously you can't spend the whole of the game lying in front of a bulldozer, so you must find a way to keep the bulldozer where it is. This involves the towel (believe it or not). However the solution to this puzzle is not an obvious one. Think about the situation you find yourself in when Ford arrives on the scene, what would your reaction really be?

I hope this helps D.R., if anybody has any more problems with this or any other game out for the Einstein, either write to the Secretary of

the group, or drop me a line (my address was at the back of the March issue of the Newsletter). If you want a reply before the next issue of the Newsletter please include a SAE.

- Tony Stansfield.

~~~~~

MACHINE CODE PROGRAMMING FOR THE BEGINNER  
(or assembly language for the initiated)  
(using the BBCBASIC assembler)  
By CHRIS GILES

No adverse comments this month. Its either getting better or lots of you have dropped out. I have had a request on a routine (to pick up multiple key presses) so I will try to fit that in. Any other suggestions greatly received.

Now on to disc access. To do this from scratch would involve many issues of the newsletter so lets do it the easy way. Look at the back of your MOS/DOS handbook at the section on CP/M compatibility.

Well go on, look!! Does it make any sense? Not to me at first viewing but with a little assistance it clicked into place. There are 40 functions available from CP/M and the way to access them is to put the function number into the 'C' register and do a call to location 0005. Simple! Well nearly. If the routine needs information passed to it this must be passed down by some of the other registers and if the routine is to return information this will be passed up by those same registers.

I have used our routine from last week to do the Menu so that much should be familiar and lines 320 to 660 should need no explanation. Lines 120,130 and 140 are constants used in the program. CP/M is almost self explanatory. It is the location to be called to execute a CP/M function. FDESCA is the area of memory we are setting aside for the File Descriptor block. It is explained fairly well in the MOS DOS book. BUFFER is the area of memory that we will be using for the input buffer in CP/M.

There are lots of notes in the listing which explain what each section does but there are some bits that need further explanation. Here goes. Lines 1100 to 1120 do the same thing as the clear screen routine in lines 320 to 340 but this is the CP/M version. In the CP/M version the character to be printed is put into the 'E' register and instead of an RST 8 with a DEFB we load the 'C' register with 2 which is the print a character function number and do a CALL to location 0005. Lines 1160 to 1180 takes whatever is typed in at the keyboard and stores it in the area we have designated as the input buffer. The first two characters of the input buffer are used by CP/M and I haven't figured out what for yet. Any info will be welcome!! Don't forget I am learning this as I am going along.

The routine from lines 1220 to 1270 is rather neat and needs some extra explanation. It revolves around the LDIR instruction. LDIR stands for Load Increment and Repeat. What it does is take the contents of the location addressed by HL and puts them in the location addressed by DE. Then it Increments HL and DE decrements BC and if BC is NOT zero it Repeats. If you point HL at one area of memory and DE at another area of memory then the LDIR instruction will move BC bytes from HL to DE. Remember it, it will prove very useful.

The next clever instruction is the DJNZ. This stands for Decrement Jump Non Zero. What this one does is Decrement B, and if it is Not Zero Jump to the label.

Well I hope that you find this useful. Next month, as I said earlier, I hope to have a routine to detect multiple key presses, very



useful for games programs!!

Wish me luck 'cause I think it is likely to be the hardest one I have looked at yet.

```

120 CPM=0005           ;Location to call for CP/M
130 FDESCA=34000       ;Location we are using for the File Descriptor
140 BUFFER=34200       ;Location we are using for buffer
320 .START LD A,12
330 RST 8
340 DEFB 158
350 .MENU LD HL,MMESS1
360 CALL PRM
370 LD HL,MMESS2
380 CALL PRM
390 LD HL,MMESS3
400 CALL PRM
410 LD HL,MMESS4
420 CALL PRM
430 LD HL,MMESS5
440 CALL PRM
450 .getkey RST 8
460 DEFB 156
470 CP 49
480 CALL Z,create
490 CP 50
500 CALL Z,erase
510 CP 51
520 CALL Z,open
530 CP 52
540 CALL Z,close
550 CP 53
560 JP Z end
570 JP MENU
580 .PRM LD B,(HL)
590 INC HL
600 .PRM1 LD A,(HL)
610 RST 8
620 DEFB 158
630 INC HL
640 DEC B
650 RET Z
660 JP PRM1
670 .create CALL getnam ;WE NEED A NAME
680 CALL OPNCPM         ;MUST CHECK TO SEE IF IT EXISTS FIRST
690 JR NZ FILEXI        ;IF IT DOES GO TO ERROR MESSAGE
700 CALL CRECPM         ;OTHERWISE CREATE IT
710 JR Z NOSPACE        ;BUT IF THERE IS NO SPACE GO TO ERROR MESSAGE
720 RET                 ;FINISHED
730 .erase CALL getnam ;WE NEED A NAME
740 CALL ERACPM         ;ERASE IT
750 RET                 ;FINISHED
760 .open CALL getnam   ;WE NEED A NAME
770 CALL OPNCPM         ;OPEN IT
780 JR Z NOFILE         ;BUT GO TO ERROR IF FILE DOES NOT EXIST
790 RET                 ;FINISHED
800 .close CALL getnam  ;WE NEED A NAME
810 CALL CLOCPM         ;CLOSE IT
820 RET                 ;FINISHED
830 .end RET;THIS ONE RETURNS YOU TO BASIC
840 .FILEXI LD HL,ERR1   ;POINT TO ERROR MESSAGE
850 JP PRM               ;JUMP TO IT BECAUSE THE PRINT A MESSAGE
860 .NOSPAC LD HL,ERR2   ;

```



```

870 JP PRM ;ROUTINE ENDS WITH A 'RET' SO IT WILL
880 .NOFILE LD HL,ERR3 ;
890 JP PRM ;GO BACK TO THE MENU ROUTINE.
900 .INVAL LD HL,ERR4 ;
910 JP PRM ;CLEVER ISN'T IT??
920 .OPNCPM LD DE,FDESCA ;GET FILE DATA
930 LD C,15 ;FUNCTION NUMBER TO OPEN FILE
940 CALL CPM ;DO IT
950 CP 255 ;SET ZERO FLAG IF NO FILE
960 RET ;FINISHED
970 .CLOCPM LD DE,FDESCA ;GET FILE DATA
980 LD C,16 ;FUNCTION NUMBER TO CLOSE FILE
990 CALL CPM ;DO IT
1000 RET ;FINISHED
1010 .CRECPM LD DE,FDESCA ;GET FILE DATA
1020 LD C,22 ;FUNCTION NUMBER TO CREATE FILE
1030 CALL CPM ;DO IT
1040 CP 255 ;SET ZERO FLAG IF NO SPACE IN DIRECTORY
1050 RET ;FINISHED
1060 .ERACPM LD DE,FDESCA ;GET FILE DATA
1070 LD C,19 ;FUNCTION NUMBER TO ERASE FILE
1080 CALL CPM ;DO IT
1090 RET ;FINISHED
1100 .getnam LD E,12 ;CLEAR SCREEN CHARACTER, 'E' REGISTER USED IN CP/M
1110 LD C,2 ;FUNCTION NUMBER TO PRINT A CHARACTER TO THE SCREEN
1120 CALL CPM ;DO IT
1130 LD HL,PROMPT ;GET MESSAGE
1140 CALL PRM ;PRINT IT
1150 CALL CLRBUF ;CLEAR INPUT BUFFER
1160 LD DE,BUFFER ;POINT CP/M AT BUFFER
1170 LD C,10 ;FUNCTION NUMBER TO GET A LINE OF INPUT
1180 CALL CPM ;DO IT
1190 CALL CLRFBDB ;CLEAR FILE DESCRIPTOR BLOCK
1200 CALL CHECK ;CHECK FOR VALID FILENAME
1210 RET ;FINISHED
1220 .CLRBUF LD HL,BUFFER ;POINT TO START OF BUFFER
1230 LD (HL),32 ;PUT A SPACE IN IT
1240 LD DE,BUFFER+1 ;POINT TO THE NEXT BUFFER POSITION
1250 LD BC,128 ;128 POSITIONS ALLTOGETHER
1260 LDIR ;SEE TEXT FOR EXPLANATION
1270 RET ;FINISHED
1280 .CLRFBDB LD HL,FDESCA ;SAME
1290 LD (HL),0 ;AS
1300 LD DE,FDESCA+1 ;ABOVE
1310 LD BC,36 ;BUT
1320 LDIR ;TO
1330 RET ;FILE DESCRIPTOR BLOCK
1340 .move LD A,(HL) ;GET THE BYTE POINTED TO
1350 AND 127 ;MAKE SURE IT IS NOT GRAPHICS
1360 CP 46 ;IS IT A FULL STOP
1370 JR NZ,TRANS ;IF NOT TRANSFER TO FILE DESCRIPTOR BLOCK
1380 .PAD DEC HL ;IF IT IS MOVE BACK ONE POSITION
1390 LD A,32 ;USE A SPACE
1400 .TRANS LD (DE),A ;AND PUT IT IN THE FILE DESCRIPTOR BLOCK
1410 INC HL ;ADVANCE A BYTE (IF WE HAVE PUT IN A SPACE BECAUSE
1420 INC DE ;NEXT IN FDESCA (OF THE FULL STOP THEN ADVANCE TO
1430 DJNZ move ;SEE TEXT (IT. FILLS ALL EIGHT POSITIONS
1440 INC HL ;JUMP PAST FULL STOP
1450 RET ;FINISHED
1460 .check LD HL,BUFFER+2 ;WE DONT WANT THE FIRST TWO CHARACTERS
1470 LD DE,FDESCA ;THE FIRST CHARACTER
1480 LD A,0 ;OF THE FILE DESCRIPTOR
1490 LD (DE),A ;MUST BE ZERO

```

```

1500 INC DE ;NEXT CHARACTER
1510 .check1 LD B,8 ;EIGHT LETTERS IN FILENAME?
1520 CALL move ;MOVE EIGHT OR PAD WITH SPACES
1530 LD B,3 ;EXTENT IS THREE CHARACTERS
1540 CALL move ;MOVE THREE OR PAD WITH SPACES
1550 RET ;FINISHED
1560 .MMESS1 DEFB 20 ;MESSAGES FROM HERE ON
1570 DEFM "1 CREATE a File"
1580 DEFB 10
1590 DEFB 13
1600 .MMESS2 DEFB 19
1610 DEFM "2 ERASE a File"
1620 DEFB 10
1630 DEFB 13
1640 .MMESS3 DEFB 18
1650 DEFM "3 OPEN a File"
1660 DEFB 10
1670 DEFB 13
1680 .MMESS4 DEFB 19
1690 DEFM "4 CLOSE a File"
1700 DEFB 10
1710 DEFB 13
1720 .MMESS5 DEFB 11
1730 DEFM "5 END "
1740 DEFB 10
1750 DEFB 13
1760 .ERR1 DEFB 14
1770 DEFM "File exists "
1780 DEFB 10
1790 DEFB 13
1800 .ERR2 DEFB 11
1810 DEFM "No Space "
1820 DEFB 10
1830 DEFB 13
1840 .ERR3 DEFB 10
1850 DEFM "No File "
1860 DEFB 10
1870 DEFB 13
1880 .ERR4 DEFB 18
1890 DEFM "Invalid Filename"
1900 DEFB 10
1910 DEFB 13
1920 .PROMPT DEFB 41
1930 DEFM "Please enter a valid Filename (No Drive)"
1940 DEFB 10
1950 DEFB 13
1960 RET

```

~~~~~

PROGRAM BUILDER  
part two  
by  
Vic Day

Last month we had the details and core listing, in this month's article we have the modified FRAME listing and two source listings.  
(and Vic thought I had forgotten these .ED )

FRAME2

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1 PRINT "THIS IS PROGRAM "; 2 PRINT " F R A M E " 3 GOSUB 16000 4 PRINT "FOR CONSTRUCTION OF PROGRAMS" 5 PRINT " TYPE '?' FOR INSTRUCTIONS" 7 ON ERR GOTO 290 100 REM the body of FRAME is lines 100 - 500 (do not alter) 110 L\$="TSX?ABCDEFGHJKLMN" 120 M\$=MID\$(L\$,5,LEN(L\$)) 130 M8=1:M9=0 150 PRINT "WHAT NEXT "; 160 M9=LEN(L\$)+2 170 ON M8 GOTO 480,320 180 FOR N90 = 1 TO LEN(L\$) 190 IF LEFT\$(K\$,1)=MID\$(L\$,N90,1) THEN M9=N90:GOTO 210 200 NEXT N90 210 IF M9&gt;LEN(L\$) THEN PRINT "Unknown command":GOTO 150 250 IF M9&lt;5 THEN ON M9 GOTO 290,300,310,500 260 ON M9-4 GOSUB 1000,2000,3000,4000,5000,6000,7000,8000,9000,     10000,11000,12000,13000,14000 280 GOTO 150 290 CLS:END 300 PRINT "YOUR NEW SEQUENCE PLEASE ";:INPUT M\$:M9=0 310 M8=2 320 M9=M9+1 330 IF M9&gt;LEN(M\$) THEN M9=0:M8=1:GOTO 170 340 K\$=MID\$(M\$,M9,1) 350 PRINT TAB(10);M\$ 360 PRINT TAB(M9+10);"" 470 GOTO 180 480 INPUT K\$ 490 GOTO 180 500 CLS:PRINT "USE:-" 510 PRINT "A to M For a single process" 910 PRINT " ? For Instructions" 920 PRINT " S To set up a new Sequence" 930 PRINT " X To eXecute the sequence"                 </pre> | <pre> 940 PRINT " T To Terminate" 990 GOTO 150 1000 PRINT "PROCESS A" 1900 RETURN 2000 PRINT "PROCESS B" 2900 RETURN 3000 PRINT "PROCESS C" 3900 RETURN 4000 PRINT "PROCESS D" 4900 RETURN 5000 PRINT "PROCESS E" 5900 RETURN 6000 PRINT "PROCESS F" 6900 RETURN 7000 PRINT "PROCESS G" 7900 RETURN 8000 PRINT "PROCESS H" 8900 RETURN 9000 PRINT "PROCESS I" 9900 RETURN 10000 PRINT "PROCESS J" 10900 RETURN 11000 PRINT "PROCESS K" 11900 RETURN 12000 PRINT "PROCESS L" 12900 RETURN 13000 PRINT "PROCESS M" 13900 RETURN 14000 PRINT "PROCESS N" 14900 RETURN 15000 REM SET COMMON SUBROUTINES HERE 15900 RETURN 16000 REM SET DEFINED FUNCTIONS HERE 16900 RETURN 17000 REM DATA STATEMENTS HERE 17900 RETURN 20000 END                 </pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

SOURCE FILES

SOURCE 1

```

20313 UT
TO TEST 'BUILDER'
DIM A(15), B(15)
K=10
N=0
END
INPUT
TO LOAD YOUR DATA
FOR N=1 TO K
INPUT "A VALUE PLEASE ";V
                
```



```

A(N)=V
B(N)=V*V
NEXT N
END
CALCULATE
TO PRODUCE PRIMARY STATS
S=0
S2=0
FOR N=1 TO K
S=S+A(N)
S2=S2+B(N)
NEXT N
END
LIST
TO SEE THE RESULTS
FOR N=1 TO K
PRINT N, A(N),B(N)
NEXT N
PRINT
PRINT "SUM = ";S
PRINT "MEAN = ";S/(N-1)
PRINT
PRINT "VARIANCE IS ";S2/(N-1)-(S/(N-1))^2
END
EDITOR
TO ENABLE YOU TO ALTER A VALUE
INPUT "PLEASE GIVE Nr. OF ITEM ";L
INPUT "PLEASE GIVE NEW VALUE ";V
A(L)=V:B(L)=V*V
END
ATEST
MOD ARITHMETIC
DEF FNR(X)=INT(X*RND(1))
DATA 5,6,7,0
RESTORE
READ N
IF N=0 THEN RETURN
FOR Z=1 TO 6
A=FNR(N)
B=FNR(N)
X=A+B
PRINT A+"B"= ";
SUBA
I=INT(X/N)
W=X-I*N
PRINT W"MOD ";N;
RTN
GOSA
X=A*B
PRINT A*"B"= ";
GOSA
PRINT "      "
NEXT Z
PRINT
GTO-14
END
FINISH

```

SOURCE 2

```

21586 GAGE
TO CALCULATE MORTGAGES
END
PRINCIPAL
TO CALCULATE PRINCIPAL
INPUT "INTEREST RATE ";I:I=I/100
INPUT "TERM IN YEARS, MONTHS ";Y,M:TERM=Y+M/12
INPUT "MONTHLY REPAYMENT ";R:R=R*12
P=R/(I+(I/((1+I)^TERM-1)))
FMT 6,2
PRINT "PRINCIPAL AMOUNT ";P
END
INTEREST
TO CALCULATE INTEREST
INPUT "PRINCIPAL ";P
INPUT "TERM IN YEARS, MONTHS ";Y,M:TERM=Y+M/12
INPUT "MONTHLY REPAYMENT ";R:R=R*12
TEMP=0.1
REPEAT
I=TEMP
TEMP=R/P-(I/((1+I)^TERM-1))
UNTIL ABS(TEMP-I)<.000001
FMT 2,3
PRINT "INTEREST RATE ";(I*100)
END
DURATION
TO CALCULATE DURATION
INPUT "PRINCIPAL ";P
INPUT "INTEREST RATE ";I:I=I/100
INPUT "MONTHLY REPAYMENT ";R:R=R*12
TERM=LOG(P*I/(R-P*I)/LOG(1+I))+1/24
Y=INT(TERM):M=INT((TERM-INT(TERM))*12)
FMT 2,0
PRINT "TERM IS "Y" YEARS, "M" MONTHS"
END
REPAYMENT
TO CALCULATE REPAYMENTS
INPUT "PRINCIPAL ";P
INPUT "INTEREST RATE ";I:I=I/100
INPUT "TERM IN YEARS, MONTHS ";Y,M:TERM=Y+M/12
R=P*(I+(I/((1+I)^TERM-1)))/12
FMT 4,2
PRINT "MONTHLY PAYMENT IS ";R
END
FINISH

```

~~~~~

SPELL

A programme to help learning to spell.

```

1 REM >>>>SPELL by PETE HEFFERNAN<<<<
10 REM COMB
12 BCOL4:TCOL15
15 PRINTCHR$(20)
20 M=31:Y=31
30 SPRITEOFF
40 CLS
50 GOTO70
60 REM COMB
70 GOSUB5110
80 W=RND(22)

```

```

90 TCOL10
100 TCOL15
110 Z$=""
120 IFW= 0THENGOSUB4060:C$="DOG":Z=1
130 IFW= 1THENGOSUB4110:C$="DUCK":Z=1
140 IFW= 2THENGOSUB4160:C$="YACHT":Z=1
150 IFW= 3THENGOSUB4210:C$="CHAIR":Z=2
160 IFW= 4THENGOSUB4260:C$="HOUSE":Z=2
170 IFW= 5THENGOSUB4310:C$="VAN":Z=3
180 IFW= 6THENGOSUB4360:C$="BOAT":Z=1
190 IFW= 7THENGOSUB4410:C$="CAT":Z=1
200 IFW= 8THENGOSUB4460:C$="TREE":Z=2
210 IFW= 9THENGOSUB4510:C$="CUP":Z=2
220 IFW=10THENGOSUB4560:C$="BALL":Z=1
230 IFW=11THENGOSUB4610:C$="BED":Z=2
240 IFW=12THENGOSUB4660:C$="TANK":Z=1
250 IFW=13THENGOSUB4710:C$="CLOCK":Z=2
260 IFW=14THENGOSUB4760:C$="APPLE":Z=2
270 IFW=15THENGOSUB4810:C$="SUN":Z=2
280 IFW=16THENGOSUB4860:C$="CAR":Z=3
290 IFW=17THENGOSUB4960:C$="LIGHT":Z=4
300 IFW=18THENGOSUB5010:C$="CRANE":Z=2
310 IFW=19THENGOSUB5060:C$="BOOT":Z=2
320 IFW=20THENGOSUB4910:C$="FISH":Z=1
325 IFW=21THENGOSUB4951:C$="BUS":Z=1
330 GOTO460
340 REM CHECK ENTRY
350 IFZ$="BULB"ANDW=17THENC$="BULB"
360 IFZ$="MUG"ANDW=9THENC$="MUG"
370 IFZ$="SHIP"ANDW=6THENC$="SHIP"
380 IFZ$="LORRY"ANDW=18THENC$="LORRY"
385 IFZ$="TRUCK"ANDW=18THENC$="TRUCK"
390 IFZ$="SEAT"ANDW=3THENC$="SEAT"
400 IFZ$="BIRD"ANDW=1THENC$="BIRD"
410 IFZ$="BOAT"ANDW=2THENC$="BOAT"
420 IFZ$="SHIP"ANDW=2THENC$="SHIP"
430 IFC$=Z$THENGOSUB5220
440 IFC$=Z$THENCLS:GOTO80
450 IFC$<>Z$THENZ$="":SPEED150:PRINT@30,6;C$:SPEED255:GOTO460
460 REM ALPHA
500 CLS
510 C=1
520 IFC>41THENGOTO460
530 PRINT@0,1;"How do we spell this?";
550 A=INCH
560 IFA=32THEN340
570 IFA<65THENZ$="":GOTO500
580 IFA>90THENZ$="":GOTO500
590 IFA=65THENGOSUB860:Z$=Z$+"A"
600 IFA=66THENGOSUB960:Z$=Z$+"B"
610 IFA=67THENGOSUB1100:Z$=Z$+"C"
620 IFA=68THENGOSUB1200:Z$=Z$+"D"
630 IFA=69THENGOSUB1350:Z$=Z$+"E"
640 IFA=70THENGOSUB1450:Z$=Z$+"F"
650 IFA=71THENGOSUB1600:Z$=Z$+"G"
660 IFA=72THENGOSUB1750:Z$=Z$+"H"
670 IFA=73THENGOSUB1890:Z$=Z$+"I"
680 IFA=74THENGOSUB2000:Z$=Z$+"J"
690 IFA=75THENGOSUB2160:Z$=Z$+"K"
700 IFA=84THENGOSUB3260:Z$=Z$+"T"
710 IFA=76THENGOSUB2310:Z$=Z$+"L"
720 IFA=77THENGOSUB2460:Z$=Z$+"M"
730 IFA=78THENGOSUB2560:Z$=Z$+"N"

```



```

740 IFA=79THENGOSUB2660:Z$=Z$+"O"
750 IF A=80THENGOSUB2760:Z$=Z$+"P"
760 IFA=81THENGOSUB2910:Z$=Z$+"Q"
770 IFA=82THENGOSUB3060:Z$=Z$+"R"
780 IF A=83THENGOSUB3160:Z$=Z$+"S"
790 IFA=85THENGOSUB3410:Z$=Z$+"U"
800 IFA=86THENGOSUB3510:Z$=Z$+"V"
810 IFA=87THENGOSUB3610:Z$=Z$+"W"
820 IFA=88THENGOSUB3710:Z$=Z$+"X"
830 IFA=89THENGOSUB3890:Z$=Z$+"Y"
840 IFA=90THENGOSUB3800:Z$=Z$+"Z"
850 GOTO530
860 REM LETTER A
870 PRINT@C,11;" *** *"
880 PRINT@C,12;" *  **"
890 PRINT@C,13;"*   *"
900 PRINT@C,14;"*   *"
910 PRINT@C,15;"*   *"
920 PRINT@C,16;"*  ***"
930 PRINT@C,17;" *** *"
940 LETC=C+8
950 RETURN
960 PRINT@C,6;"*"
970 PRINT@C,7;"*"
980 PRINT@C,8;"*"
990 PRINT@C,9;"*"
1000 PRINT@C,10;"*"
1010 PRINT@C,11;"* ***"
1020 PRINT@C,12;"**  *"
1030 PRINT@C,13;"*   *"
1040 PRINT@C,14;"*   *"
1050 PRINT@C,15;"*   *"
1060 PRINT@C,16;"**  *"
1070 PRINT@C,17;"* ***"
1080 LETC=C+8
1090 RETURN
1100 REM LETTER C
1110 PRINT@C,11;" ***"
1120 PRINT@C,12;" *  *"
1130 PRINT@C,13;"**"
1140 PRINT@C,14;"**"
1150 PRINT@C,15;"**"
1160 PRINT@C,16;" *  *"
1170 PRINT@C,17;" ***"
1180 LETC=C+7
1190 RETURN
1200 REM LETTER D
1210 PRINT@(C+6),6;"*"
1220 PRINT@(C+6),7;"*"
1230 PRINT@(C+6),8;"*"
1240 PRINT@(C+6),9;"*"
1250 PRINT@(C+6),10;"*"
1260 PRINT@C,11;" *** *"
1270 PRINT@C,12;" *  **"
1280 PRINT@C,13;"*   *"
1290 PRINT@C,14;"*   *"
1300 PRINT@C,15;"*   *"
1310 PRINT@C,16;"*  **"
1320 PRINT@C,17;" *** *"
1330 LETC=C+8
1340 RETURN
1350 REM LETTER E
1360 PRINT@C,11;" ***"

```

```

1370 PRINT@C,12;" *   *"
1380 PRINT@C,13;"*     *"
1390 PRINT@C,14;"*****"
1400 PRINT@C,15;"*"
1410 PRINT@C,16;" *   *"
1420 PRINT@C,17;"   ***"
1430 LETC=C+8
1440 RETURN
1450 REM LETTER F
1460 PRINT@C,6;"   ***"
1470 PRINT@C,7;" *   *"
1480 PRINT@C,8;" *"
1490 PRINT@C,9;"*****"
1500 PRINT@C,10;" *"
1510 PRINT@C,11;" *"
1520 PRINT@C,12;" *"
1530 PRINT@C,13;" *"
1540 PRINT@C,14;" *"
1550 PRINT@C,15;" *"
1560 PRINT@C,16;" *"
1570 PRINT@C,17;" *"
1580 LETC=C+7
1590 RETURN
1600 REM LETTER G
1610 PRINT@C,11;"   *** *"
1620 PRINT@C,12;" *   ***"
1630 PRINT@C,13;"*     *"
1640 PRINT@C,14;"*     *"
1650 PRINT@C,15;"*     *"
1660 PRINT@C,16;" *   ***"
1670 PRINT@C,17;"   *** *"
1680 PRINT@C,18;"   *"
1690 PRINT@C,19;"   *"
1700 PRINT@C,20;"   *"
1710 PRINT@C,21;" *   *"
1720 PRINT@C,22;"   ***"
1730 LETC=C+8
1740 RETURN
1750 PRINT@C,6;"*"
1760 PRINT@C,7;"*"
1770 PRINT@C,8;"*"
1780 PRINT@C,9;"*"
1790 PRINT@C,10;"*"
1800 PRINT@C,11;"* ****"
1810 PRINT@C,12;"**   *"
1820 PRINT@C,13;"*     *"
1830 PRINT@C,14;"*     *"
1840 PRINT@C,15;"*     *"
1850 PRINT@C,16;"*     *"
1860 PRINT@C,17;"*     *"
1870 LETC=C+8
1880 RETURN
1890 REM LETTER I
1900 PRINT@C,9;" *"
1910 PRINT@C,11;" *"
1920 PRINT@C,12;" *"
1930 PRINT@C,13;" *"
1940 PRINT@C,14;" *"
1950 PRINT@C,15;" *"
1960 PRINT@C,16;" *"
1970 PRINT@C,17;" *"
1980 LETC=C+4
1990 RETURN

```

```

2000 REM LETTER J
2010 PRINT@C,9;"      * "
2020 PRINT@C,11;"     * "
2030 PRINT@C,12;"     * "
2040 PRINT@C,13;"     * "
2050 PRINT@C,14;"     * "
2060 PRINT@C,15;"     * "
2070 PRINT@C,16;"     * "
2080 PRINT@C,17;"     * "
2090 PRINT@C,18;"     * "
2100 PRINT@C,19;"     * "
2110 PRINT@C,20;"     * "
2120 PRINT@C,21;" *  * "
2130 PRINT@C,22;" *** "
2140 LETC=C+7
2150 RETURN
2160 REM LETTER K
2170 PRINT@C,6;" * "
2180 PRINT@C, 7;" * "
2190 PRINT@C, 8;" * "
2200 PRINT@C, 9;" * "
2210 PRINT@C,10;" *  * "
2220 PRINT@C,11;" *  * "
2230 PRINT@C,12;" *  * "
2240 PRINT@C,13;" *  * "
2250 PRINT@C,14;" *  * "
2260 PRINT@C,15;" *  * "
2270 PRINT@C,16;" *  * "
2280 PRINT@C,17;" *  * "
2290 LETC=C+8
2300 RETURN
2310 REM LETTER L
2320 PRINT@C,6;" * "
2330 PRINT@C, 7;" * "
2340 PRINT@C, 8;" * "
2350 PRINT@C, 9;" * "
2360 PRINT@C,10;" * "
2370 PRINT@C,11;" * "
2380 PRINT@C,12;" * "
2390 PRINT@C,13;" * "
2400 PRINT@C,14;" * "
2410 PRINT@C,15;" * "
2420 PRINT@C,16;" * "
2430 PRINT@C,17;" * "
2440 LETC=C+4
2450 RETURN
2460 REM LETTER M
2470 PRINT@C,11;" *  *  * "
2480 PRINT@C,12;" *  *  * "
2490 PRINT@C,13;" *  *  * "
2500 PRINT@C,14;" *  *  * "
2510 PRINT@C,15;" *  *  * "
2520 PRINT@C,16;" *  *  * "
2530 PRINT@C,17;" *  *  * "
2540 LETC=C+10
2550 RETURN
2560 REM LETTER N
2570 PRINT@C,11;" *  *  * "
2580 PRINT@C,12;" *  *  * "
2590 PRINT@C,13;" *  *  * "
2600 PRINT@C,14;" *  *  * "
2610 PRINT@C,15;" *  *  * "
2620 PRINT@C,16;" *  *  * "

```



```

2630 PRINT@C,17;"*      *"
2640 LETC=C+8
2650 RETURN
2660 REM LETTER O
2670 PRINT@C,11;"   ***"
2680 PRINT@C,12;" *    *"
2690 PRINT@C,13;"*      *"
2700 PRINT@C,14;"*      *"
2710 PRINT@C,15;"*      *"
2720 PRINT@C,16;" *    *"
2730 PRINT@C,17;"   ***"
2740 LETC=C+8
2750 RETURN
2760 REM LETTER P
2770 PRINT@C,11;"*   ***"
2780 PRINT@C,12;"***    *"
2790 PRINT@C,13;"*      *"
2800 PRINT@C,14;"*      *"
2810 PRINT@C,15;"*      *"
2820 PRINT@C,16;"***    *"
2830 PRINT@C,17;"*   ***"
2840 PRINT@C,18;"**"
2850 PRINT@C,19;"**"
2860 PRINT@C,20;"**"
2870 PRINT@C,21;"**"
2880 PRINT@C,22;"**"
2890 LETC=C+8
2900 RETURN
2910 REM LETTER Q
2920 PRINT@C,11;"   *** *"
2930 PRINT@C,12;" *    **"
2940 PRINT@C,13;"*      *"
2950 PRINT@C,14;"*      *"
2960 PRINT@C,15;"*      *"
2970 PRINT@C,16;" *    **"
2980 PRINT@C,17;"   *** *"
2990 PRINT@C,18;"      *"
3000 PRINT@C,19;"      *"
3010 PRINT@C,20;"      *"
3020 PRINT@C,21;"      *  *"
3030 PRINT@C,22;"      **"
3040 LETC=C+8
3050 RETURN
3060 REM LETTER R
3070 PRINT@C,11;"*   ***"
3080 PRINT@C,12;"***    *"
3090 PRINT@C,13;"*      "
3100 PRINT@C,14;"*      "
3110 PRINT@C,15;"*      "
3120 PRINT@C,16;"*      "
3130 PRINT@C,17;"*      "
3140 LETC=C+7
3150 RETURN
3160 REM LETTER S
3170 PRINT@C,11;"   ***"
3180 PRINT@C,12;"*      *"
3190 PRINT@C,13;"**"
3200 PRINT@C,14;"   ***"
3210 PRINT@C,15;"      *"
3220 PRINT@C,16;"*      *"
3230 PRINT@C,17;"   ***"
3240 LETC=C+6
3250 RETURN

```

```

3260 REM LETTER T
3270 PRINT@C,6;"*"
3280 PRINT@C,7;"*"
3290 PRINT@C,8;"*"
3300 PRINT@C-1,9;"*****"
3310 PRINT@C,10;"*"
3320 PRINT@C,11;"*"
3330 PRINT@C,12;"*"
3340 PRINT@C,13;"*"
3350 PRINT@C,14;"*"
3360 PRINT@C,15;"*"
3370 PRINT@C,16;"*   *"
3380 PRINT@C,17;"  ***"
3390 LETC=C+6
3400 RETURN
3410 REM LETTER U
3420 PRINT@C,11;"*   *"
3430 PRINT@C,12;"*   *"
3440 PRINT@C,13;"*   *"
3450 PRINT@C,14;"*   *"
3460 PRINT@C,15;"*   *"
3470 PRINT@C,16;" *   ***"
3480 PRINT@C,17;"  *** *"
3490 LETC=C+8
3500 RETURN
3510 REM LETTER V
3520 PRINT@C,11;"*   *"
3530 PRINT@C,12;"*   *"
3540 PRINT@C,13;" *   *"
3550 PRINT@C,14;" *   *"
3560 PRINT@C,15;" *  *"
3570 PRINT@C,16;" *  *"
3580 PRINT@C,17;"   *"
3590 LETC=C+8
3600 RETURN
3610 REM LETTER W
3620 PRINT@C,11;"*   *"
3630 PRINT@C,12;"*   *"
3640 PRINT@C,13;"*   *"
3650 PRINT@C,14;"*   *   *"
3660 PRINT@C,15;" * * * * *"
3670 PRINT@C,16;" * * * * *"
3680 PRINT@C,17;" *   *"
3690 LETC=C+10
3700 RETURN
3710 PRINT@C,11;"*   *"
3720 PRINT@C,12;" *   *"
3730 PRINT@C,13;" *  *"
3740 PRINT@C,14;"   *"
3750 PRINT@C,15;" *  *"
3760 PRINT@C,16;" *   *"
3770 PRINT@C,17;"*   *"
3780 LETC=C+8
3790 RETURN
3800 PRINT@C,11;"*****"
3810 PRINT@C,12;"   *"
3820 PRINT@C,13;"   *"
3830 PRINT@C,14;"   *"
3840 PRINT@C,15;"   *"
3850 PRINT@C,16;"   *"
3860 PRINT@C,17;"*****"
3870 LETC=C+8
3880 RETURN

```

```

3890 REM LETTER Y
3900 PRINT@C,11;"*      *"
3910 PRINT@C,12;"*      *"
3920 PRINT@C,13;"*      *"
3930 PRINT@C,14;"*      *"
3940 PRINT@C,15;"*      *"
3950 PRINT@C,16;" *    ***"
3960 PRINT@C,17;" ***  *"
3970 PRINT@C,18;"      *"
3980 PRINT@C,19;"      *"
3990 PRINT@C,20;"      *"
4000 PRINT@C,21;" *    *"
4010 PRINT@C,22;" ***"
4020 LETC=C+8
4030 RETURN
4040 REM SPRITES
4050 GOTO4760
4060 REM DOG
4070 SHAPE152,"00000000011F1D1F1F010101010101000000000000
      20408F8F8F8F808080800"
4080 MAG2
4090 SPRITE6 ,190,175,1,152
4100 RETURN
4110 REM BIRD
4120 SHAPE152,"000000000C161E3F6702020100000000000000102053
      EC514E40810E04040C000"
4130 MAG2
4140 SPRITE6 ,190,175,1,152
4150 RETURN
4160 REM YACHT
4170 SHAPE152,"000103070F1F3F7FFF00FF7F3F00000080004060607
      078787C80FFFEFC000000"
4180 MAG2
4190 SPRITE6 ,190,175,1,152
4200 RETURN
4210 REM CHAIR
4220 SHAPE152,"070704070404070F1F3F252121212100E0E020E0202
      0E0E0A020200000000000"
4230 MAG2
4240 SPRITE6 ,190,175,1,152
4250 RETURN
4260 REM HOUSE
4270 SHAPE152,"00003F7FFF80BCA4BC8080BDA5BD81FF0020FCFEFF0
      13D253D0101BDA5BD81FF"
4280 MAG2
4290 SPRITE6 ,190,175,1,152
4300 RETURN
4310 REM VAN
4320 SHAPE152,"000103070F1F10101F3F7FFFBFFFE00FCFCFCFCDC9
      CBCFCF8F8E0C080800000"
4330 MAG2
4340 SPRITE6 ,190,175,1,152
4350 RETURN
4360 REM BOAT
4370 SHAPE152,"0000000000000000000000003FF7F3F000000040000216
      60400D8D8FCFFFEFC0000"
4380 MAG2
4390 SPRITE6 ,190,175,1,152
4400 RETURN
4410 REM CAT
4420 SHAPE152,"080F0A0D0707070F0F1818310000000080808080008
      0C3E4F8F0D89808180000"
4430 MAG2

```



```

4440 SPRITE6 ,190,175,1,152
4450 RETURN
4460 REM TREE
4470 SHAPE152,"000564123B65150C150B0701010101010010A0A4585
      CA1C6B8C8C48280808080"
4480 MAG2
4490 SPRITE6 ,190,175,12,152
4500 RETURN
4510 REM CUP
4520 SHAPE152,"000000001020406070707070703010000000000C0201
      43AF2F4F8F0F0E0C00000"
4530 MAG2
4540 SPRITE6 ,190,175,6,152
4550 RETURN
4560 REM BALL
4570 SHAPE152,"01070F1F1F3F3F3F3F3F3F1F1F0F0701C0F0F8FCFCF
      EFEFEFEFEFEFCFCF8F0C0"
4580 MAG2
4590 SPRITE6 ,190,175,1,152
4600 RETURN
4610 REM BED
4620 SHAPE152,"0000000010202030708102143FEFC84840000C020101
      0F0F070D08000000000000"
4630 MAG2
4640 SPRITE6 ,190,175,4,152
4650 RETURN
4660 REM TANK
4670 SHAPE152,"000000001F1F1FFF80804F3F000000000000000080F
      E80F80E02E4F800000000"
4680 MAG2
4690 SPRITE6 ,190,175,12,152
4700 RETURN
4710 REM CLOCK
4720 SHAPE152,"03040812204840504048201208040307E01088A4828
      981F5010902248810F8FC"
4730 MAG2
4740 SPRITE6 ,190,175,1,152
4750 RETURN
4760 REM APPLE
4770 SHAPE152,"0000020101070F1E1E3F3F3F1F0F070370E0C00000C
      0E0707038F8D8F0E0C080"
4780 MAG2
4790 SPRITE6 ,190,175,12,152
4800 RETURN
4810 REM SUN
4820 SHAPE152,"11090500611307E70727CB111424080010242408C2E
      4F0F7F0F0E2C904A29090"
4830 MAG2
4840 SPRITE6 ,190,175,10,152
4850 RETURN
4860 REM CAR
4870 SHAPE152,"00000000000000030408087FFFFFFF3800000000000
      000F0908888FFFFFFF1C"
4880 MAG2
4890 SPRITE6 ,190,175,5,152
4900 RETURN
4910 REM FISH
4920 SHAPE152,"000000000F1F2F7F7F3F1F0F000000000000000008
      4ECFCFCCEC840000000000"
4930 MAG2
4940 SPRITE6 ,190,175,14,152
4950 RETURN
4951 REM BUS

```

```

4952 SHAPE152,"FF8888FFFF91919F9F9FFF3000000000FF8989FFFF1
      111FFFFFFFFF0C00000000"
4953 MAG2
4954 SPRITE6 ,190,175,6,152
4956 RETURN
4960 REM LIGHT
4970 SHAPE152,"0000070F1F1F1F1F0F0F0F0F070303030000C0E0F0F
      0F0F0E0E0E0E0C0808080"
4980 MAG2
4990 SPRITE6 ,190,175,14,152
5000 RETURN
5010 REM CRANE
5020 SHAPE152,"00002030382C2623210000FFFFFFF303000000000000
      0001F91D17FFFFFFFFF0C0C"
5030 MAG2
5040 SPRITE6 ,190,175,11,152
5050 RETURN
5060 REM BOOT
5070 SHAPE152,"0000000000000000001F3F3F3F1F0F0000000038447C7
      C7CF8F8FCFCFCFCFC9C0000"
5080 MAG2
5090 SPRITE6 ,190,175,6,152
5100 RETURN
5110 REM BACK
5120 SHAPE0,"FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
      FFFFFFFFFFFFFFFFFFFFFF"
5130 MAG2
5140 SPRITE16,182,183,15,0
5150 MAG2
5160 SPRITE20,198,183,15,0
5170 MAG2
5180 SPRITE24,182,167,15,0
5190 MAG2
5200 SPRITE28,198,167,15,0
5210 RETURN
5220 IFZ=1THENGOTO5260
5230 IFZ=2THENGOTO5300
5240 IFZ=3THENGOTO5340
5250 IFZ=4THENGOTO5400
5260 FORI=190TO1STEP-1
5270 SPRITE6,I,175,1,152
5280 NEXTI
5290 RETURN
5300 FORI=1TO200
5310 MAG3
5320 IFI=200THENRETURN
5330 NEXTI
5340 I=190
5350 J=175
5360 SPRITE6,I,J,1,152
5370 I=I-2
5380 J=J-2
5390 IFJ>8THENGOTO5360ELSEReturn
5400 FORI=1TO200
5410 Y=RND(10)
5420 MAG3
5430 IFY=>4THENH=10
5440 IFY<4THENH=14
5450 SPRITE6,190,175,H,152
5460 IFI=200THENRETURN
5470 NEXTI

```



## USING MOS E INSTRUCTION

The E instruction is an essential tool for debugging machine code. It is therefore most unfortunate that Tatung make no attempt to explain its use. The instruction sets a breakpoint at the address specified, so that the program runs until it reaches the address and then stops to allow examination of memory and registers. The problem is how to get into MOS to give the instruction once the program is running.

With simple programs, the sequence LOAD, MOS, G will work perfectly well. However if the program requires a command tail (e.g. a second file name), this is ignored by LOAD. I have overcome this in the past by changing the first byte of the program to OFFH, which has almost the same effect as E. After calling in the normal way the byte is changed back and processing resumes with another E instruction. I have recently discovered that this can cause obscure problems with the next program because the memory store for the E instruction is only reinitialised on reset.

I now have a file TAIL.COM consisting of the single machine code instruction 0C7H ( RST 0 ). On calling TAIL 1:data or whatever arguments are needed, DOS writes the tail (at 080H) and returns. Then the sequence LOAD progname, MOS, G0100 bbbb starts the program, sets the breakpoint bbbb and allows the command tail to be picked up when required. G overwrites any earlier breakpoints left in store.

## MOS INSTRUCTIONS E & G

Further experience with the original version of TAIL has shown that although it works in the way intended, there is a second source of chaos. This arises because the default stack is only initialised by a normal program call and not by LOAD. The result is that on entering MOS, any stack left lying around by a program (usually as a result of breaking out of a run during testing) is still active. This would not matter in general since programs should set up their own stack, but unfortunately MOS appears to use a RET when transferring control to programs, and the Program Counter is pushed onto the stack, dumping 0100H in the middle of your code when you are in the process of tracking down a programming error.

I have revised TAIL.COM to initialise

(1) the MOS stack register store and

(2) the E-instruction stores,

in addition to its effect in setting up the command tail. The program should be run routinely before using MOS for entering a program, even when there is no command tail. It is now possible to write a breakpoint (OFFH) in a .COM file and then call it directly with its normal command tail, if TAIL is called first.

There is still an unsolved problem: in testing interrupt routines from MOS. I have found that there is a write protect on the MOS stack which leads to the loss of the return address and disabling of interrupts. Further contributions would be welcome.

I can only hope that there are no more booby traps still to be found. This emphasises again the need for Tatung to explain how MOS is supposed to be used.



TAIL.COM

0100	31 1A EB	LD	SP,0EB28H
0103	ED 73 65 FB	LD	(0FB65H),SP
0107	21 00 00	LD	HL,0
010A	22 5D FB	LD	(0FB5DH),HL
010D	3E FF	LD	A,0FFH
010F	32 5C FB	LD	(0FB5CH),A
0112	C7	RST	0

C. P. Wallis  
Mar. 1986

~~~~~

Herewith follow up article on my experiences using a computer in my business.

### NINE MONTHS ON WITH MY EINSTEIN

Last year I talked about the "Micro Simplex Accounting System" it has turned out to be excellent.

At the same time I purchased a programme called "THE CRACKER" which is a Spreadsheet programme, with a view to producing a "CASH FLOW" forecast. This has turned out to be a most valuable purchase, not only was I able to do the "CASH FLOW" forecast for the year, but on entry of the monthly figures the forecasted yearly figures are updated immediately.

I have found that "THE CRACKER" can be used for many management functions, namely:-

1. Comparison of SALES monthly.
2. Comparison of WAGES monthly.
3. Comparison of GROSS PROFITS monthly (estimated cash & percentage).
4. Checking on turnover with particular Suppliers.
5. Preparing ORDERS in advance of infrequent Reps. calls.
6. Raising and Printing orders for mailing, (This is considerably cheaper than printed order forms).
7. Keeping Diary dates of Reps. calls & Staff holidays etc.
8. Listing items of UNUSUAL STOCK with correct Suppliers description.
9. Writing Letters.
10. Key Blank Locations.
11. Comparison of prices from different Suppliers.
12. Listing Retail Prices from Nett Priced Invoices (this saves time when the buying price has'nt changed)
13. Printing Mail Labels.
14. "THE CRACKER" will also print "GRAPHS", very useful on Sales comparison etc.
15. End of month statements.

All the above I actually do regularly, at the moment I am trying to put staff wages into the computer using "THE CRACKER" to save all the adding up at the end of the TAX year, already it has pointed out to me an error, this was where Pensions deductions are made before tax is deducted, in this case the Employee was on Jury Service and didn't receive wages for two weeks, but the pensions was still payable and therefore still Tax deductible, the tax was not adjusted when it should have been.

It is of course sometimes a bit tedious building up the type of format that you require, but once you have done it you change it around to input the figures or change the addresses whenever you fancy.

"THE CRACKER" does all the calculations for you.

CONCLUSIONS: MY EINSTEIN AND THE TWO PROGRAMMES I USE ARE THE BEST THING SINCE SLICED BREAD!.

DENNIS BARNETT F.L.M.A.

~~~~~

### CRACKER GRAPHICS

It's a pity that the CRACKER's printed manual doesn't cover the impressive graphics capabilities of this cost-effective package. To supplement the scrappy and misleading notes which came with the software, I've put together a few notes which may save other users' time.

To start with, CRACKER uses a lot of disc space and if you have two disc drives it certainly pays to prefix all your filenames with 1: so that apart from the security and backup copies automatically provided, you don't clutter up the disc on drive 0, which need only have CRACKER.COM and CRACKER.OVLY on it. CRACKER.HLP is not essential for running the package.

The plotting commands must all be put in column A, so the spreadsheet proper must start in B. The essential functions are as follows, and with the exception of TYPEPLOT their arguments must be co-ordinates (crd) of cells containing the data for the plot. The requirements should be clear from the list below and the sample bar graph (see back page).

MAINTITLE(crd)	
SUBTITLE(crd)	
XLABEL(crd...crd)	Titles for the keys to the various plot symbols.
TIMELABEL(crd...crd)	The labels for the points on the x-axis (usually years or months, but can be any units).
YTITLE(crd)	
XTITLE(crd)	
YVALUE(n,crd...crd)	You need one of these commands for each set of points you want to plot. For example, my first two commands were: YVALUE(1,C6...H6) and YVALUE(2,C7...H7).
TYPEPLOT(n or crd)	Values of 1 to 6 inclusive control the type of plot produced.

Some of the limits given for the maximum lengths and numbers of labels are very misleading. My specimen graph uses the quoted maximum lengths wherever they are correct. However, if you use too many XLABELS and TIMELABELS the bars of the chart become so narrow that the hatching patterns can't be distinguished. The patterns repeat after the first six anyway. A reasonable display is m XLABELS and n TIMELABELS, where  $m \cdot n = 36$ , which you can get for example from six XLABELS and six TIMELABELS. Using six TIMELABELS the labels run into each other at 6 characters length, so have a care with abbreviations for months. Four digits and a decimal point should be OK for numerical annotations of the x-axis.

If you widen column A and use the eXchange command, the plot instructions will be displayed in full, which helps checking.

~~~~~



Chuckie Egg Tips

By Katie Molyneux aged 12  
(also sent in by Tracy Crouch aged 10 )

Of all the games we have my favourites are Chuckie Egg, Hunch, Phrogger, Mummy, Flics, Baker, Kung Fu, Shark Hunter, and I quite like Galaxoids.

On all games it is rather annoying when you get killed off too early to get to play the higher levels.

I have found a way to make my favourite computer game, Chuckie Egg, go to the higher levels. Simply change the keys used, by pressing 'R, and changing them to Q=up : Z =down : C=left: B= right , leaving JUMP as space bar then, by pressing all these keys at once, when the game is starting (i.e. moving chickens lower level) you will then find you have jumped a level,

This can be repeated time and time again until you' get to the level you require.

The Advantage: You are able to practice the higher levels without having to go through all the levels every time you get killed off.

A similar trick would be very handy on Maxima, My Dad can't get past level 11, if anybody knows a way please tell him so as I can get back to playing Chucky.

~~~~~

OPEN QUESTION TO SOFTWARE PRODUCERS

When writing score type games why not include into the package the facility to dump the final score to the printer. This can then be used to ratify any wishful boasting or act as a certificate of merit. Lets face it on some games you may spend hours only to switch off and loose all .

Martin Page

~~~~~

How do you like the printer ?.

Here are the details:

Dot Matrix with 9 pin head, NLQ in all pitches, ( Standard, Italic, Compressed ), Bi-directional with logic seeking. Graphics mode, 32 international characters, 1k buffer and is Epson RX-80 compatible.

With luck and a lot more time I will try to include some italics in the next Newsletter.

Keith.

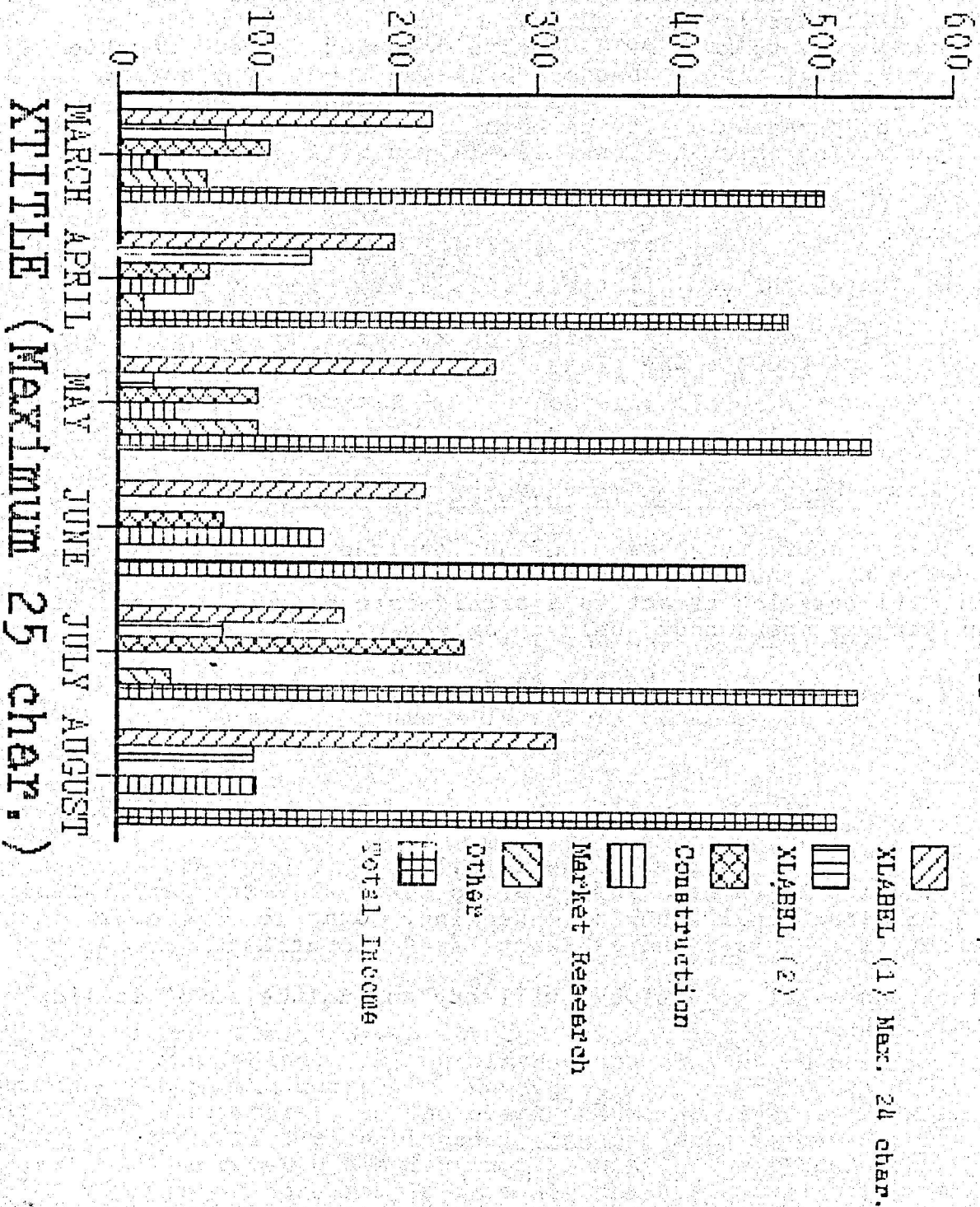
~~~~~



# MAINTITLE (Max. 24 char.)

SUBTITLE (Maximum 18 characters)

YTITLE (Maximum 25 char.)



XTITLE (Maximum 25 char.)