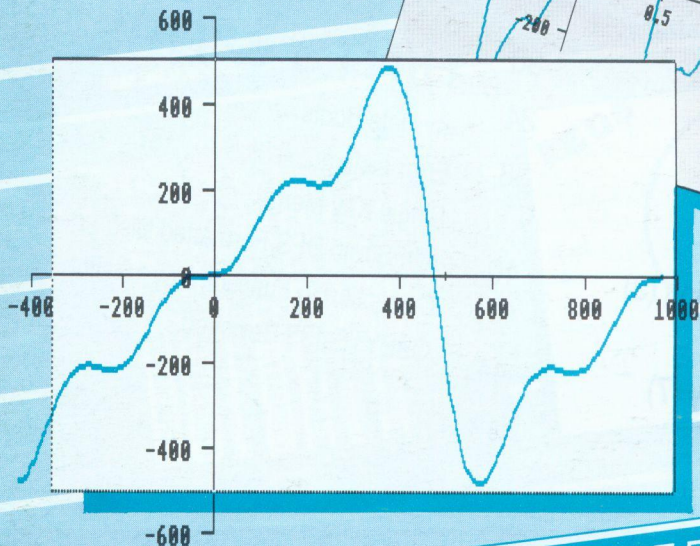
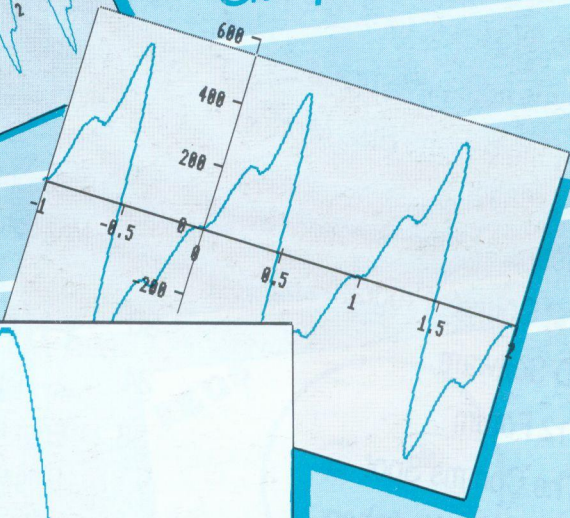
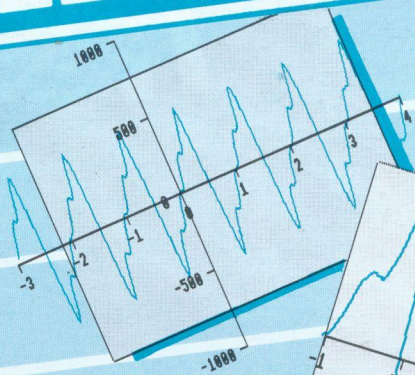


Vol.10 No.7 December 1991

BEEBUG

FOR THE
BBC MICRO &
MASTER SERIES

Self Numbering
Graph Program



- PD SOFTWARE
- OP-ART PATTERNS
- A DISC ORGANISER FOR ADFS
- PITFALL PETE

FEATURES

- Op-Art Patterns
- Self Numbering Graph Program
- A Disc Organiser for ADFS (1)
- Patterns from a Keyboard (2)
- The Future of the BBC Micro in Education
- Teletext and the Function Keys
- First Course: Program Logic (2)
- PD Software
- 512 Forum
- The Comms Spot
- BEEBUG Workshop:
Simulation Modelling (3)
- BEEBUG Function/
Procedure Library (7)
- Pitfall Pete

REGULAR ITEMS

- 6 Editor's Jottings 4
- 8 News 5
- Points Arising 48
- 13 RISC User 56
- 18 Postbag 57
- Hints and Tips 59
- 21 Personal Ads 60
- 24 Subscriptions & Back Issues 62
- Magazine Disc 63

HINTS & TIPS

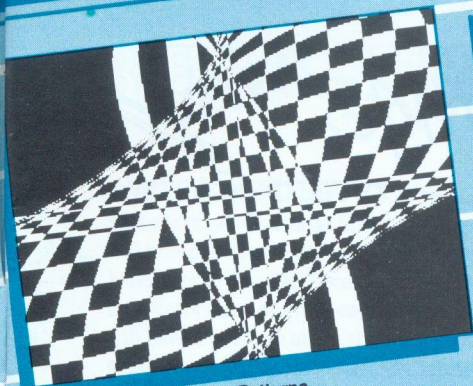
- 30 Invisible Boots
- 34 Error Listing
- 38 Single Key Merge
- 40 ADFS and DFS Formatted Disc
- Correction to Function Key 10 Hint
- 44 OSBYTE 135 Revisited
- 49

PROGRAM INFORMATION

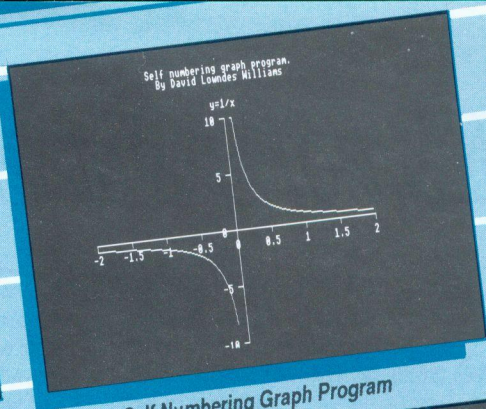
All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly as printed, and checked carefully. When entering a listing, pay special attention to the

difference between the digit one and a lower case l (L). Also note that the vertical bar character (Shift \) is reproduced in listings as |.

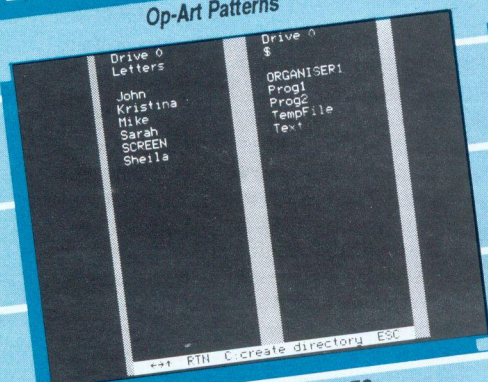
All programs in BEEBUG magazine will run on any BBC micro with Basic II or later, unless otherwise indicated. Members with Basic I are referred to the article on page 44 of BEEBUG Vol.7 No.2 (reprints)



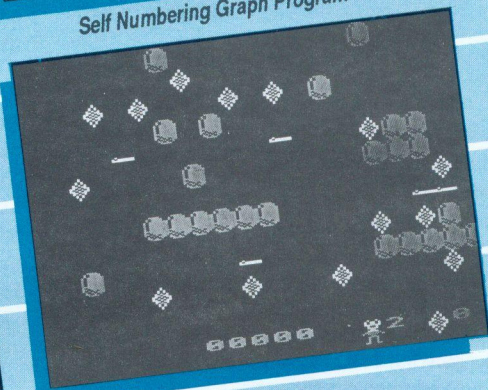
Op-Art Patterns



Self Numbering Graph Program



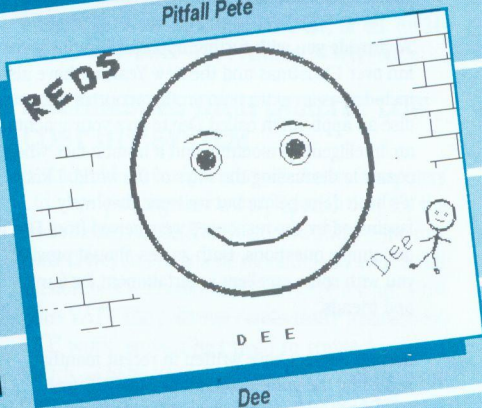
Disc Organiser for ADFS



Pitfall Pete



Patterns from a Keyboard



Dee

available on receipt of an A5 SAE), and are strongly advised to upgrade to Basic II. Any second processor fitted to the computer should be turned off before the programs are run.

Where a program requires a certain configuration, this is indicated by symbols at the beginning of the article (as shown opposite). Any other requirements are referred to explicitly in the text of the article.

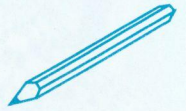


Program needs at least one bank of sideways RAM.



Program is for Master 128 and Compact only.

Editor's Jottings



This issue will be the last one to be mailed out before Christmas, and the following issue will be that for January/February 1992 which we expect to mail out mid January.

Because this is the Christmas issue we have included in the magazine one of our all-time favourite games, *Pitfall Pete* (enhanced from when it was first published in BEEBUG), and the magazine disc for this issue not only contains the complete game but a set of additional screens to provide added entertainment.

To provide you with further opportunities for some fun over Christmas and the New Year, we have also included as an extra item on this month's magazine disc an application called *Dee* to give your machine an 'intelligent' personality and a human face when it comes to discussing the ways of the world. I know it's been done before but we were absolutely fascinated by the responses we received from *Dee* to our simple questions. Both games should provide you with some excellent entertainment for family and friends.

Several readers have written in recent months regarding the games content of BEEBUG (see Postbag in this issue). I have already stated my position on this issue, but let me just reassure everybody that it is not our intention to turn BEEBUG into a games oriented magazine, far from it. And from the next issue we will return to our usual slightly more serious approach to the world of BBC micros.

In fact this issue has some excellent items in it anyway, with David Lowndes Williams' self-numbering graph plotting routines, and Peter Miles'

ADFS disc organiser, plus a new instalment of our Function/Procedure Library to name but a few - in fact, something for everybody.

We also have some excellent programs already lined up for future issues which we believe will continue to show just how capable the BBC micro and Master series of computers still is for many applications, despite the undoubted attractions of other more recent systems. Many readers maintain that a BBC micro continues to meet all their needs quite adequately, and I for one have long used a Master Compact for all my own personal word processing and financial accounting.

I see no real reason to change my habits, but if, because of my work with both RISC User and BEEBUG, I find an Archimedes does become the only machine I use at home, then I will more than likely just run some of my existing programs on that system. For those without such needs a BBC micro will continue to do much useful work for years to come. The only disappointing feature is that most software houses now devote all their attention to the Archimedes market. However, there is still much excellent software to be had if you are prepared to look for it, and the Personal Ads in BEEBUG can prove a highly fruitful source for otherwise unobtainable items. They are also a highly effective way of selling any computer software or hardware which you no longer need. Public Domain software is also on the upsurge for the BBC micro - see the new column in this issue.

Well, that is enough from your editor for now. Best wishes for 1992.

M.W.

FORTHCOMING SHOWS

This year's pre-Christmas *Computer Shopper Show* will take place at the Wembley Exhibition Centre, London from 5th - 8th December 1991. The show organisers have recently confirmed that Acorn will have a presence at the show, so it could be a good opportunity to see all the latest Acorn products.

In the new year, a major event for educational users will be *BETT 1992* from 22nd - 25th January 1992. *BETT 1992* will feature over 200 exhibitors, free subject based seminars, a Software Centre, an Acorn Village, a Publishing Village, and a Special Needs Area. RISC Developments will be there too, demonstrating all our latest software and hardware, and featuring our two magazines, *BEEBUG* and *RISC User*.

In addition, *The Education Show* will be held at the NEC, Birmingham from 5th - 7th March 1992.

For more information on the *Computer Shopper Show* contact Blenheim Pel on 081-868 4466, and for *BETT 1992* and *The Education Show* the organisers are EMAP who can be contacted on 071-404 4844.

NEW GAMES RELEASE FROM SUPERIOR

Indefatigable Superior has released a new games collection for the BBC micro and Master series. *Play it again Sam 16* features four complete games in a compilation comprising *Hostages*, *Vertigo*, *Perplexity*, and *Pipemania*. *Hostages* offers commando action and strategic planning, while *Perplexity* is a three dimensional strategy game for all Repton and Pacman fans. *Vertigo* is a completely new release with five levels of fun and frustration as you try to balance your way round fifty weird and wonderful structures. Last but not least, *Pipemania* is the highly acclaimed game from Empire Software.

Prices range from £12.95 for cassette, £14.95 for 5.25" disc, and £19.95 for the Master Compact disc version. For more information contact Superior Software at P.O.Box 6, Brigg, South Humberside DN20 9NH, tel. (0652) 658585, or your local dealer.

WAKEFIELD USER GROUP CHANGES NAME

The well established (since April 1983) *Wakefield BBC Micro User Group* is changing its name to *Wakefield Acorn Computer User Group*. From January 1992 meetings will be held on the first Thursday of each month at a new venue, the West Yorkshire Sports and Social Club, Sandal Hall Close, Sandal Magna, Wakefield. The group will continue to support all the Acorn/BBC range. For more information about the group contact them at 1 Wavell Garth, Sandal Magna, Wakefield, West Yorkshire WF2 6JP, or tel. (0924) 255515 or (0924) 379778 evenings or weekends.

ARCHIMEDES HIGH QUALITY LOW COST DTP SYSTEM

Beebug Ltd., in conjunction with Acorn Computers Ltd., has put together a very special deal for the Archimedes A3000. This features a 2Mbyte internal hard drive, Acorn colour monitor, and with a full version of the *Ovation DTP* package for just £999 (plus VAT). The software comes ready installed for DTP work, so you just need to connect up and switch on to be ready to go. *DTP* is one of the most popular applications for any Archimedes, and *Ovation* includes full word processing capability as well. There are also special offer prices on this system for educational users.

Beebug Ltd. are at 117 Hatfield Road, St Albans, Herts AL1 4JS or tel. (0727) 40303.

In the November News we quoted the number for SCET incorrectly. The number should be 041-334 9314.

Op-Art Patterns



Derek Greenacre explores the world of Op-art with the aid of his Master.

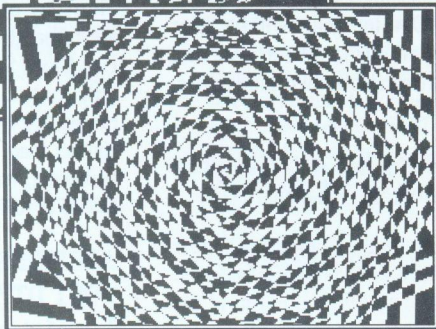
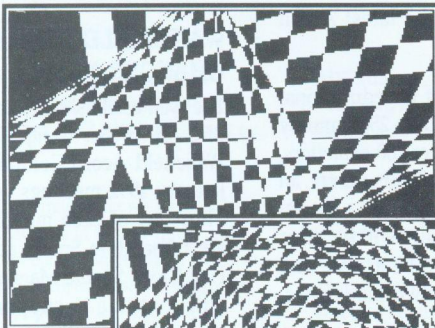
We do not see with our eyes but rather with our eyes and brain acting in unison. What we see is conveyed along our optic nerve to our brain. Here the brain processes what it sees by inverting the upside down image from the lens and

analysing the data to try and make sense of it. It is this analytical process in which the accurate seeing of reality is seriously flawed. We are all familiar with how our eye can be tricked

with optical illusions, yet we seem less aware of the role that the eye/brain combination plays in art. Consider any movie film; we know that in reality the eye is fooled into thinking that the rapid succession of static frames is a continuous moving image. Similarly, in a painting, the artist fools our eye with perspective to make us believe we are viewing a three dimensional scene when we know for a fact that the surface of the canvas is flat.

Op-Art takes this fooling of the eye to its logical conclusion by deliberately making pictures which confuse the image reaching the brain. This is achieved by drawing a multitude of very small similar shapes. Because the shapes are similar, the brain is tricked into thinking that they are identical and that the surface they are portrayed on is curved. The brain then re-processes the information as the data is updated, and this causes the dazzling effects associated with Op-art.

The computer is an excellent tool for generating these Op-art patterns as it easily draws a multitude of small accurate shapes, and the added benefit of movement greatly increases the dazzle effect.



THE PROGRAM

This exploits the colour inversion of the PLOT statements. On the Master series this proves ideal for fragmenting larger shapes into smaller component

pieces. The sine and cosine functions produce a rotational design expanding from the centre, and the program cycles through various PLOT statements. The colour can be changed by altering line 150,

and experimenting with different function values in the PROCtrig routine easily produces wildly different patterns. I have not been too concerned about the formal structure of the program concentrating instead on the visual effects produced.

The program runs for about eleven minutes, pausing briefly as it completes each picture. There are the occasional periods when little changes on the screen. However, in order to keep the program short, these have not been edited out and they do not detract from the overall pace of the images. Considering the relatively few lines in the program it produces a wide variety of dazzling Op-art shapes.

RECOMMENDED READING

Introducing Op-art by John Lancaster (published by Batsford).

Art and Visual Perception by Rudolph Arnheim (published by Faber).

Optical Art by Rene Parola (published by Reinhold).

Geometrical Patterns by Richard Slade (published by Faber).

```

10 REM Program OpArt
20 REM Version B1.0
30 REM Author Derek Greenacre
40 REM BEEBUG December 1991
50 REM Program subject to copyright
60 REM Master 128 or Master Compact
70 :
100 MODE 4
110 ON ERROR:MODE7:PROCError:END
120 step%=10
130 VDU 29,640;512::MOVE0,0
140 VDU23,1,0;0;0;0;
150 VDU 19,1,0,0,0,0,0,19,0,6,0,0,0
160 :
170 PROC1:PROC1
180 PROC2
190 J%=INKEY(400):CLS
200 PROC3
210 step%=5
220 J%=INKEY(400):CLS
230 PROC4
240 J%=INKEY(400):CLS
250 step%=15
260 PROC5
270 J%=INKEY(400):CLS
280 step%=10
290 PROC6
300 VDU7:END
310 :
1000 DEF PROC1
1010 FOR H%=1 TO 800 STEP step%
1020 PROCtrig
1030 MOVE-X%, -Y%:PLOT206,X%,Y%:MOVE 0,0
1040 NEXT
1050 ENDPROC
1060 :
1070 DEF PROC2
1080 FOR H%=1 TO 800 STEP step%
1090 PROCtrig
1100 MOVE-X%, -Y%:PLOT158,X%,Y%:MOVE 0,0
1110 NEXT
1120 ENDPROC

```

```

1130 :
1140 DEF PROC3
1150 FOR H%=1 TO 800 STEP step%
1160 PROCtrig
1170 MOVE-X%, -Y%:PLOT102,X%,Y%:MOVE 0,0
1180 NEXT
1190 ENDPROC
1200 :
1210 DEF PROC4
1220 FOR H%=1 TO 800 STEP step%
1230 PROCtrig
1240 MOVE-X%+H%, -Y%+H%:PLOT 102,X%-H%,Y
%-H%
1250 NEXT
1260 ENDPROC
1270 :
1280 DEF PROC5
1290 FOR H%=1 TO 800 STEP step%
1300 PROCtrig
1310 MOVE-X%+H%, -Y%+H%:PLOT 102,X%-H%,Y
%-H%:MOVE0,0:PLOT 158, -X%+H%, -Y%+H%
1320 NEXT
1330 ENDPROC
1340 :
1350 DEF PROC6
1360 D%=86:flag%=0
1370 REPEAT
1380 S%=4
1390 REPEAT
1400 MOVE0,0:MOVE0,0
1410 FOR H%=0 TO 900 STEP S%
1420 PLOT D%,COSH%*H%,SINH%*H%
1430 IF flag%=1 PLOT 70, -COSH%*H%, -SINH
%*H%
1440 NEXT:J%=INKEY(400):CLS:S%=S%+1
1450 UNTIL S%=8:flag%=flag%+1
1460 IF flag%=2 D%=158:S%=5
1470 UNTIL flag%>2
1480 PRINTTAB(10,12);"That's All Folks"
1490 ENDPROC
1500 :
1510 DEF PROCtrig
1520 T=H%/100
1530 X%=SIN(T)*H%
1540 Y%=COS(T)*H%
1550 ENDPROC
1560 :
1570 DEF PROCError
1580 REPORT:VDU7
1590 PRINT TAB(5,15);CHR$(130);"Error n
umber ";ERR;" at line ";ERL
1600 ENDPROC

```

Self Numbering Graph Program

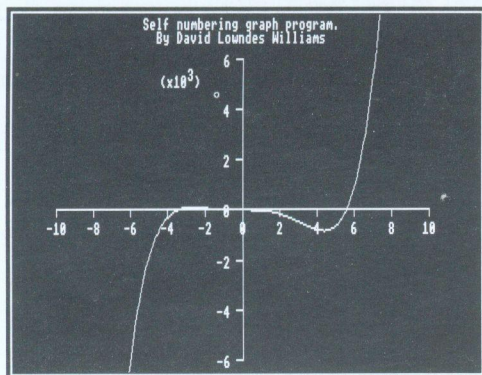
by David Lowndes Williams

INTRODUCTION

Many graph programs available for the BBC just label the minimum and maximum value on each axis and put a few regularly spaced dashes along the axis to produce an impression of scale. The program presented here demonstrates the technique of labelling a graph with "nice numbers". This results in a more professional looking graph.

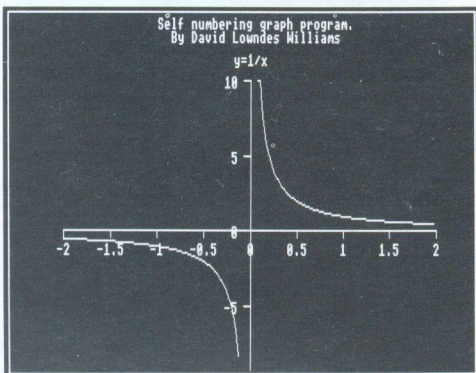
WHAT IT DOES

Given the minimum and maximum values for each axis, the program will number the axes, and scale the graph accordingly. It will cope with very large and very small numbers because it can make use of engineering notation (see the article in BEEBUG Vol.10 No.5). The numbers it chooses for the scale are "nice" and resemble the numbers that a sensible human would pick.



The program draws a periodic function on the screen after numbering the axes. A box appears on the screen which you may edit using the cursor keys. <Shift> increases the speed at which the editing occurs. <Z> causes the bottom left corner

to be moved, and <X> the top right corner. <Return> instructs the computer to draw a new graph. The edges of the box are the limits for the new graph. <Escape-S> saves the screen under the filename "screen".



ENTERING THE PROGRAM

BBC micro disc users will have to lower the value of PAGE a little before the program can be run. After saving the program, under the filename "GRAPH", run the program by issuing the following:

```
PAGE=&1500  
CHAIN"GRAPH"
```

FIDDLING WITH THE FUNCTION

The graph function to be used is defined from line 1000 onwards, and may be changed as required. The limits of the axes are MINX, MAXX, MINY, and MAXY. They are set initially in lines 170 and 180. These are changed by PROCedit, which performs box editing. The length of one complete oscillation is set by the variable *Period* in line 140. This variable would, of course, become redundant if FNfunction contained no

Self Numbering Graph Program

reference to it, or to the variable `const` which is derived from it.

In the program as it stands, `FNfunction(x)` is the truncated Fourier series of a saw-tooth waveform with period *Period*.

HOW IT WORKS

To produce a "nicely" numbered graph, we ensure that the difference between successive labels is always a "nice number". We define "nice numbers" as 1,2 and 5, and these numbers multiplied by 0.1, 10, 100, 1000 etc. The function `FNniceNumber(num)` produces a list of nice numbers (see Fig. 1), and returns the one which lies closest to the argument, *num*. Given the minimum value and the maximum value (for one axis), together with the desired number of labels, `PROclimits` returns the highest (H) and lowest (L) values of the labelling as well as the step size (S), which is determined by `FNniceNumber`. The program performs "loose-labelling" which means it draws the nearest "nice" number labels outside the range given by min and max.

`PROclimits` performs the "nice labelling". The rest of the program supports this and allows any function, which can be specified by `FNfunction`, to be displayed.

HOW THE GRAPH IS DISPLAYED

If *y* is the value of `FNfunction(x)` where *x* is the independent variable, then the co-ordinates of a single point defined by this function are (*x*,*y*). This point needs to be mapped onto a point on the screen. This is done by two functions, `FNscreenx(x)` and `FNscreeny(y)`,

i.e. (*sx*,*sy*)=(`FNscreenx(x)`,`FNscreeny(y)`). where *sx* and *sy* are the co-ordinates on the screen.

The reverse process of converting from screen co-ordinates to co-ordinates relating to the function, is performed within `PROCedit`, at lines 2750 to 2800. This is to enable the region bounded by the box to be used to calculate the limits of the new graph. This process also occurs, to a lesser extent, at line 1280 by `FNpiece`. `FNpiece` determines the step size, between successive points on the curve, in units appropriate to `FNfunction`. It does this from the pixel resolution of the screen (mode 0 has 2 screen units per pixel).

The program is very useful for drawing mathematical functions. It could easily be converted to plot graphs from tables of values. The lines that actually plot the

curve are 1310 to 1350, so replace these with your own routines if you wish.

TIPS FOR PLOTTING GRAPHS

Many mathematical expressions are only valid for certain ranges of values. If care is not taken, errors like: "division by zero", "-ve root" and "log range" can occur. Also, "accuracy lost", "exp range" and "Too big" will prevent values being returned by `FNfunction(x)`. To ensure that we don't induce such errors we must not instruct the computer to do an impossible calculation. We must therefore first test if, for example, the argument of a `SQR()` is -ve. The variable `undefined%` is set to `TRUE` when the function is undefined. `FNfunction(x)` must still, however, be set to something. When `undefined%=TRUE` the point on

0.1	10	1000
0.2	20	2000
0.5	50	5000
1	100	
2	200	
5	500	

Figure 1. List of "Nice Numbers"

Self Numbering Graph Program

the graph is MOVEed to, if FALSE, the computer issues a DRAW command. Hence, you must carefully pick the value that you set **FNfunction(x)** to when it is actually undefined.

You may wish to title the graphs, this may be done using **PROctext**. For example:

```
1085 PROctext("Saw-Tooth",640,900,"C")
```

The arguments are: the string to be printed, the screen co-ordinates, followed by "L", "C" or "R". These tell the procedure if the co-ordinate refers to a point at the left, the centre or right of the string. The procedure then locates the text in the desired position. This procedure can be used in your own programs, but it must be altered slightly to work correctly in graphics modes other than mode 0.

Reference: *"Nice numbers for graph labels"* Paul S. Heckbert. P61-63 Graphics Gems, edited by Andrew S. Glassner. Academic Press, 1990.

```
10 REM Program Graph
20 REM Version B3.0
30 REM Author David Lowndes Williams
40 REM BEEBUG December 1991
50 REM Program subject to copyright
60 :
100 ON ERROR GOTO270
110 MODE0
120 VDU19,0,4,0,0,0
130 PROCdimension
140 Period=1E-3
150 const=2*PI/Period
160 :
170 MINX=-Period:MAXX=2*Period
180 MINY=-550:MAXY=550
190 REPEAT
200 CLS
210 PROCgraph(MINX,MAXX,MINY,MAXY)
220 PROCeditLimits(MINX,MAXX,MINY,MAXY)
```

```
230 UNTIL FALSE
240 :
250 END
260 :
270 IF ERR=17 AND INKEY=82 THEN *SAVE"
screen"3000 8000
280 IF ERR=17 AND INKEY=1 OSCLI"FX4,0"
:END
290 REPORT:PRINT" at line ";ERL
300 *FX4,0
310 END
320 :
1000 DEF FNfunction(x)
1010 LOCALd
1020 undefined%=FALSE
1030 IF undefined%=TRUE d=0 ELSE d=318*
SIN(x*const)-159*SIN(2*x*const)+106*SIN(
3*x*const)-80*SIN(4*x*const)
1040 =d
1050 :
1060 DEF PROCgraph(dminx,dmaxx,dminy,dm
axy)
1070 PROctext("Self numbering graph pro
gram.",640,1015,"C")
1080 PROctext("By David Lowndes William
s",640,983,"C")
1090 IF ABS(dmaxx)>ABS(dminx) PROCform(
dmaxx) ELSE PROCform(dminx)
1100 dmaxx=dmaxx*10^(-e(1))
1110 dminx=dminx*10^(-e(1))
1120 exponent%(0)=e(1)
1130 IF ABS(dmaxy)>ABS(dminy) PROCform(
dmaxy) ELSE PROCform(dminy)
1140 dmaxy=dmaxy*10^(-e(1))
1150 dminy=dminy*10^(-e(1))
1160 exponent%(1)=e(1)
1170 PROClimits(dminx,dmaxx,6)
1180 sminx=L:smaxx=H:stpx=S
1190 PROClimits(dminy,dmaxy,4)
1200 sminy=L:smaxy=H:stpy=S
1210 vecx=smaxx-sminx:vecy=smaxy-sminy
1220 p=FNScreeny(0)
1230 IF p<150 OR p>850 HEIGHT=150 ELSE
HEIGHT=p
1240 p=FNScreenx(0)
1250 IF p<200 OR p>1080 POSN=200 ELSE P
OSN=p
1260 PROChorizontal(POSN,HEIGHT)
```

Self Numbering Graph Program

```
1270 PROCvertical(POSN,HEIGHT)
1280 piece=FNpiece(sminx,smaxx)
1290 mtx=10^(exponent%(0))
1300 mty=10^(exponent%(1))
1310 MOVEFNscreenx(sminx),FNscreeny(FNfunction(sminx*mtx)/mty)
1320 FOR t=sminx TO smaxx STEP piece
1330 value=FNfunction(t*mtx)/mty
1340 IF undefined%=FALSE DRAW FNscreenx(t),FNscreeny(value) ELSE MOVE FNscreenx(t),FNscreeny(value)
1350 NEXT
1360 ENDPROC
1370 :
1380 DEF FNpiece(a,b)=(b-a)*4/800
1390 :
1400 DEF PROCdimension
1410 DIM s(1):DIM e(1):REM forms
1420 DIM m%(3):REM nice table
1430 m%(0)=3:m%(1)=1:m%(2)=2:m%(3)=5
1440 DIM sk(4):REM nice stack
1450 DIM exponent%(1)
1460 ff%=1
1470 ENDPROC
1480 :
1490 DEF PROClimits(min,max,N%)
1500 LOCAL step,stp
1510 sk(1)=0:sk(2)=0:sk(3)=0
1520 step=(max-min)/N%
1530 stp=FNniceNumber(step)
1540 L=INT(min/stp)*stp
1550 H=FNroundUp(max/stp)*stp
1560 S=stp
1570 ENDPROC
1580 :
1590 DEF FNniceNumber(num)
1600 LOCAL z%,j%,number%
1610 FOR z%=-2 TO 7
1620 FOR j%=1 TO m%(0)
1630 number=m%(j%)*10^z%
1640 PROCstack(number,num)
1650 NEXT:NEXT
1660 =sk(0)
1670 :
1680 DEF PROCstack(n,want)
1690 sk(1)=sk(2)
1700 sk(2)=sk(3)
1710 sk(3)=ABS(n-want)
```

```
1720 IF sk(1)>=sk(2) AND sk(3)>=sk(2) sk(0)=sk(4)
1730 sk(4)=n
1740 ENDPROC
1750 :
1760 DEF FNround(n)
1770 LOCAL a,n1,n2
1780 n1=INT(n):n2=n1+1
1790 IF (n-n1)<(n2-n) a=n1 ELSE a=n2
1800 =a
1810 :
1820 DEF FNroundUp(n)
1830 LOCAL a
1840 IF INT(n)=n a=n ELSE a=INT(n)+1
1850 =a
1860 :
1870 DEF PROCform(number)
1880 LOCAL s%,l%,e%,mantissa
1890 IF number=0 s(0)=0:s(1)=0:e(0)=0:e(1)=0:GOTO 2000
1900 s%=FNround(LOG(ABS(number)))
1910 l%=LOG(ABS(number))
1920 e%=INT(l%/3)*3
1930 mantissa=number*10^(-e%)
1940 IF ABS(mantissa)<1 e%=e%-3:mantissa=mantissa*1000
1950 s(0)=number*10^(-l%)
1960 s(1)=l%
1970 IF ABS(s(0))<1 s(1)=s(1)-SGN(s(0)):s(0)=s(0)*(10^SGN(s(0)))
1980 e(0)=mantissa
1990 e(1)=e%
2000 ENDPROC
2010 :
2020 DEF PROCtext(text$,x%,y%,p$)
2030 LOCAL l%
2040 l%=LEN(text$)-1
2050 IF LEFT$(text$,1)=CHR$32 l%=LEN(text$)-1 ELSE l%=LEN(text$)
2060 text$=RIGHT$(text$,l%)
2070 l%=l%*16
2080 y%=y%+8
2090 IF p$="R" x%=x%-l%
2100 IF p$="C" x%=x%-.5*l%
2110 MOVEx%,y%
2120 VDU5
2130 PRINT;text$;
2140 VDU4
```

Self Numbering Graph Program

```
2150 ENDPROC
2160 :
2170 DEF FNscreenx(x)=200+880*(x-sminx)
/vecx
2180 :
2190 DEF FNscreeny(y)=150+700*(y-sminy)
/vecy
2200 :
2210 DEF PROChorizontal(posn,height)
2220 MOVE200,height:DRAW 1080,height
2230 FORx=sminx TO smaxx STEP stpx
2240 sx%=FNscreenx(x)
2250 MOVEsx%,height:DRAWsx%,height-16
2260 PROCtext(FNsurpress(LEFT$(STR$(x),
5)),sx%,height-40,"C")
2270 NEXT
2280 IF exponent%(0)=0 GOTO 2330
2290 PROCtext("x10",1180,height-90,"R"
)
2300 PROCtext(STR$(exponent%(0)),1180,h
eight-74,"L")
2310 l%=LEN(STR$(exponent%(0)))*16
2320 PROCtext(" ",1180+l%,height-90,"L"
)
2330 ENDPROC
2340 :
2350 DEF PROCvertical(posn,height)
2360 MOVEposn,150:DRAWposn,850
2370 FORy=sminy TO smaxy STEP stpy
2380 sy%=FNscreeny(y)
2390 MOVEposn,sy%:DRAWposn-16,sy%
2400 PROCtext(FNsurpress(LEFT$(STR$(y),
5)),posn-30,sy%,"R")
2410 NEXT
2420 IF exponent%(1)=0 GOTO 2470
2430 PROCtext("x10",posn-130,802,"R")
2440 PROCtext(STR$(exponent%(1)),posn-1
30,818,"L")
2450 l%=LEN(STR$(exponent%(1)))*16
2460 PROCtext(" ",posn-130+l%,802,"L")
2470 ENDPROC
2480 :
2490 DEF PROCbox(x1,y1,x2,y2,t%)
2500 IF t%=1 MOVE x2,y1:DRAWx1,y1:DRAWx
1,y2:PLOT29,x2,y2:PLOT29,x2,y1 ELSE MOVE
x2,y1:PLOT29,x1,y1:PLOT29,x1,y2:DRAWx2,
y2:DRAWx2,y1
2510 ENDPROC
2520 :
```

```
2530 DEF PROCeditLimits(A,B,C,D)
2540 LOCAL X,Y,xl,xh,y1,yh,g%
2550 xh=FNscreenx(B*10^-exponent%(0))
2560 xl=FNscreenx(A*10^-exponent%(0))
2570 yl=FNscreeny(C*10^-exponent%(1))
2580 yh=FNscreeny(D*10^-exponent%(1))
2590 *FX4,1
2600 GCOL3,1
2610 PROCbox(xl,y1,xh,yh,ff%)
2620 REPEAT
2630 g%=GET
2640 PROCbox(xl,y1,xh,yh,ff%)
2650 X=0:Y=0
2660 IF g%=136 X=-2:IF INKEY-1 X=-10
2670 IF g%=137 X=2:IF INKEY-1 X=10
2680 IF g%=138 Y=-4:IF INKEY-1 Y=-24
2690 IF g%=139 Y=4:IF INKEY-1 Y=24
2700 IF ff%=1 xl=xl+X:yl=y1+Y ELSE xh=x
h+X:yh=yh+Y
2710 IF g%=88 OR g%=120 ff%=2
2720 IF g%=90 OR g%=122 ff%=1
2730 PROCbox(xl,y1,xh,yh,ff%)
2740 UNTIL g%=13
2750 IF xl>xh temp=xh:xh=xl:xl=temp
2760 IF yl>yh temp=yh:yh=yl:yl=temp
2770 MINX=(xl-200)*vecx/880+sminx
2780 MAXX=(xh-200)*vecx/880+sminx
2790 MINY=(yl-150)*vecy/700+sminy
2800 MAXY=(yh-150)*vecy/700+sminy
2810 MAXX=MAXX*10^exponent%(0):MINX=MIN
X*10^exponent%(0)
2820 MAXY=MAXY*10^exponent%(1):MINY=MIN
Y*10^exponent%(1)
2830 GCOL0,1
2840 ENDPROC
2850 :
2860 DEF FNsurpress(text$)
2870 LOCAL l%
2880 IF INSTR(text$,".")=0 THEN GOTO 29
50
2890 REPEAT
2900 l%=LEN(text$)
2910 IF RIGHT$(text$,1)="0" text$=LEFT$(
text$,l%-1)
2920 UNTIL l%=0 OR RIGHT$(text$,1)<>"0"
2930 l%=LEN(text$)
2940 IF RIGHT$(text$,1)="." text$=LEFT$(
text$,l%-1)
2950 =text$
```

B

A Disc Organiser for ADFS (1)

by Peter Miles

On a number of occasions I have wanted to look at the catalogues of two ADFS discs simultaneously and then to copy files from one to the other. It is frustrating and time consuming to have to mount each disc in turn and catalogue the appropriate directory, then remember the correct syntax for the copy command. If, like me, you use DOS machines at work, you can easily get confused by the different syntaxes of DOS and ADFS commands.

```
Drive 0          Drive 0
Letters          $
                ORGANISER1
                Prog1
                Prog2
                TempFile
                Text
                ↑
↑↑↑ RTN Create directory ESC
```

The directory display screen

The program presented here solves these problems by providing a method within ADFS to organise files on two discs at a time by means of a clear screen display which shows both disc catalogues for any directory together. For those with only a single disc drive, the program will show two different directories simultaneously. File handling commands such as copy, delete, rename etc. are selected from the menu having chosen the file name by placing the cursor bar over it. Additionally, the program can be used to load, run and examine the files

which are found in the disc catalogue. A simple routine decides what type the file is (e.g. Basic, View, data etc.) and directs the program accordingly.

The program is written entirely in Basic, is well structured and easy to follow. It is presented in 2 parts due to its length. This month's listing provides the backbone of the program: the cataloguing facilities and the ability to move through the directory structures of the two discs. Also included in this part is the copy facility, thus providing the opportunity to compare your backup disc with the original and copy files across as necessary. Next month's listing will include all the other procedures to provide the remaining features outlined above.

ENTERING THE PROGRAM

Type the program in and save to disc using the file name Organiser1. Do not change any line numbers as these must tie in with the line numbers in part 2. The REMs in lines 2680, 2690 and 2730 can be omitted if desired as these only serve as a reminder of new lines which will be included in part 2. Take great care when typing in the error trapping routine in lines 270-380 because a mistake here can cause the machine to hang. I suggest the error trapping is checked as soon as DEFPROCend has been typed in (i.e. having typed up to line 1230) in order to prevent problems later. If the incomplete program (i.e. up to line 1230) is run then an error should be successfully flagged in the magenta window, and pressing Escape should put the small red window on the screen and allow you to terminate the (part-complete) program by selecting option 2. Having tested this section, complete the typing of listing 1 and save it before running.

A Disc Organiser for ADFS

USING THE PROGRAM

Before running the program make sure the current filing system is ADFS (by performing *ADFS if necessary) otherwise a 'Bad Command' error will be given.

When the program is started it prompts the user to insert the disc(s) to be 'organised'. Following this, the user must decide whether to display one or two 'panels'. The panel is a coloured text window in which the disc catalogue will be displayed. For each panel selected, the user must specify the drive to be assigned. The program offers defaults for all these settings and pressing any invalid key will enter the default value. When the catalogue is displayed, the cursor bar can be moved up or down using the cursor keys, and if two panels are being displayed, the cursor bar can be moved to the other panel either with the left or right cursor key or TAB.

Pressing Return when the cursor bar is on a directory name will move into that directory, while pressing Return on a filename will offer the opportunity of copying that file to the drive/directory displayed on the other panel. This option is selected by pressing 2. Although only one option is available at this stage, selecting it with 2 is consistent with the operation following the addition of part 2 of the program. Having pressed 2, the file will be copied after confirmation which is requested in the large cyan window. Note that the visual approach demands that copying a file is only allowed from the drive/directory on one panel to the drive/directory on the other.

Pressing Escape will leave the menu or prompt window and return you to the catalogue screen. Pressing Escape again will cause the little red exit/restart window to appear; select one of the options or press Escape to remove the window.

ERROR TRAPPING

A considerable amount of error trapping is necessary in a program of this sort to make it foolproof to the user. Most incorrect inputs are trapped by the program. However, to minimise the amount of code, the error trapping routine in lines 270-380 is used to display the error messages provided by ADFS, rather than trap any disc filing errors within the program itself. This is done by calling PROCwindow to make the presentation consistent. Errors with numbers 146 and above are ADFS generated so the program line number is not displayed to the user, whereas if ERR is less than 146 the program line number is displayed. This helps in program debugging.

WINDOWS

I consider the visual presentation of the program to be particularly important. The attractive presentation of this utility has been achieved by using coloured text windows in mode 7. Each panel is an example of this, while other messages appear in a central window of size and colour appropriate to the type of message. An examination of PROCwindow will reveal that this procedure not only creates the window and colours it in, but also lines the left hand side with control characters to determine the colour of any text subsequently written.

One difficulty that this display presents is the restoration of the screen after the window is finished with. This is achieved by taking a 'snap-shot' of the screen before a window might be used, then replacing the screen with the 'snap-shot' in order to erase the window. The 'snap-shot' consists of a copy of screen memory which is stored in sideways RAM number 4 (see line 170), and read back when required (e.g. line 370). The screen has to be saved whenever the program

leaves FNselect (line 2560) because this function may have scrolled the catalogue list in either or both panels.

ESCAPE

The Escape key is the exit route from any menu or screen, but the manner in which it achieves this varies through the program. When the cursor bar is present on the catalogue screen the program is running FNselect and here the normal escape action is turned off using *FX229,1. Pressing Escape will then return ASCII 27 and this value is passed from the function into code%. However, as the function is terminated *FX229,0 turns normal escape action on. PROCcontrol recognises code%=27 (line 2820) and terminates, taking the program to line 190 and causing the exit/restart window to be drawn. If Escape is pressed again the normal interrupt routine is called so control is passed to the error trapping routine in lines 270-380. The exit from this routine is controlled by the variable errflag%, so the program can proceed to the appropriate place.

PART 2

Next month all the remaining file handling facilities will be presented and the file menu will be expanded to complete the program.

```

10 REM Program Organiser1
20 REM Version B3.1
30 REM Author Peter Miles
40 REM For ADFS only
50 REM BEEBUG December 1991
60 REM Program subject to Copyright
70 :
100 MODE 7
110 ON ERROR GOTO 270
120 PROCinit
130 PROCstart
140 PROCpanelinfo("left"):PROCbackgrou
nd("left"):PROCdisplay("left")
150 IF NOT rpanel% THEN 170
160 PROCpanelinfo("right"):PROCbackgro

```

```

und("right"):PROCdisplay("right")
170 *SRWRITE 7C00 7FE7 8000 4
180 PROCcontrol
190 IF endflag% THEN PROCend
200 PROCwindow("S",15,129,131)
210 PRINT"Redefine panels 1""Finish
";SPC11;"2"
220 temp$=GET$
230 IF temp$="1" THEN MODE 7:GOTO 130
240 IF temp$<>"2" THEN 220
250 PROCend
260 :
270 REM ERROR CONTROL
280 IF ERR=17 THEN 340
290 PROCwindow("L",12,133,131)
300 PRINT"Error:":REPORT
310 IF ERR<146 THEN PRINT""at program
line ",ERL
320 PRINT""...any key to continue."
330 REPEAT UNTIL GET:VDU26:CLS
340 IF errflag%=0 THEN rpanel%=FALSE:G
OTO200
350 VDU26,23 11
360 clrflag%=FALSE:*FX4,1
370 CLS:*SRREAD 7C00 7FE7 8000 4
380 GOTO 180
390 :
1000 DEFPROCinit
1010 errflag%=0:*SHADOW 1
1020 bufsiz%=7168:*FX4,1
1030 DIM dir$(47),file$(47),list%(7),ll
ist$(48),rlist$(48)
1040 DIM block% 18,buffer% 30,cblock% 1
3,cbuffer% bufsiz%,control% 12,output% 1
6,string% 11
1050 clrflag%=FALSE:endflag%=FALSE:self
lag%=FALSE:rpanel%=FALSE
1060 r$=CHR$129:g$=CHR$130:y$=CHR$131:b
$=CHR$132:m$=CHR$133:c$=CHR$134:bl$=CHR$
156:bac$=CHR$157
1070 ENDPROC
1080 :
1090 DEFPROCwindow(size$,top%,backcol%,
textcol%)
1100 VDU26,30:PRINT'
1110 IF size$="L" THEN 1160
1120 FOR N%=top% TO top%+5
1130 IF rpanel% AND NOT clrflag% PRINTT
AB(5,N%)CHR$backcol%;bac$;CHR$textcol%;S

```

A Disc Organiser for ADFS

```
PC20;b$;bac$ ELSE PRINTTAB(5,N%)CHR$back
col$;bac$;CHR$textcol$;SPC20;bl$
1140 NEXT
1150 VDU28,8,top%=5,26,top%+1:ENDPROC
1160 FOR N%=top% TO top%+10
1170 IF rpanel% AND NOT clrflag% PRINTT
AB(3,N%)CHR$backcol$;bac$;CHR$textcol$;S
PC28;b$;bac$ ELSE PRINTTAB(3,N%)CHR$back
col$;bac$;CHR$textcol$;SPC28;bl$
1180 NEXT
1190 VDU28,6,top%+10,32,top%+1:ENDPROC
1200 :
1210 DEFPROCend
1220 VDU26,23 1 11:*FX4,0
1230 CLS:END
1240 :
1250 DEFPROCstart:errflag%=0
1260 title$=g$+bac$+CHR$141+m$+"Disc Or
ganiser "+bl$
1270 PRINTTAB(6)title$'TAB(6)title$
1280 PRINT''c$;"Insert disc(s)''b$;
"...any key when ready.":REPEAT UNTIL GE
T
1290 PRINT''c$;"How many panels (1 or
2)?";y$;"1";:VDU8:temp=GET
1300 IF temp=50 THEN PRINT"2":rpanel%=T
RUE ELSE PRINT"1":rpanel%=FALSE
1310 PRINT'c$;"Drive for left panel (0
or 1)?";y$;"0";:VDU8:temp=GET
1320 IF temp=49 THEN PRINT"1":ldrive%=1
ELSE PRINT"0":ldrive%=0
1330 IF NOT rpanel% THEN rdrive%=2:GOTO
1360
1340 PRINT'c$;"Drive for right panel (
0 or 1)?";y$;"0";:VDU8:temp=GET
1350 IF temp=49 THEN PRINT"1":rdrive%=1
ELSE PRINT"0":rdrive%=0
1360 ldir$="$":rdir$="$":side$="left"
1370 P%=0:R%=2:lrow%=2:rrow%=2
1380 VDU23 11:CLS:ENDPROC
1390 :
1400 DEFPROCcatalogue
1410 LOCAL N%,A%,C%,X%,Y%,name$
1420 FOR N%=0 TO 9 STEP 3:control%!N%=0
:NEXT
1430 filecount%=0:dircount%=0
1440 REPEAT
1450 control%?1=string%MOD256
1460 control%?2=string%DIV256
```

```
1470 control%?5=1
1480 X%=control%MOD256:Y%=control%DIV25
6
1490 C%=0:A%=8:CALL(&FFD1)
1500 IF control%?5<=0?(string%+1?stri
ng%)=13:name$=$(string%+1)
1510 IF name$="" OR control%?5>0 THEN 1
540
1520 IF FNfileinfo(name$)=1 OR FNfilein
fo(name$)=&FF THEN filecount%=filecount%
+1:file$(filecount%)=name$
1530 IF FNfileinfo(name$)=2 THEN dircou
nt%=dircount%+1:dir$(dircount%)=name$
1540 UNTIL control%?5>0
1550 ENDPROC
1560 :
1570 DEFNFNfileinfo(f$)
1580 LOCAL X%,Y%
1590 $output%=f$
1600 block%?0=output%MOD256
1610 block%?1=output%DIV256
1620 X%=block%MOD256:Y%=block%DIV256
1630 A%=5:=(USR(&FFDD)AND&FF)
1640 :
1650 DEFPROCpanelinfo(side$)
1660 LOCAL N%,item%
1670 PROCdrive(side$):PROCcatalogue
1680 IF side$="right" THEN 1930
1690 VDU28,0,1,19,0:CLS:VDU28,0,2,19,0
1700 PRINT r$;bac$;c$;"Drive ";ldrive%;
TAB(18);bl$
1710 PRINT r$;bac$;c$;FNdirname(ldir$);
TAB(18);bl$
1720 item%=0:IF ldir$<>"$" THEN llist$(
1)=r$+^ DIR":item%=1
1730 IF dircount%=0 THEN 1820
1740 FOR N%=1 TO dircount%
1750 item%=item%+1
1760 llist$(item%)=r$+dir$(N%)
1770 REPEAT
1780 llist$(item%)=llist$(item%)+ "
1790 UNTIL LEN(llist$(item%))=12
1800 llist$(item%)=llist$(item%)+&DIR"
1810 NEXT N%
1820 IF filecount%=0 THEN 1900
1830 FOR N%=1 TO filecount%
1840 item%=item%+1
1850 llist$(item%)=m$+file$(N%)
1860 REPEAT
```



```

1870 llist$(item%)=llist$(item%)+ "
1880 UNTIL LEN(llist$(item%))=15
1890 NEXT N%
1900 list%(0)=item%:list%(2)=1
1910 IF list%(0)<22 THEN list%(4)=list%
(0) ELSE list%(4)=22
1920 list%(6)=list%(4)+1:ENDPROC
1930 VDU28,20,1,39,0:CLS:VDU28,20,2,39,
0
1940 PRINT r$;bac$;c$;"Drive ";rdrive%;
TAB(18);bl$
1950 PRINT r$;bac$;c$;FNdirname(rdir$);
TAB(18);bl$
1960 item%=0:IF rdir$<>"$" THEN rlist$(
1)=r$+"^" DIR":item%=1
1970 IF dircount%=0 THEN 2060
1980 FOR N%=1 TO dircount%
1990 item%=item%+1
2000 rlist$(item%)=r$+dir$(N%)
2010 REPEAT
2020 rlist$(item%)=rlist$(item%)+ "
2030 UNTIL LEN(rlist$(item%))=12
2040 rlist$(item%)=rlist$(item%)+ "DIR"
2050 NEXT N%
2060 IF filecount%=0 THEN 2140
2070 FOR N%=1 TO filecount%
2080 item%=item%+1
2090 rlist$(item%)=m$+file$(N%)
2100 REPEAT
2110 rlist$(item%)=rlist$(item%)+ "
2120 UNTIL LEN(rlist$(item%))=15
2130 NEXT N%
2140 list%(1)=item%:list%(3)=1
2150 IF list%(1)<22 THEN list%(5)=list%
(1) ELSE list%(5)=22
2160 list%(7)=list%(5)+1:ENDPROC
2170 :
2180 DEFPROCdrive(side$)
2190 IF side$="left" THEN OSCLI("MOUNT
"+STR$ldrive%) ELSE OSCLI("MOUNT "+STR$r
drive%)
2200 IF side$="left" THEN OSCLI("DIR "+
ldir$) ELSE OSCLI("DIR "+rdir$)
2210 ENDPROC
2220 :
2230 DEFPROCdisplay(type$)
2240 LOCAL N%,panel$,siden%
2250 IF type$="up" AND list%(P%+2)=1 TH
EN ENDPROC

```

```

2260 IF type$="up" THEN list%(P%+2)=lis
t%(P%+2)-1:list%(P%+4)=list%(P%+4)-1:sid
eno%=P%
2270 IF type$="down" AND list%(P%+4)=li
st%(P%) THEN ENDPROC
2280 IF type$="down" THEN list%(P%+2)=1
ist%(P%+2)+1:list%(P%+4)=list%(P%+4)+1:s
ideno%=P%
2290 IF type$="left" THEN siden%=0 ELS
E IF type$="right" THEN siden%=1
2300 IF type$="left" OR type$="right" T
HEN panel$=type$ ELSE panel$=side$
2310 PROCclearpanel(panel$)
2320 IF list%(siden%)=0 THEN VDU26:END
PROC
2330 FOR N%=list%(siden%+2) TO list%(s
ideno%+4)
2340 IF siden%=0 THEN PRINT llist$(N%)
ELSE PRINT rlist$(N%)
2350 NEXT
2360 VDU26:ENDPROC
2370 :
2380 DEFPROCbackground(side$)
2390 IF side$="left" THEN VDU28,0,24,19
,2 ELSE VDU28,20,24,39,2
2400 CLS
2410 IF side$="left" THEN FOR N%=0 TO 2
4:PRINTTAB(18)bl$:NEXT:VDU30
2420 FOR N%=0 TO 24:PRINT b$;bac$;SPC16
;bl$:NEXT:ENDPROC
2430 :
2440 DEFPROCclearpanel(side$)
2450 IF selflag% THEN 2480
2460 IF side$="left" THEN VDU28,2,23,17
,2 ELSE VDU28,22,23,37,2
2470 CLS
2480 IF side$="left" THEN VDU28,2,24,17
,2 ELSE VDU28,22,24,37,2
2490 ENDPROC
2500 :
2510 DEFPROCcontrol:errflag%=1
2520 REPEAT
2530 PRINTTAB(0,24)g$;bac$r$;CHR$91;CH
R$93;CHR$94;" RTN C:create directory
ESC ";bl$;
2540 code%=FNselect
2550 PRINTTAB(0,24)SPC36;
2560 *SRWRITE 7C00 7FE7 8000 4

```

Continued on page 54

Patterns from a Keyboard (2)

by Jeff Gorman

Last month's article offered a brief explanation of how to use the main program. This section offers listings for the promised reverse feed and the report options, and also some insight into the structure of the core elements of the program.

TO COMPLETE THE PROGRAM

For safety's sake make sure you have at least one backup copy of last month's listing. Load this program and enter PROCreverse and PROCreport from this month's listing. Remove the REMs from the front of lines 2360 and 2370, and save the complete version using a new filename. You can now test the new facilities.

PROGRAM OUTLINE

Readers may recall that the program started as a straightforward printout of a textual statement or keyboard character covering almost the entire area of an A4 sheet, but like Harriet Beecher Stowe's Topsy, it "grow'd".

CHANGING THE TYPEFACE

One feature of this growth involves using character widths other than the default "Pica" (10 characters/inch) offered by most printers. The density of print can be modified by selecting the alternative pitches of "Elite" at 12 characters/inch or "Compressed" at 17 chrs/inch. Each of the three in turn can be modified to print at twice their normal width in "Expanded" mode,

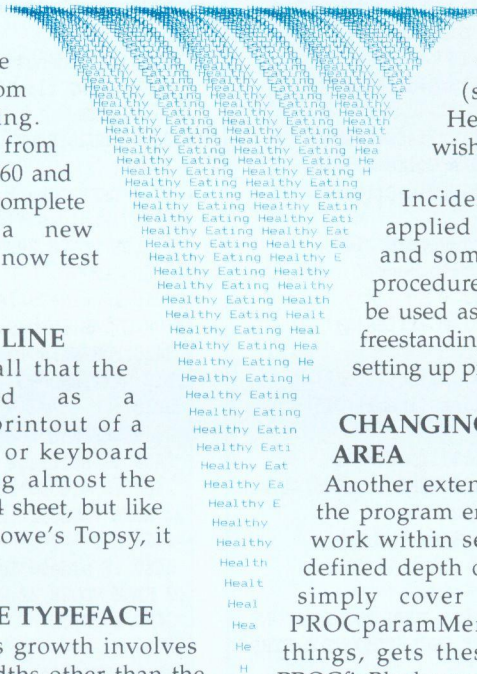
and/or as superscript or subscript characters. PROCstyleMenu puts the options on screen. FNmenu enables an asterisk shaped line selector to be tracked up and down the left-hand margin to determine which VDU codes are sent to the printer each time Return is pressed. It also uses the array *typ\$()* to note the style for use in the report stage. Line 1120 offers an option to replace the parameters "0,0," with one's own VDU codes so as to set up options specific to one's own printer (such as "Double Height") if the user so wishes.

Incidentally, the coding applied to PROCstyleMenu and some of its dependent procedures could fairly easily be used as a basis for a simple freestanding utility program for setting up printer codes.

CHANGING THE PRINTED AREA

Another extension to the scope of the program enables the printer to work within set margins and to a defined depth of print, rather than simply cover the entire sheet. PROCparamMenu, amongst other things, gets these dimensions and PROCfixBlock uses VDU codes to set the margins and also gets the depth.

When printing a text in a continuously joined stream within defined margins, the printer has to know how many complete texts are needed to cover the area and how many characters are needed to fill up to the very end of the



final line. FNfixLne calculates the necessary number of complete texts (*numPhrs%* in PROCfixBlock) and the remainder (*rem%*).

"CROPPED" AND "UNMODIFIED" MODES

A third elaboration involves the printing of a number of text repetitions suitably padded to fit between the set margins, and then, as printing progresses, cropping a character from the beginning of each line, and in "Unmodified Mode", putting characters back again at the end of the line. Simple cropping produces output of a triangular shape with a downward facing apex and a strongly marked oblique pattern. Restoring the characters generates a rectangular output in the same style.

Inverting the subtractive process by starting from one character and adding one per line inverts the triangle. Readers who have already experimented will have found that the shape of the triangles depends on whether the printer left justifies, centralises, or right justifies.

The actual cropping is achieved by changing the variable *chrslne%* in the statement below:

```
line$=RIGHT$(maxLne$,chrslne%)
```

If *chrslne%* is decreased by one unit with each pass of the printer head, the left-hand character will be nipped from the line, causing each printed character to be offset by one unit to the left, thereby generating a diagonally "slewed" pattern.

If *line\$=LEFT\$(maxLne\$,chrslne%)* and *chrslne%* is increased by one unit per line, a triangle with an upwards facing apex is produced.

The characters needed to be added to the end of the line to generate rectangular cropped mode output are found by making:

```
lne$=RIGHT$(maxLne$,chrslne%)+MID$(line$,rem%,1)
```

rem% being the difference between the full and cropped line lengths.

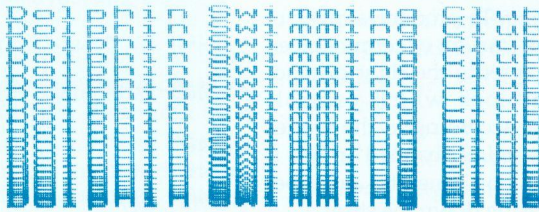
CHANGING THE PAPER FEED

To comply with the depth instruction, the printer needs to know how many lines need to be printed to move the paper forwards to reach the intended depth (variable *depth*). This can be less than straightforward since the default value of six lines per inch can be changed if the user selects a non-standard value for *Feed* during the setting up process.

Paper feed per line can range from a movement as fine as 1/216" to 255/

216" which is approximately equal to 1.2". The actual maximum feed has been limited to a little under 0.6" because the effect of a greater line separation seems of little aesthetic value. If the spacing is uniform, simple arithmetic can cope with finding the number of lines to make up a given depth, but if the line spacing is gradually increased or decreased as the printing progresses (see "Banding" below), calculation becomes more difficult, so a dummy run, operated by FNnumFeeds, is instituted to produce the desired result by running through a REPEAT loop, counting the passes as it goes, until the vertical displacement approximately equals the depth set at the input stage.

The penultimate line of FNnumFeeds generates *end%* which is used if reverse



Patterns from a Keyboard

feed is to be operated at the end of the print run.

BANDING

Within the printer control code for line spacing (VDU1,27,1,51,1,*n*) *n* controls the extent of the feed (see PROCshade). By placing *n* (actually *spac%*) in a REPEAT loop (in PROCprintLine), which increments (or decrements) *spac%* by one unit at each pass of the loop, a gradually increasing or decreasing spacing can be obtained, giving a "banded" effect.

Note that shaped blocks are produced by starting with a line length of one character, or starting at full length and progressing to the inverse. The number of feeds needed to complete the block is therefore determined by the number of characters per line. To produce a block of the right size the starting or terminal feed (*spac%*) has to be set to the right distance (in PROCprintBlock which controls PROCprintLine), according to the modes the user has selected. If "banded" mode is selected, a repeat loop generates a number of bands progressing down the sheet.

The illustrations this and last month demonstrate some of the effects which can be achieved. This is very much a program which rewards time spent experimenting with different inputs and parameters.

```
10 REM Program PrtPtn2
20 REM Version 1.1
30 REM Author Jeff Gorman
40 REM BEEBUG December 1991
50 REM Program subject to copyright
4410 :
4420 DEF PROCreverse
4430 PROCwnd(0,24,39,17,-1)
4440 PRINT y$ "Reverse to start (Y/N)? "
";
4450 IF FNip(1,-1,"YN")="N" ENDPROC
4460 stage%=back% DIV 216
4470 IF stage% >0 FOR roll%=1 TO stage%
:VDU2,1,27,1,106,1,216,3:NEXT
```

```
4480 extra%=back% MOD 216
4490 VDU2,1,27,1,106,1,extra%,3
4500 revr=TRUE:VDU2,1,27,1,64,3:ENDPROC
4510 :
4520 DEF PROCrecord:PROCwnd(0,24,39,17,-1)
4530 PRINTY$"Record parameters (Y/N)? "
;
4540 IF FNip(1,-1,"YN")="N" ENDPROC
4550 VDU2,1,27,1,64,1,15,3:*FX3,10
4560 IF bnd%=1 bnd$=" band, " ELSE bnd$
="- bnd, "
4570 PRINT "Font & Mode";
4580 PRINT TAB(17);:FOR no%=1 TO num%-1
4590 PRINT typ$(no%) SPC1;
4600 NEXT:@%=&02010A
4610 PRINT "Graphic unit"TAB(17)[" A$
"]"
4620 IF shap prDpth=vert%/216 ELSE prDp
th=depth
4630 IF crpd AND band prDpth=prDpth*bnd
%
4640 PRINT"Left Margin Right Margin "
;
4650 PRINT"Block Width Format (S/R) "
;
4660 PRINT"Justification (L/C/R) ";
4670 PRINT"Line format (J/C/U) ";
4680 PRINT"Bands (G/B/U) Num.Bands "
4690 PRINT"Approx ";lm""";
4700 PRINT TAB(13)"Approx ";rm""";
4710 PRINT TAB(27)"Approx ";pbw;""";
4720 PRINT TAB(42)sh$ TAB(60)jstf$;
4730 PRINT TAB(81)js$ TAB(103)gbu$;:@%
=&0A
4740 PRINT TAB(116);bnd%
4750 PRINT"Incr/Decr. feed Paper feed
";
4760 PRINT"Apex (U/D) Print depth ";
4770 PRINT"Stripes (S/V) Print Mode"
4780 PRINT TAB(7) id$ TAB(21);feed%;
4790 @%=&02010A
4800 PRINT TAB(32) ud$ TAB(45);prDpth"
""";
4810 PRINT TAB(58) ov$;
4820 PRINT TAB(68) prefix$ "directional
";
4830 FOR lneFeed%=1 TO 8:PRINT:NEXT:*FX
3,0
4840 VDU3
4850 ENDPROC
```

The Future of the BBC Micro in Education

by Mark K. Sealey

A couple of months ago Acorn announced the new A5000 computer system. In launching the latest in its 32-bit range (which also has a Learning Curve option specifically for education) Acorn ended speculation and yet another gap in the market was plugged.

But the company justifiably renowned above all others for its support of education also showed its intention to stick with the highly successful format established by the Archimedes 400 series, 540 and A3000 machines.

Although the new range represents an upgrade in appearance and operating system, it is in many ways more of the same. It is more of a successful formula that may tempt you once and for all away from the 8-bit technology - nearly ten years old now, a long time in computing - and into the world of the Wimp, Desktop and almost obligatory hard discs for your school or college.

Should you do so? Which factors are valid in influencing your decision? Are there not many things that the BBC B, Compact, Master and even Electron ranges will do just as well? This month's Beebug Education aims to help you in your choice.

HISTORY

The BBC micro was launched ten years ago next spring. Those around at that time will remember the excitement of being able to outperform significantly the Commodore Pets, the few TRS80s,

ZX81s and Spectrums and rather more RML 380Zs and 480Zs common both in secondary school science laboratories and in an increasing number of primary schools.

This outperformance was not, though, in terms of price but of operating speed, ease of use and several features of specific value to the educational user.

Most - if not all - of these features, which have a direct bearing on the way the computers are used in the curriculum of pupils of all ages, remain quite valid selling points for the BBC 8-bit machines to this day. A few of them are worthy of mention.

One of the most effective ways of introducing young children to and familiarising them with key concepts (and, for that matter, of motivating older pupils) is by the use of real objects as opposed to drawings or even films. With the BBC and its user port and some fairly basic software, real experiments (like temperature measurement) can be done with very few add-ons. An Archimedes would require nearly £100 to be spent on the right expansion card - the user port scores again.

Through its use (and that of other ports), the 8-bit computer can be made both to speak and to process sounds itself, and to digitise and scan images in a variety of ways. Thanks to mode 7, pupils commonly use the same screen font and some of the conventions of Teletext and Viewdata on-line systems.

The Future of the BBC Micro in Education

OBSOLESCENCE

But hardware does not last for ever. The BBC B, Electron and Compact are now officially obsolete. Acorn's technical support, if not always the most efficient, has usually been so thorough that it seems likely the older machines can be supported as far as spares and repairs are concerned for some time to come.

In a not dissimilar vein, many of the fundamental design principles established by Acorn from early on have proved so sensible that technicians (and the technically shy but adventurous) who might be afraid to familiarise themselves with a TOS, FCBs and the intricacies of CP/M are quite happy to spend time and effort effectively 'upgrading' software to run under new circumstances - say with a sideways RAM board fitted, or on two discs instead of one.

These design features include hierarchical directory filing systems (in ADFS), a very transparent yet accessible operating system in ROM, and a logic in the user interface whereby the same effect is obtained in several ways (e.g. turning the printer on by VDU 2 or ^B, or even PRINT CHR\$(2) for compatibility with older Basics).

In some senses it is quite remarkable that the command used ten years ago to set the baud rate at the serial port (*FX7,n) is still at the heart of any routine to achieve this with the Archimedes. There is a Nimbus emulation of BBC Basic; there is not a BBC emulation of RM Basic, although the 512 Second Processor can run CP/M!

Maybe much of the foregoing accounts for the very large user base, for the predictability of issues and ways of

meeting them which come up at INSET sessions throughout the country, and for the the generally very high level of expertise and formal support which follow any user of the BBC machines.

Although Acorn has never made the big time with high street retailers, the support for schools and colleges anxious to stick with their tried and tested hardware looks set to continue in a very real way for some time to come.

It must also be borne in mind that the home computer which crops up most in both the home and school environments is unlikely to be an Amiga (which scarcely gets a look-in educationally) or a Nimbus (which very very few homes can surely own) but a BBC B. It counts for quite a lot when all sorts of tasks begun at school can be completed at home and vice versa as well as speeding up the general process of familiarisation.

THE SOFTWARE BASE

Obviously, anyone making out the case in the early 80s for standardisation on BBCs in schools and colleges was aware of the dire lack of software. This is always the case when a new machine first appears - nothing could be further from the truth now. A glance at the specific educational catalogues like those produced by AVP, Ricketts or the major third-party producers makes it clear that there are many thousands of titles available to fulfil most educational needs - far, far more than those for any other hardware.

On the other hand, it has to be said that most of these titles and many more for PCs will run under the Archimedes 6502 and excellent new (v1.6) DOS emulators.

Exceptions to these are all those programs that are copyright protected and hence cannot be copied to a disc format readable by the 32-bit machines. You almost always need a 5.25" disc drive too.

It has always been the case that during the first few weeks and months of using a new computer, memory seems unlikely ever to be used up. It has always been the case, too, that it very soon is. Furthermore, it used to be claimed that memory on the BBC especially was in particularly short ration.

Nowadays there are few programs that do not make very imaginative and clever use of the amount of memory available, which is admittedly limited by Archimedes/A3000 standards. This is sometimes by use of sideways ROMs or RAM, expansion boards, shadow screen memory or just highly skilful programming. Sophisticated graphics (and to a lesser extent sound) in part compensate for those occasions when pupils cannot do what is required of them because of memory limitations. Maybe when compiling lengthy theses at University level Wordwise Plus won't do. But those basic workhorse word processors, databases, spreadsheets and art packages can still be very effective. The same goes for the enormous amount of good CAL software currently available.

SPECIAL NEEDS

A particular area which has been exceptionally well supported by the 8-bit computers is Special Needs. The number of peripherals (from the Concept Keyboard and touchscreen to photonic wand and Micro Mike) is in no small

way responsible for this. And a very good case can easily be made out that these peripherals work more effectively on the BBC than they do on any of its rivals. Certainly the software is there; the de facto standard for its publication is likely to be a BBC version first and others later - if at all.

Thanks largely to organizations like the SEMERCs and their offshoots, the know-how complements nicely the dedication of those who work in special education and who positively delight in being part of a process that disseminates and shares information about the use of the BBC - in newsletters, through the on-line databases, and other specialist but extremely vigorously used media.

THE PUBLICATIONS

A machine is only as good as its software base, then; but almost as important is the wealth (or lack) of written information that supports it. Here again, the picture is clear. Magazines like Beebug still do a sterling job in providing information and technical guidance and acting as a clearing house for hints and tips.

True, most of the other heavyweights have gone over predominantly (if not exclusively) to the 32-bit range. On the other hand, many of the journals (such as those produced by MAPE, MUSE, BLUG and many more) still quote examples and expect programs to run in, dare it be said, the lowest denominator - the good old BBC 8-bit format. The same is true of many books.

CONCLUSIONS

So, is the BBC really the lowest denominator? Is the technology so

Continued on page 26

Teletext and the Function Keys

by Denis Atkinson

Teletext in mode 7 (and mode 135 on the Master series) is a useful, fascinating and rewarding facility. Its operation is discussed (somewhat briefly) on pages 94 to 97 of the Welcome Guide for the Master 128, and pages 128 to 136 of the BBC User Guide. In addition, Appendix B of each Guide shows that the teletext control codes are to be found in the range 129 to 159. For example, codes 129 to 136 control access to the printing of text characters in the seven standard colours of red, green, yellow, blue, magenta, cyan and white. The remainder of the codes cover the colour of the graphics (145 to 151) and various special facilities such as double height (141). To print characters in (say) the colour red requires the insertion of CHR\$(129) between the instruction PRINT and the text, thus:

```
PRINT CHR$(129) "Message"
```

An alternative way of obtaining the same effect is to use the keyboard directly by pressing Shift and the function key f1. The change of colour to text, therefore, may be obtained by using Shift-f1 through to Shift-f7 (corresponding to the control codes 129 to 136). The Guides also list Shift-f8 for flashing text and Shift-f9 to return to non-flashing text.

Teletext graphics are produced in each of the seven colours from the control codes 145 to 151 followed by any character other than an upper-case letter, thus:

```
PRINT CHR$(145) "a"
```

In the case of graphics the alternative approach via the keyboard is Ctrl-f1 (for colour red) followed, of course, by the appropriate letter.

It should now be apparent that no alternative function key method is set out in the Guides for the remaining facilities of double height, return to normal height, conceal display, contiguous graphics, separated graphics, black background, new background, hold graphics and release graphics.

The greatest demand for most users of teletext is for changes in the colour of the text. It seems rather a pity, therefore, that such changes demand the use of two keys, viz Shift and a red function key.

To produce a program which provides access via the keyboard for all the teletext facilities and also uses just a single key operation for the text colours requires a search through the *FX commands. The lists of the *FX and VDU commands which are set out in the Appendices appear quite daunting, simply because of their length.

A detailed survey of these commands has recently appeared in BEEBUG (Vol.9 No.10 through to Vol.10 No.5), and they certainly repay careful study. The FX list, for example, reveals that *FX225, 226 and 227 provide control over the status of the function keys. Using *FX225,n (where n is in the range 2 to 255) assigns ASCII n to f0, ASCII(n+1) to f1 and so on through to ASCII(n+9) to f9. Now we know that codes 129 to 137 are our teletext controls for the colours plus flashing and steady, so *FX225,129 will cause f0 to generate the text colour red, f1 the colour green and so on through the seven colours, whilst f7 and f8 produce the flashing/non-flashing facilities.

So far, so good, but what about the other codes for the graphic colours and the special facilities? Turning again to Appendix B in the Guides, there are two unused codes, 138 and 139, so the remaining codes can be covered neatly by two further ranges of 140 to 149 and 150 to 159. All we need, then, is a similar access via the Shift-function key combinations for the range 140 to 149 and via Ctrl-function keys for the final ten of 150 to 159. These are readily provided by *FX226,140 and *FX227,150 which operate in the same way as *FX225 with Shift-f0 to f9 and Ctrl-f0 to f9 respectively, just as *FX225 does for f0 to f9. We now have a complete and simple way of making all the teletext facilities readily available from the keyboard using the function keys or combinations of them with Shift or Ctrl. Moreover, the text colour change controls are activated by a single key press.

The program given here presents an easy way to set up the function keys for all the teletext codes. When run, you can have the effects of the keys displayed if so desired, and then the correct *FX calls are made to change the effects of the function keys, as described above.

The three *FX commands 225, 226 and 227 have been incorporated into lines 220 to 260 as PROCfnkeys. The *FX225,129 in line 230, for example, causes f0 to generate ASCII 129, the code for teletext red, f1 to generate ASCII 130 (or green) and so on. *FX226,140 similarly causes Shift-f0 to access ASCII 140 (normal height) and so on through f1 to f9; and *FX227,150 establishes the Ctrl-f0 to f9 series of graphic cyan through to release graphics. It is worth observing that some of the function keys are not used. Also,

remember that the actual graphics characters are provided by pressing the non-upper case keys. It would be useful to print out a function keystrip (using, for example, the program from BEEBUG Vol.6 No.4) showing the effects of the keys, and then the program could be shortened by only retaining lines 10-80 and 100-200.

```
10 REM Program Teletex
20 REM Version B1.0
30 REM Author D.N.Atkinson
40 REM BEEBUG December 1991
50 REM Program subject to copyright
60 :
70 MODE 7
80 ON ERROR GOTO 120
90 PROCChelp
100 PROCfnkeys
110 END
120 IF ERR=17 THEN GOTO 70
130 REPORT:PRINT" at line ";ERL
140 END
150 :
160 DEF PROCfnkeys
170 *FX225,129
180 *FX226,140
190 *FX227,150
200 ENDPROC
210 :
220 DEF PROCChelp
230 CLS
240 PRINT TAB(4,2)CHR$141"FUNCTION KEY
DESIGNATIONS"
250 PRINT TAB(4,3)CHR$141"FUNCTION KEY
DESIGNATIONS"
260 FOR X%=0 TO 7
270 READ name$
280 PRINT TAB(6,X%+5)"f";X%;;PRINT TAB
(8,X%+5)CHR$(129+X%);name$
290 NEXT
300 PRINT TAB(18,12)"and";CHR$137"f8 S
TEADY"
310 PRINTTAB(1,13)"SHIFTf0":PRINTTAB(9
```

Teletext and the Function Keys

```
,13) "NORMAL HEIGHT"  
320 PRINTTAB(1,14) "SHIFTf1":PRINTTAB(8  
,14)CHR$141"DOUBLE HEIGHT"  
330 PRINTTAB(8,15)CHR$141"DOUBLE HEIGH  
T"  
340 FOR Y%=0 TO 4  
350 READ graph$  
360 PRINT TAB(18,Y%+5)CHR$135"SH.f";Y%  
+5;:PRINTTAB(24,Y%+5)CHR$(145+Y%);graph$  
370 NEXT  
380 FOR Y%=5 TO 6  
390 READ moregr$  
400 PRINT TAB(17,Y%+5)CHR$135"CTL.f";Y  
%-5;:PRINTTAB(24,Y%+5)CHR$(145+Y%);moreg  
r$  
410 NEXT  
420 FOR Z%=0 TO 2  
430 READ misc$  
440 PRINT TAB(6,Z%+16) "CTLf";Z%+2;:PRI
```

```
NT TAB(13,Z%+16)misc$  
450 NEXT  
460 FOR Z%=0 TO 3  
470 READ final$  
480 PRINT TAB(6,Z%+19) "CTLf";Z%+6;:PRI  
NT TAB(13,Z%+19)final$  
490 NEXT  
500 ENDPROC  
510 :  
520 DATA RED, GREEN, YELLOW, BLUE, MAGENTA  
, CYAN, WHITE, FLASHING  
530 DATA GRAPHIC RED, GRAPHIC GREEN, G  
RAPHIC YELLOW, GRAPHIC BLUE, GRAPHIC MAGE  
NTA  
540 DATA GRAPHIC CYAN, GRAPHIC WHITE  
550 DATA CONCEAL DISPLAY, CONTIGUOUS GR  
APHICS, SEPERATED GRAPHICS  
560 DATA BLACK BACKGROUND, NEW BACKGROU  
ND, HOLD GRAPHICS, RELEASE GRAPHICS B
```

The Future of the BBC Micro in Education (continued from page 23)

outworn that we should abandon it today? Certainly, the path towards the A3000 or A5000 is an attractive one. Certainly the true desktops (however good their 8-bit imitators) are a superior and more intuitive interface. True, the range of appealing software packages now available for the 32-bit range compares in many ways more favourably with what was available for the BBC four years after it first appeared. Multimedia applications like Magpie and Genesis, DTP applications like Impression and Ovation, and the multi-tasking nature of the Desktop, which allows pupils simply to drag or click on files for the relevant application to start, are a real boon to busy teachers and pupils anxious just to get on with it.

The reason for this advance, though, must be that the very programmers who

have created marvels like Pendown and Multistore were practised BBC experts before changing up. The 8-bit range served them well.

If it is sheer processing power you want, undoubtedly an ARM 3 with RiscOS 3, 4 Mbytes of RAM and a low-cost hard disc can scarcely be bettered anywhere. Nor will the 8-bit scene last for ever. If, on the other hand, it's the support, familiarity, vast range of serviceable software, reliable peripherals and add-ons, experts (or someone willing to have a go) round every corner and all the collective wisdom accrued over ten years, which only a classic attracts, then there's a lot of life in the old dog yet; witness the fact that BEEBUG sells all the trade-in BBCs it can get.

B

1st course

Program Logic (2)

Alan Wrigley continues his investigation into the use of logic in expressions and program construction.

In last month's *First Course* we considered the different ways in which the human brain and the Basic language look at logical constructions. I want to pursue this theme this month by first introducing the subject of "operator precedence".

An operator, in this sense, is a mathematical or boolean function such as +, -, AND, EOR etc. When we perform a mathematical calculation in our heads, or on paper, there are no hard and fast rules on operator precedence (I am talking here about everyday usage, not the formal discipline of maths); the "logical" way to handle a calculation to the human mind is to take each operator in turn from left to right. In other words, we bracket each successive element of the calculation starting from the left. Thus an expression such as:

$$8+4*5-3$$

will usually be calculated as:

$$((8+4)*5)-3$$

giving a result of 57. However, there is no rule which says that elements should be calculated sequentially like this; it is purely the western convention of reading from the left which suggests that this is the logical way to treat the calculation.

A computer language cannot work in this way. It must have rules to follow which are immutable. If you were to perform the calculation given above (without the brackets of course!) in Basic, you would get the answer 25, as a result of operator precedence being brought into play. Figure 1 shows the complete operator precedence table - an

Group 1	
-	Unary minus
+	Unary plus
NOT	Logical NOT
Function calls	
()	Brackets
! ? \$	Indirection operators
Group 2	
^	Exponentiation
Group 3	
*	Multiplication
/	Division
DIV	Integer division
MOD	Integer remainder
Group 4	
+	Addition
-	Subtraction
Group 5	
=	Equal
<>	Not equal
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
Group 6	
AND	Logical AND
Group 7	
OR	Logical OR
EOR	Logical EOR

Figure 1. Operator precedence

operator in a lower-numbered group always has precedence over one in a higher group. What this means in practice is that Basic scans the expression, calculates the highest operators first, and uses the results to calculate the next-highest group, and so on until the lowest rank has been calculated. Where two or more operators

First Course

of the same rank are involved in a single element of the calculation, then they are taken from left to right.

This has important consequences for any programs which use operators (in other words, most software). Once again, you need to be aware of the difference between the human and the electronic way of looking at things. Take the following example:

```
IF A AND B=A AND B THEN PRINT "True"
```

Now the human brain may well say that this will always print "True", because the expressions on either side of the equals sign will always be equal. But if we look at the table, we will see that *equal* has a higher precedence than *AND*, so Basic will in fact evaluate this as:

```
IF A AND (B=A) AND B THEN PRINT "True"
```

Now this will only print "True" if A happens to be equal to B (and both are non-zero). It is very easy to get confused by this when you are writing programs, because sometimes the result from the expression will be what you expected, and sometimes it will not. This makes debugging very difficult. To take a practical example of the kind of pitfall that might await you if you fail to take precedence into account, consider the following sentence: "if the colour is green and the size is large, or if the colour is red and the size is medium, I'll buy it". The sense of this is crystal clear, and indeed you can translate this directly into Basic as:

```
IF C=G AND S=L OR C=R AND S=M THEN B=TRUE
```

This will give exactly the same result as we would expect when speaking the sentence in English, because the equals signs are evaluated first, followed by the AND operators, and finally the OR. Now consider the following sentence: "if the colour is green or the size is large, and the height is 8 feet or the width is 6 feet,

I'll buy it". Again, this is crystal clear in English (though a little unlikely in practice!), but translating it into Basic as:

```
IF C=G OR S=L AND H=8 OR W=6 THEN B=TRUE
```

will *not* necessarily give the result we expect, because we are assuming that now the OR operators will be evaluated before the AND, which is not the case. In other words, the statement must now be written:

```
IF (C=G OR S=L) AND (H=8 OR W=6) THEN B=TRUE
```

As a further example of the need to be aware of what is going on, consider the following two lines of Basic:

```
10 IF A AND 1 THEN PRINT "True"
```

```
20 IF A AND 1=TRUE THEN PRINT "True"
```

At first sight, it might seem that both will produce the same result, since as we saw last month, the "=TRUE" part of the conditional statement is implicitly understood in line 10. But because the equals sign is explicitly stated in line 20, then operator precedence will ensure that it is evaluated *before* the AND. In other words, what we will get is:

```
IF A AND (1=TRUE) . . .
```

Line 10 above, then, will print "True" for all odd values of A (i.e. all those in which bit 0 is set), while Line 20 will never print "True", since 1 can never equal -1 and therefore the whole expression will always be false.

This example raises a further interesting pitfall of logical construction. You might think, after reading the above, that if we substitute line 20 with:

```
20 IF (A AND 1)=TRUE THEN PRINT "True"
```

then this will now always produce the same result as line 10. But this is *not* the case. The reason for this is that, if =TRUE is *implicitly* stated, as in line 10, then any non-zero result will be treated as true, as we saw last month. But if =TRUE is

explicitly stated, then only a result giving the actual value of TRUE, i.e. -1, will be treated as true.

USING EXPRESSIONS IN STATEMENTS

Last month I said that *any* expression will be evaluated by Basic to a numerical result. This might be the actual numerical value produced by a mathematical or boolean expression, or it may be a value of -1 (TRUE) or 0 (FALSE) produced by an expression which cannot otherwise produce a numerical result.

We can often use this fact to great advantage to make our programs shorter and sometimes faster (though at the expense of a little clarity). For example, suppose we are printing some text onto a sheet of labels, and the gap between labels horizontally might be 4 characters wide or it might be 7. A variable, *gap%*, will have a value of 1 for the smaller gap size, 2 for the larger, or zero for no gap. We could achieve the result we want with the following lines:

```
IF gap%=1 PRINT"      "
IF gap%=2 PRINT"          "
```

but an alternative way is as follows:

```
PRINT STRING$(-4*(gap%>0)-3*(gap%=2),
" ")
```

which actually saves 9 bytes of memory over the first version, but is much more difficult to understand. Nevertheless, when memory is tight as is often the case with programs on the BBC micro, it is a useful technique.

The way this is executed by Basic is as follows: if *gap%=1* or *gap%=2*, then *gap%>0* is true (i.e. -1). The first part of the expression thus becomes -4×-1 , or 4. Additionally, if *gap%=2*, then the second part becomes -3×-1 , or 3. So if *gap%* is 1 or 2, four spaces are printed, plus a further three if *gap%=2*.

MULTIPLE IF STATEMENTS

Finally this month I want to look at a useful technique which can often speed up the execution of your programs. If you have a single conditional statement which depends on a number of expressions all being true, you can rewrite it as a series of conditional statements instead. For example, when Basic meets the following line:

```
IF A=B AND C=D AND E=F AND G=H THEN P
RINT "True"
```

it will evaluate the entire expression before deciding whether it is true. However, if you rewrite this as:

```
IF A=B IF C=D IF E=F IF G=H THEN PRINT
"True"
```

what will happen is that each conditional statement will be considered separately. If any expression is false, then Basic will skip the rest, looking for either an ELSE statement or the end of the line. This means, in effect, that if for example $A=B$ is false, time is not wasted evaluating $C=D$, $E=F$ and $G=H$ before deciding not to print "True".

As you will see from this article and its predecessor, logic is a very tricky subject, and can be difficult to explain as well as understand. I hope I have given you enough information to avoid some of the pitfalls that might otherwise await you when writing your own programs. **B**

Use your Beeb as a Capacitance, Inductance and Frequency Meter

Software and simple circuit diagrams supplied on DFS disc

Only £9.95 for 5.25" or 3.5" disc
(please state which size)

Cheques made payable to;

**Peter Helsdon, 32 Burns Crescent,
Chelmsford, Essex CM2 0TS**

PD Software

Alan Blundell introduces the first of a new series of articles investigating the PD software scene for the BBC micro.

This is the first of what I hope will become a series of columns in BEEBUG aimed specifically at letting you know what is happening in the world of public domain software for the BBC micro and Electron. Following the article about the BBC PD library in the October 1991 issue (Vol.10 No.5), I received literally hundreds of letters and requests for catalogues and sample software. Comments clearly showed that PD software does serve a need in maintaining an interest in 8-bit computing.

If you didn't already know, then it should be clear from what you have just read that I run a PD software library. In fact, there are several public domain software libraries dedicated to the BBC and Electron. I will be giving a list of those known to me in a future article, but the emphasis will be on news about the software which is available, new developments, and reviews of particular types of software in the public domain. If you want to know more about libraries, look for their adverts and write for their information (not forgetting to enclose an SAE).

WHAT SOFTWARE IS AVAILABLE?

So, you might ask, what is available, and is it any good? Most types of software you can think of are available in the public domain, but for this month, I will mention just one area, games.

There is no shortage of 'serious' software in the public domain, but almost everyone has at least some interest in computer games, and the public domain

won't disappoint you in the variety and range of software which is available. It has to be said, though, that the quality of the games available in the public domain is very variable, ranging from simple offerings to those of commercial quality. Programmers of all levels of ability tend to try their hand at a game of one sort or another at some time, and availability of public domain games reflects this, probably more so than in any other area of PD software.

Games are also the area to which the unprincipled apply themselves most frequently, notably by disguising or passing off typed-in magazine listings (even BEEBUG's!) as public domain when this is not the case. This is, of course, unfair both to the copyright holder and to you - especially if you already have the magazine disc - as well as being illegal.

John Henson was an early contributor of games to the PD world, and his games (*Nightmare Park*, a version of *Lunar Lander*, and others) have been popular, perhaps because of this. It's fair to say though that these games can reasonably be described as of 'magazine quality' - still not bad value at about £1.50 for a disc full of them. They are better than some, but there are some very professionally produced games available as well! *Pantheon*, by Lars Osterballe and Benny Lonstrup, was written for commercial sale but somehow didn't get taken up by a software house. It is a well-produced game in two parts, a pre-game 'shoot-em-up' which leads into a large arcade adventure which is the real

meat of the game. The game itself is distributed as PD, but a map and other helpful materials are available directly from the authors for a small charge.

RISK - The Computer Game, by Adam Sandman, is a computer version of the well-known board game. It is available in four versions, ranging from an Electron-specific version which can work from tape, thro' a BBC DFS-based version, to a Master Series (ADFS) version with detailed, enhanced graphics.

Sorcerer's Domain and its sequel (*S D II*), both by A.M.Flintham are large, Repton-style games, with multiple game screens and logical problems to solve. Whilst they don't have the polished feel of Repton, each does come complete with sprite and screen editors, so that new screens can be devised.

ADVENTURE GAMES

Successful adventure games rely on the involvement of the player in the story line much more so than arcade-style games. Thus the effort of creation places more emphasis on the story line and puzzles in the game itself than into programming techniques such as optimising a sprite display routine. You might expect, then, that producing a good adventure game is a less esoteric art. From the games which are available in the public domain, this seems to be true - there are some very good ones about.

Many of them have been written with the use of commercial adventure creation systems, such as *ALPS* from Alpine Software, or *Adventurescape* from ASP. *First Contact* and *Dreadnaught* are fair examples of these, both written by P.Woodhouse (not by Philip Hawthorne, creator of *ALPS*, as was recently reported

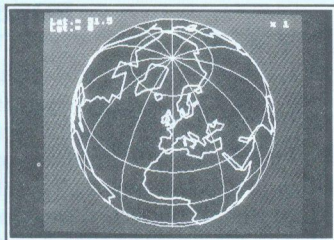
in another magazine - although that is no reflection on their quality as games). Philip Hawthorne has in fact recently released the *ALPS* adventure, *Plague Planet* as shareware. This is an ex-commercial product of excellent quality. Perhaps it represents the beginning of a trend in the wider availability of commercial programs which are now no longer actively marketed by their producers. Honesty pays with this item of shareware: paying the registration fee of £5.00 brings a further disc full of goodies!

Insomnia by Ian Hudson is a popular adventure game, displaying an off-beat sense of humour. *Insomnia* in fact represents one of the good things about the public domain - that things don't stand still! Some of the most popular PD programs have attracted enhancements from other contributors (a theme I shall return to in a future column). In this case, Duncan Lilly has produced a map, a collection of hints and tips and a complete solution to *Insomnia*, all of which are themselves available in the public domain if (or when) you get stuck.

Lastly for this month, the public domain also boasts its own adventure creation system, the *Bazzasoft Adventure Creator*. Don't let the name put you off - this is a real adventure creation system, which comes complete with two sample adventure games.

Next month, I'll look at a more serious area of PD computing, with a review of what's available in the area of programming languages. In the meantime, if you have any suggestions or questions relating to the public domain, do write to me at the usual BEEBUG address. **B**

Applications I Disc

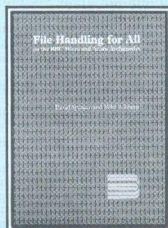


- BUSINESS GRAPHICS** - for producing graphs, charts and diagrams
- VIDEO CATALOGUER** - catalogue and print labels for your video cassettes
- PHONE BOOK** - an on-screen telephone book which can be easily edited and updated
- PERSONALISED LETTER-HEADINGS** - design a stylish logo for your letter heads
- APPOINTMENTS DIARY** - a computerised appointments diary
- MAPPING THE BRITISH ISLES** - draw a map of the British Isles at any size
- SELECTIVE BREEDING** - a superb graphical display of selective breeding of insects
- PERSONALISED ADDRESS BOOK** - on-screen address and phone book
- THE EARTH FROM SPACE** - draw a picture of the Earth as seen from any point in space
- PAGE DESIGNER** - a page-making package for Epson compatible printers
- WORLD BY NIGHT AND DAY** - a display of the world showing night and day for any time and date of the year

File Handling for All

on the BBC Micro and Acorn Archimedes

by David Spencer and Mike Williams



Computers are often used for file handling applications yet this is a subject which computer users find difficult when it comes to developing their own programs. *File Handling for All* aims to change that by providing an extensive and comprehensive introduction to the writing of file handling programs with particular reference to Basic.

File Handling for All, written by highly experienced authors and programmers David Spencer and Mike Williams, offers 144 pages of text supported by many useful program listings. It is aimed at Basic programmers, beginners and advanced users, and anybody interested in File Handling and Databases on the Beeb and the Arc. However, all the file handling concepts discussed are relevant to most computer systems, making this a suitable introduction to file handling for all.

The book starts with an introduction to the basic principles of file handling, and in the following chapters develops an in-depth look at the handling of different types of files e.g. serial files, indexed files, direct access files, and searching and sorting. A separate chapter is devoted to hierarchical and relational database design, and the book concludes with a chapter of practical advice on how best to develop file handling programs.

The topics covered by the book include:

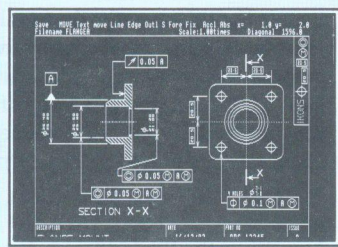
- Card Index Files, Serial Files, File Headers, Disc and Record Buffering, Using Pointers, Indexing Files, Searching Techniques, Hashing Functions, Sorting Methods, Testing and Debugging, Networking Conflicts, File System Calls

The associated disc contains complete working programs based on the routines described in the book and a copy of *File*, a full-feature Database program originally published in BEEBUG magazine.

ASTAAD

Enhanced ASTAAD CAD program for the Master, offering the following features:

- * full mouse and joystick control
- * built-in printer dump
- * speed improvement
- * STEAMS image manipulator
- * Keystrips for ASTAAD and STEAMS
- * Comprehensive user guide
- * Sample picture files



	Stock Code	Price		Stock Code	Price
ASTAAD (80 track DFS)	1407a	£ 5.95	ASTAAD (3.5" ADFS)	1408a	£ 5.95
EDIKIT (EPROM)	1451a	£ 7.75			
EDIKIT (40/80T DFS)	1450a	£ 5.75	EDIKIT (3.5" ADFS)	1452a	£ 5.75
Applications II (80 track DFS)	1411a	£ 4.00	Applications II (3.5" ADFS)	1412a	£ 4.00
Applications I Disc (40/80T DFS)	1404a	£ 4.00	Applications I Disc (3.5" ADFS)	1409a	£ 4.00
General Utilities Disc (40/80T DFS)	1405a	£ 4.00	General Utilities Disc (3.5" ADFS)	1413a	£ 4.00

Please add p&p

Board Games

SOLITAIRE - an elegant implementation of this ancient and fascinating one-player game, and a complete solution for those who are unable to find it for themselves.

ROLL OF HONOUR - Score as many points as possible by throwing the five dice in this on-screen version of 'Yahtze'.

PATIENCE - a very addictive version of one of the oldest and most popular games of Patience.

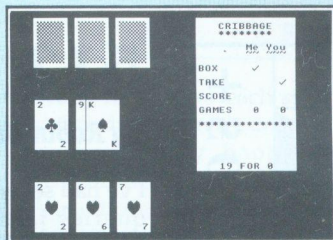
ELEVENSES - another popular version of Patience - lay down cards on the table in three by three grid and start turning them over until they add up to eleven.

CRIBBAGE - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

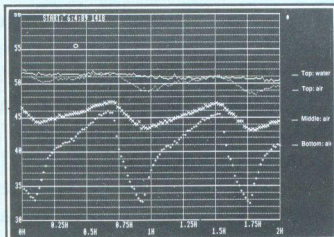
TWIDDLE - a close relative of Sam Lloyd's sliding block puzzle and Rubik's cube, where you have to move numbers round a grid to match a pattern.

CHINESE CHEQUERS - a traditional board game for two players, where the object is to move your counters, following a pattern, and occupy the opponent's field.

ACES HIGH - another addictive game of Patience, where the object is to remove the cards from the table and finish with the aces at the head of each column.



Applications II Disc



CROSSWORD EDITOR - for designing, editing and solving crosswords

MONTHLY DESK DIARY - a month-to-view calendar which can also be printed

3D LANDSCAPES - generates three dimensional landscapes

REAL TIME CLOCK - a real time digital alarm clock displayed on the screen

RUNNING FOUR TEMPERATURES - calibrates and plots up to four temperatures

JULIA SETS - fascinating extensions of the Mandelbrot set

FOREIGN LANGUAGE TESTER - foreign character definer and language tester

LABEL PROCESSOR - for designing and printing labels on Epson compatible printers

SHARE INVESTOR - assists decision making when buying and selling shares.

Arcade Games

GEORGE AND THE DRAGON - Rescue 'Hideous Hilda' from the flames of the dragon, but beware the flying arrows and the moving holes on the floor.

EBONY CASTLE - You, the leader of a secret band, have been captured and thrown in the dungeons of the infamous Ebony Castle. Can you escape back to the countryside, fighting off the deadly spiders on the way and collecting the keys necessary to unlock the coloured doors?

KNIGHT QUEST - You are a Knight on a quest to find the lost crown, hidden deep in the ruins of a weird castle inhabited by dangerous monsters and protected by a greedy guardian.

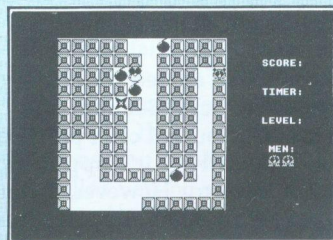
PITFALL PETE - Collect all the diamonds on the screen, but try not to trap yourself when you dislodge the many boulders on your way.

BUILDER BOB - Bob is trapped on the bottom of a building that's being demolished. Can you help him build his way out?

MINEFIELD - Find your way through this grid and try to defuse the mines before they explode, but beware the monsters which increasingly hinder your progress.

MANIC MECHANIC - Try to collect all the spanners and reach the broken-down generator, before the factory freezes up.

QUAD - You will have hours of entertainment trying to get all these different shapes to fit.



	Stock Code	Price		Stock Code	Price
Arcade Games (40/80 track DFS)	PAG1a	£ 5.95	Arcade Games (3.5" ADFS)	PAG2a	£ 5.95
Board Games (40/80 track DFS)	PBG1a	£ 5.95	Board Games (3.5" ADFS)	PBG2a	£ 5.95
File Handling for All Book	BK02b	£ 9.95			
File Handling for All Disc (40/80T DFS)	BK05a	£ 4.75	File Handling for All Disc (3.5" ADFS)	BK07a	£ 4.75
Joint Offer book and disc (40/80T DFS)	BK04b	£ 11.95	Joint Offer book and disc (3.5" ADFS)	BK06b	£ 11.95

Please add p&p. UK: £1.00 first item (50p for every additional item), Europe and Eire: £1.60 first item (80p every additional item), Elsewhere: £2.60 first item (£1.30 every additional item)

Tel. (0727) 40303

Fax. (0727) 860263



512 Forum

by Robin Burton

I've had several queries lately about the GEM collection

supplied with the 512. I'm not very familiar with these programs because, apart from a quick look now and again when I get a query, I never use them.

Once again we're indebted to David Harper who has spent some time investigating the system and its many foibles (i.e. bugs and quirks!) Before getting buried in GEM though, since it's a lengthy topic, I think mention of a couple of other points is in order.

REASONABLE REQUESTS

The first matter is one of good manners and common sense. I've not mentioned it for some time, and only a few are guilty. To the rest of you I therefore apologise for going through all this again.

If you have a 512 related problem you can write to me, c/o Beebug, and I'll help if I can. Obviously I can't promise a solution, especially if it's application specific, but I will try. However, if you'd like a reply you *must* supply return postage, an address label, as well as packing if you send discs.

It may seem a small point, but it's normal courtesy if nothing else, especially when you're using a free service. There are several hundred of you but just one of me, so help me to spend time on your problem, instead of writing your address.

Also, if you think I might be able to help please be absolutely explicit. I had a short letter a few weeks ago that said, in

effect, 'I'm having problems running application XYZ in my 512, can you help me because I'm told it should work?'. Not much to go on, is it?

The sort of information I need to make sense of a question before I can start to think about an answer includes: What sort of problems? How far does the program get when you run it? Does any of it work? What happens when it fails? Are there any error displays? Which version of DOS Plus are you using? Have you tried both 1.2 and 2.1? Did you load a second COMMAND.COM? Which version of the package do you have? How much memory does your 512 have? What steps have you tried to fix the problem? Do you use floppies or a winchester? Have you access to a PC for program installation? Do you have the original issue discs?

With the right information I'll know what you've done and won't waste time duplicating your efforts or asking more questions.

For the query above I found, after three letters and much wasted time, that the application in question had been installed for a PC VGA display and it was the installed copy which didn't work on the 512! Furthermore the user didn't have the issue discs. I declined to help further because, quite apart from the fact that it was a lost cause, I can't help anyone to pirate software. If you have a problem with a package but you don't have a legitimate copy of the software, don't ask.

If you have a legitimate copy make sure you supply all the information you can.

All of the above plus anything else you can think of that might be relevant. Send a copy of the software and the issue discs (all discs returned) and I'll do my best. Dr. Philip Draper did it all exactly right, and he now has a working copy of TimeWorks DTP.

I don't keep copies of software for which I'm not licensed, or which I don't use, so it's virtually certain that I won't have a copy of your particular package. For example I don't have a single spreadsheet or database on my discs, not even shareware versions, because I don't use them.

I do appreciate how little support and help there is for 512 users, after all I've been one myself for about six years, and that's exactly why I try to help.

Thanks, I feel better now that's over.

TRYING SOFTWARE

On a related matter I had a letter from Ron Thompson a while ago outlining his procedure for trying new software. The rules are simple, but as the mention of reloading COMMAND.COM proved a few months back, it's easy to overlook basics, so here's a quick checklist.

If you're trying a program for the very first time, with no idea whether it will work or not, you can avoid risks and ensure you've tried everything by following the points below. All this assumes of course that the 512 will read the discs. If not, go to the last paragraph of this section.

If you normally use a winchester, don't. It's best avoided until you're positive the software isn't going to do something stupid. A corrupted floppy disc is a nuisance, but a corrupted hard disc is a minor disaster, even with good back-ups!

Ideally re-boot for each step, even if the system doesn't crash, and it's a good idea to use a copy boot disc with no AUTOEXEC file too, so as to avoid any unusual or personalised setting up of your system. If any attempt fails, try the next option.

1. Try with DOS Plus 2.1 and nothing else loaded.
2. Try a second copy of COMMAND.COM.
3. Try with Problem Solver, if you have it.
4. Try a second copy of COMMAND.COM and Problem Solver.
5. Try with DOS Plus 1.2 only.
6. Try steps 1 to 3 using DOS Plus 1.2.

If you have a winchester you can also try the software from that for each of the above steps, but only so long as you're sure there won't be any ill effects and you have up to date backups.

If all that fails, you can try installing the program on a PC if you have access to one, but remember to choose CGA mono display if that's one of the installation options. You can then see if a copy of the installed version works. Ron put it quite neatly: if it still fails, give up.

GEM

Both of the main issues of the 512's DOS Plus, 1.2 and 2.1, have included a version of the GEM collection (General Environment Manager) on the issue discs.

Most 512 users will quickly move on to other applications that suit their needs better, but to the new 512 user the supplied GEM software can be extremely useful. There's both a word processor and a graphics package thrown in too, so

without spending any cash you can actually run programs.

Unfortunately, like numerous other programs in the 512's supplied software, the testing and debugging of the GEM collection leaves much to be desired.

I guess that Acorn's lack of interest in the 512 itself is directly reflected in the standard of testing applied to its supplied software. This is an unfortunate and less than encouraging introduction to the system for any new 512 user.

Like many applications, GEM needs to be configured for your own system before you use it fully and this is where you meet your first batch of problems. The program you need is GEMSETUP, but that's on disc 4, the miscellaneous disc, not on either of the two GEM discs. What's more, Acorn's so called 512 User Guide fails to even mention it (I wonder how many potential 512 users have given up at this stage over the years).

Assuming you discover the program at all, you now meet 'phase II' problems. Nine times out of ten GEMSETUP doesn't work! It seems to run OK, it even tells you it has saved your configured settings on the GEM start-up disc, so all appears to be well... until you try to run GEM, that is.

Apparently the program writes garbage to the disc more often than not, so GEM either hangs or crashes immediately you try to use it. It is true that if you repeatedly try GEMSETUP it might work eventually, but it's a slow and frustrating process since you'll crash the machine on every unsuccessful attempt. Before writing this Forum I tried it five times, with a 100% failure record (and a re-boot needed every time).

David has supplied me with information which allows the system to be set up and used much more easily, though as you might expect it means doing a bit of direct file editing.

The first impression gained from the GEM discs is usually 'what a lot of files!'. True, but you need be concerned only with one of them here. If you want to try to get GEM working properly don't forget to use copies of the issue discs (use 'copy an entire disc' in DISK.COMD). This month we'll look into what files are included and what they're for, then next month we can investigate changing the system.

GEM FILES

Like most applications, when it starts up, GEM expects to find certain files which indicate to the software how it should configure itself for the system in question. As is also common in many DOS applications, the file extensions in GEM are meaningful, each telling the system the purpose of the file.

Let's start with a look in directory GEMSYS for example. Three of the files are nothing to do with GEM system set-up, these are OUTPUT.APP, OUTPUT.RSC and FORMAT.COM. The last of these is simply an 800K disc formatter, which GEM uses, so it must remain here. However, as I've mentioned before in the Forum, you can copy this file to your other working discs and use it as a direct means of formatting 800K (only) discs instead of going through the long-winded procedures in DISK.COMD.

The other two files mentioned above are pretty obviously concerned with output of some sort. The '.APP' extension tells us that OUTPUT.APP is a program file, much like an .EXE file. To be recognised in GEM as an executable program file though, the

extension must be APP. By the way, don't think you can change any executable file extension to another type, you can't. Every executable file must be constructed in exactly the right way for its type, so the internal format of .COM, .CMD, .EXE and .APP (and any others in other systems) programs is completely different.

The RSC extension, as in OUTPUT.RSC, means resource. In simple terms an RSC file holds variables and strings which are used by the corresponding .APP program. In this case therefore, OUTPUT.APP gets some of its data and its text messages from OUTPUT.RSC.

You may also see a file called DEFAULT.OPT, which, as its name suggests, contains default options saved from OUTPUT.APP if the 'make default' command has been used. Now you know about these files you can ignore them from now on.

You'll also have noticed a number of files with a '.FNT' extension. These are all font files which contain the definitions of the various character fonts which can be used in GEM. When you get into GEM you'll find that you can change the current screen display typestyle by selecting different ones from a menu. These are all in the '.FNT' files.

Another file you will see (it's actually the one called in the GEM.BAT file) is called GEMVDI.EXE. This is clearly a normal .EXE file, and while the GEM portion of the name is obvious, the VDI bit isn't. It means 'Virtual Device Interface' (I know, "What's a virtual device?").

A virtual device is any input/output device attached to the computer. Notionally GEM can process input or output through any number of devices (screen, keyboard, printer etc.) and each

of these might and usually does have completely different physical characteristics. So that any program can interface with a range of different hardware configurations in different machines, in DOS each device type has driver, a program which 'knows' the actual physical characteristics of each real device while presenting a 'standard' appearance to an application. Of course, the range of actual devices may also vary from system to system.

In effect GEMVDI is a further refinement of this idea, so that GEM need only ever talk to GEMVDI for all input or output. GEMVDI receives data in internal (GEM) format, but with a code which tells GEMVDI which device to send the data to. GEMVDI therefore ensures that the data is presented to the device driver in the correct format for the device type. When input is received it is initially passed to GEMVDI, which buffers it if necessary before passing it to the main application again in the internal format the application can process. GEMVDI therefore allows GEM to talk to physical devices without GEM needing to know anything about them. You can now forget about GEMVDI.

The file we're really interested in is ASSIGN.SYS. Obviously from the extension type it's a system file (as opposed to a program or data file), but what about ASSIGN. This is the file which contains the data which informs GEMVDI about the range of device drivers and fonts the system can use. As you might guess, it assigns the various device drivers to a particular type of input or output in GEMVDI so GEM can use them.

Sorry, we're out of space for this month. If you want to know what happens next, read next month's 512 Forum!

The Comms Spot

The latest developments in the world of comms, by Alan Wrigley

Most of you will know by now that Micronet, the Prestel area for computer enthusiasts, has closed permanently. British Telecom broke the sad news to subscribers at the end of September, and the Net closed down on 31st October, having been in operation since 1983.

In the early days of home computing, Micronet achieved a cult status among those who could boast a modem as part of their equipment. It was always one of the liveliest areas of Prestel, being frequented largely by people who logged on to meet and chat to others, and make friends, rather than simply to access information services. Chatlines mushroomed, areas catering for most of the popular micros were started up, and multi-user games such as Shades were introduced and quickly captured the imagination of many Netters. At its peak, Micronet had 21000 subscribers, and was supported by many companies in the Acorn market, including Beebug who had a database of 450 frames on the Net.

Prestel never really took off in the mass market as many people had hoped, and for a long time Information Providers (IPs) such as Micronet appeared to be welcomed, as they drew customers to the service. However, it had been apparent for some time before the closure announcement that small-time domestic users, who had helped to establish Prestel as a going concern, had outlived their usefulness. No doubt BT management felt that such a service sat uneasily alongside the high-tech, high-profile business that they were so busy promoting.

The first nail in the coffin came two years ago, when time charges were introduced in the evenings, for which access had previously been free. Many subscribers

left the service at that point, feeling that their loyalty over the years had been betrayed. Then last year all the chatlines were suddenly closed. There were other reasons for this (which is a story in itself), but clearly the closure did no harm to BT's long-term plan to dispose of Micronet. The loss of the chatlines was a serious blow, since the concept of instant communication with other users had probably been the main selling point of the service. Thousands of people up and down the country logged on every night for a chat, and many friendships were made (there have even been marriages which resulted from meeting online).

SID

A month before the demise of Micronet, Acorn had stunned many of its more loyal customers by closing SID (its Support Information Database). SID was seen by dedicated users of Acorn micros as a source of useful information, and as a declaration of commitment from Acorn to its customers. Although SID was naturally more heavily geared towards the Archimedes market, nevertheless the BBC micro was still supported, with bulletin boards and information areas specifically for BBC users.

Acorn claims that closure of SID made sense because its objectives of fostering communication between users, could be more effectively met by global electronic mail networks such as Fidonet, to which many of the UK bulletin boards are affiliated, but many people will see it as an indication that customer support does not have a high priority at Acorn.

ALTERNATIVES

So what does the future hold for comms enthusiasts? There are still, of course, bulletin boards around the country, but

as we pointed out in our last survey (Vol.9 Nos. 1 & 2), those catering specifically for the BBC micro are now very thin on the ground.

Micronet itself is encouraging members to subscribe to CompuServe, which is an international online service not dissimilar to Prestel in some ways. For example, it has information areas for news, finance, travel, entertainment, hobbies and so on. Of most interest to ex-Micronet members will be the UK Computing Forum. New sections have been added to this specifically to cater for our "obscure" British micros such as Acorn and Sinclair machines (bear in mind that CompuServe originates from the USA and therefore will be heavily geared towards the PC user). Micronet negotiated special incentives with CompuServe for its members, such as free introductory membership and a usage credit.

Unfortunately, the cost of using CompuServe is likely to discourage many people. Connection charges are around £7.50 per hour, but in addition to this, users outside London will need to access the service via BT's PSS Dialplus network, which carries a further £4.80 per hour (daytime) or £2.70 per hour (evening and night) connection charge, on top of the actual call charges themselves. London subscribers can dial CompuServe direct, for which the charge is the same as Dialplus for peak hours, but only 18p per hour off-peak. Membership fee is modest at around £1.20 per month, but overall the cost of using the service appears quite high. On top of this, of course, many of the information services require extra charges, just as with Prestel.

For information on how to access the service, contact CompuServe at 15/16 Lower Park Row, Bristol BS99 1YN, or telephone their Customer Service department on 0800 289458 (calls free within UK).

TAO

Possibly the most thriving area on Micronet was the multi-user game *Shades*. In its latter days, it has become less a game, more a whole social structure, with the emphasis shifting from game-playing to socialising, or for the more imaginative players, role-playing. Many permanent friendships have been made between people who first crossed swords on *Shades*, and the game's creator was aware that there was a need to find *Shades* a new home to prevent the whole structure from collapsing.

A new service has been set up, called Tessier Ashpool Online (TAO), which has taken over many of the popular areas of Micronet. As well as *Shades*, Clubspot and Dialtalk will be available on TAO, and a full mailbox service too. Perhaps the most welcome feature for those who remember the "good old days" is that TAO is reinstating the chatlines for which Micronet was so popular.

The service is based in London (you can try it out on 071-275 9996 - viewdata system at all speeds up to 2400). However, access will be possible from other areas via the Mercury 5000 network, which should be available by the time you read this. Mercury 5000 is similar to BT's Dialplus mentioned earlier, and as well as connection charges and the cost of the calls, a £40 registration fee is charged by Mercury, so once again it seems as though users outside London will have to pay through the nose. TAO will also be charging a quarterly subscription fee just as Micronet did.

The future is not entirely bleak for comms enthusiasts, but the loss of Micronet is a sad blow, particularly for those who had stuck with it from the early days, and especially for those outside London, who seem to have got the worst of the deal all round. **B**

Simulation Modelling (3)

by Mike Williams

In this month's Workshop I want to look at another example of simulation modelling in order to highlight a problem which can arise, and to suggest a solution. This concerns the need to remove a future event from the calendar because of the occurrence of some other event. We will also take a brief look at the need which can arise for storing additional information about the entities (customers).

Our example, as before, will consider a situation where customers enter a queue according to some probability distribution, and are then served with a service time drawn from some other probability distribution (for convenience we shall assume as before that all distributions are normal, though this might not always fit the facts - it does simplify the coding as only one form of random sampling routine is needed).

The difference this time is that when a customer enters a queue, we will sample from a further distribution to determine that customer's queue tolerance, i.e. the length of time he is prepared to wait to be served before

he decides to leave the queue. One difficulty here, of course, is that queue tolerance is not necessarily fixed. A customer might be prepared to wait 10 minutes (say), but would hardly leave the queue if by that time he was the next to be served.

One solution might be to set the queue tolerance as a combination of random sampling and queue length. Alternatively, you might change the model such that when the customer's tolerance is exhausted, the decision as to whether he leaves the queue or not is determined by his position in the queue. That in turn means that more information will have to be stored in conjunction with each event than hitherto - in this case, the relative position of the customer in the queue.

We shall take a simple approach where queue tolerance is sampled at the time of joining the queue. This results in three types of event:

1. Arrival
2. End-service
3. Leaving

These can be represented in pseudo code format as follows:

Event type 1

```
if queue length = 0 and server idle
then
    set server busy,
    create end-service event
else
    increment queue length,
    create leaving event;
    create arrival event
```

Event type 2

```
if queue length > 0
then
    decrement queue length,
```



```
        create end-service event,  
        delete leaving event  
    else  
        set server idle
```

Event type 3

```
    decrement queue length
```

The only really new feature here is the deletion of a leaving event from the calendar when a customer is served. When a customer enters the queue, that customer has a leaving time entered into the calendar. If that customer starts being served before his leaving time occurs (his queue tolerance is not exhausted) then the leaving event becomes redundant, and must be removed from the calendar. The question is how this should be managed.

As things stand, events entered in the calendar are in no way associated with a specific individual - the calendar is just a list of future events. A leaving event can be treated in just the same way, but when a customer reaches the head of the queue and is served there is no way of determining which leaving event applied to that customer. At any time the number of leaving events in the calendar will be equal to the number of customers still waiting in the queue.

There are various solutions to this. One would be to assign each customer a unique arrival number (simply by counting the number of customers to arrive). This number would be stored in a further array in a position corresponding to that customer's leaving event time and event type in their respective arrays. Assuming customers are served in strict order, the leaving event to be deleted from the calendar would always be the one with the lowest customer number.

An alternative (and much simpler) approach would be to argue that when a leaving event is to be deleted, then deleting the earliest such event in the calendar, or the first in the calendar, or

even randomly selecting a leaving event would be just as good. Only trial and experiment would determine the extent to which this would be true. Since all the results obtained from such a simulation are subject to variation, because of random sampling, such a simple approach might have a negligible effect.

The program listed here (Listing 1) uses the former, more exact approach. This program follows very much the general outline of last month's program, including many of the previous functions and procedures. The program now includes three events as outlined above. The main additions and new features are described below.

To make the results more interesting I have added two counters to the variables used, *cust_A* which counts the number of customers arriving, and *cust_S* which counts the number of customers served. The difference between these two at the end of the simulation is the number of customers leaving before they are served. This information is displayed at the end of the simulation. A new array called *Cust()* has been added in which the sequential customer number can be stored for a leaving event.

There is one additional procedure (apart from the modified and extended procedures dealing with events), and this is *PROCdelete_time*. This procedure deletes a leaving event from the calendar when a waiting customer is to be served. This is somewhat similar to the procedure to advance the clock. Both search through the calendar for the smallest valid event, in the case of *PROCadvance_clock* the event with the lowest time, in the case of *PROCdelete_time* the event with the lowest associated customer number, that event then being removed from the calendar.

That is all there is to it. This model will repay some experimenting with, and you

Workshop - Simulation Modelling

may also find it instructive to add more recording routines. For example, it would be nice to know the maximum and average queue lengths at the end of each simulated period.

ATTRIBUTES

The basic principles of simulation modelling can also be extended without too much difficulty to more complex models. We have looked previously at multiple servers, and in effect multiple queues. This requires the greater use of arrays (one queue for each server, for example). It is also quite feasible to contemplate multi-layer models. By this I mean a situation where a 'customer' has to be served (maybe optionally) by more than one server. As an example, I am reminded of motorway services where hot food, cold food, drinks etc. are served from different counters at each of which a queue may form, before reaching a final queue (or queues) for payment. Where multiple servers are involved then there has to be a defined mechanism by which a customer 'chooses' a server - this may be random, or it may be related to queue length (or any other factors). We look at a more complex example of this type in the next and final part in this series.

One area we haven't dealt with in practical detail is that of attributes. For example, if we are modelling the activities of a world-wide shipping fleet, each ship at our disposal will no doubt be of a different size (tonnage), and there may be more than one type of vessel (oil tanker, cargo vessel, passenger liner, etc.). This data will have to be stored for each ship. A cargo (people, goods, or whatever) will then have to be matched with the first available ship of the appropriate characteristics. Additional information of this kind is referred to as the *attributes* of each 'entity' ('customer').

In some circumstances, order can be important, and we may need to identify

the 'customers' and their position in a queue. At this stage it usually becomes necessary to give entities greater reality (and an identity). Each entity will have associated with it an array containing its attributes. Entities are often treated in mathematical 'set' terms; the set of ships in port, the set of ships at sea, etc. In some cases, order within a set is also important, in other cases this is not so. The upshot of this is that many simulations need to perform tasks such as adding an entity to a set (maybe specifically at the head or the tail), removing an entity from a set, or searching a set for an entity with a given attribute.

For all of these reasons, a number of simulation languages have been developed for large scale modelling applications, languages like Simula and Simscript. However, BBC Basic (or any other high level language) can be used as a perfectly viable alternative, as we have seen here, though it does pay to create a set of basic building blocks and a basic program structure, as we have done, before attempting to simulate a particular system.

```
10 REM Program Event2
20 REM Version B1.0
30 REM Author Mike Williams
40 REM BEEBUG December 1991
50 REM Program subject to copyright
60 :
100 MODEL31:ON ERROR GOTO 280
110 PROCtitle
120 PROCinput
130 PROCinit
140 PROCset_time(clock+FNnormal(m1,sd1),1,0)
150 REPEAT
160 PROCadvance_clock
170 PRINT"Clock = ";clock;TAB(20);"queue = ";q
180 IF event_type=1 THEN cust_A=cust_A+1:PROCevent1
190 IF event_type=2 THEN PROCevent2
200 IF event_type=3 THEN PROCevent3
210 UNTIL clock>clock_end AND q=0
220 PRINT"Clock = ";clock;TAB(20);"que
```

```

ue = ";q
230 PRINT "Customers arriving: " cust_A
240 PRINT "Customers served: " cust_S
250 PRINT "Customers leaving: " (cust_
A-cust_S)
260 END
270 :
280 IF ERR<>17 THEN REPORT:PRINT " at 1
ine ";ERL
290 END
300 :
1000 DEF PROCtitle
1010 PRINTTAB(10,1)"S I M U L A T I O N
M O D E L"
1020 ENDPROC
1030 :
1040 DEF PROCinit
1050 clock=0:q=0:server_idle=TRUE
1060 cust_A=0:cust_S=0
1070 TN%=20:DIM Cal(TN%),Type(TN%),Cust
(TN%)
1080 ENDPROC
1090 :
1100 DEF PROCinput
1110 INPUTTAB(5,4)"Duration: " clock_en
d
1120 PRINT"Arrival times"
1130 INPUTTAB(5)"Mean: " m1
1140 INPUTTAB(5)"Standard deviation: "
sd1
1150 PRINT"Service times"
1160 INPUTTAB(5)"Mean: " m2
1170 INPUTTAB(5)"Standard deviation: "
sd2
1180 PRINT"Tolerance times"
1190 INPUTTAB(5)"Mean: " m3
1200 INPUTTAB(5)"Standard deviation: "
sd3
1210 ENDPROC
1220 :
1230 DEF FNnormal(M,SD)
1240 LOCAL I%,X:X=0
1250 FOR I%=1 TO 12
1260 X=X+RND(1)
1270 NEXT
1280 =INT(M+SGN(RND(1)-0.5)*(X-6)*SD+0.
5)
1290 :
1300 DEF PROCset_time(time,type,att)
1310 LOCAL I%:I%=-1
1320 REPEAT:I%=I%+1:UNTIL Cal(I%)=0

```

```

1330 Cal(I%)=time:Type(I%)=type:Cust(I%
)=att
1340 ENDPROC
1350 :
1360 DEF PROCadvance_clock
1370 LOCAL I%,Ltime,LI%:I%=-1
1380 REPEAT:I%=I%+1:UNTIL Cal(I%)>0
1390 Ltime=Cal(I%):LI%=I%:I%=I%+1
1400 REPEAT
1410 IF Cal(I%)>0 AND Cal(I%)<Ltime THE
N Ltime=Cal(I%):LI%=I%
1420 I%=I%+1
1430 UNTIL I%>TN%
1440 clock=Ltime:event_type=Type(LI%):a
tt=Cust(LI%)
1450 Cal(LI%)=0:Type(LI%)=0:Cust(LI%)=0
1460 ENDPROC
1470 :
1480 DEF PROCdelete_time(type)
1490 LOCAL I%,Latt,LI%:I%=-1
1500 REPEAT:I%=I%+1:UNTIL Type(I%)=type
AND Cal(I%)>0 AND Cust(I%)>0
1510 Latt=Cust(I%):LI%=I%:I%=I%+1
1520 REPEAT
1530 IF Type(I%)=type AND Cust(I%)<Latt
AND Cal(I%)>0 THEN Latt=Cust(I%):LI%=I%
1540 I%=I%+1
1550 UNTIL I%>TN%
1560 Cal(LI%)=0:Type(LI%)=0:Cust(LI%)=0
1570 ENDPROC
1580 :
1590 DEF PROCevent1
1600 IF q=0 AND server_idle THEN server
_idle=FALSE:PROCset_time(clock+FNnormal(
m2,sd2),2,0):cust_S=cust_S+1 ELSE q=q+1:
PROCset_time(clock+FNnormal(m3,sd3),3,cu
st_A)
1610 IF clock<=clock_end PROCset_time(c
lock+FNnormal(m1,sd1),1,0)
1620 ENDPROC
1630 :
1640 DEF PROCevent2
1650 IF q>0 THEN q=q-1:PROCset_time(clo
ck+FNnormal(m2,sd2),2,0):cust_S=cust_S+1
:PROCdelete_time(3) ELSE server_idle=TRU
E
1660 ENDPROC
1670 :
1680 DEF PROCevent3
1690 q=q-1
1700 ENDPROC

```



BEEBUG Function/Procedure Library (7)

by R.W.Smith

The following contribution to the Function/Procedure Library by R.W.Smith (in three parts) is intended for the simplification of non-graphical applications. It should be noted that in this context most of the routines require mode 7 or mode 135. In the main they will operate on a BBC B or the Master. Note, however, that the routines for access to the sideways RAM (part three) are only applicable to a Master or a B with extension board.

The routines are all of a general purpose nature, and as such can be fitted easily

into any program. To avoid the use of LOCAL variables, the routines use the characters # and _ for use in variable names. Note that these are the only two characters which can be used in variables other than the alphabetic and numeric characters. These characters should not therefore be used outside of the routines to avoid conflict.

This month's routines, described in our standard format, are concerned with screen formats and input.

THE FUNCTION/PROCEDURE LIBRARY (PART 7)

Routine 1: Screen Message

Type: PROCEDURE
Syntax: PROCmessage(string\$)
Purpose: To print a text message at any position on the screen.
Parameter: A string in which the first two characters are the row and the third and fourth characters are the line, at which the rest of the string is to be displayed on the screen.
Notes: The row in the first two characters and the line number in the third and fourth characters must be in numeric form (see example).
Related: Requires PROCpos (Routine No. 2)

Example:

```
10 PROCmessage("0410This is at column  
four and line 10")
```

Routine 2: Cursor Positioning

Type: PROCEDURE
Syntax: PROCpos(string\$)

Purpose: Position the cursor from text characters.
Parameter: A string with the first two characters giving the row number and the third and fourth characters giving the line number.
Notes: The string can be of any length as only the first four characters are considered.
Related: Used in Routine 1 and in Routine 8

Example:

```
10 PROCpos("0410This can be rubbish")
```

Routine 3.: Program Pause

Type: PROCEDURE
Syntax: PROCpause(N)
Purpose: To cause a pause in a program.
Parameter: N is a value equal to the number of seconds the pause is to operate for.
Notes: The pause can be terminated by pressing any key.

Related: None

Example:
10 PROCpause(10)

Routine 4.: Data Input Routine

Type: FUNCTION
Syntax: FNnumin(A,X,Y)
 To input a real number
FNdatein(X,Y)
 To input a date
FNdecin(A,X,Y)
 To input a sterling
 amount to store as pence
FNintin(A,X,Y)
 To input an integer
FNalphain(A,X,Y)
 To input a string of
 variable length up to a
 maximum length.
FNalphast(A,X,Y)
 To input a string of
 variable length with
 output of a fixed length.

Purpose: To provide a general
purpose input routine.

Parameter: A is the maximum number
of characters which can be
entered. X is the screen row
number at which input will
be shown. Y is the screen
line number.

Notes: The return variable used for
the function must comply
with the function called, i.e.
a real variable for FNnumin,
an integer variable for
FNdatein, FNdecin, FNintin
and a string variable for
FNalphain and FNalphast.
Upon entry to the function
the buffer is cleared, so no
key depressions are
accepted until the screen
call. This call is a line of dots
equalling the maximum
number of characters to be
accepted.

The date is entered as six
figures for day, month, year
and when entered these are
displayed as DD/MM/YY.
No check on the validity of
the entries is made.

A sterling amount entered
under Function FNdecin,
does not require a space or
decimal point between the
pounds and pence. This is
entered by the routine. The
Function FNalphast puts
spaces on the end of the
string entered, if required, to
output a string of fixed
length.

Negative numeric input is
defined by the first entry
being the negative sign - .
A variable CTL is used to
allow a NULL or cancelled
input.

Pressing the TAB Key as the
first entry causes the function
to exit with CTL set as TRUE.
Use of an unacceptable
character causes a beep and
the entry is ignored.

Related: Needs PROCcl, FNpsr

Example:

```
10 N=FNnumin(8,4,10)
20 DATE%=FNdatein(4,11)
30 VALUE%=FNdecin(6,4,12)
40 INT%=FNintin(8,4,13)
50 NAME%=FNalphain(20,4,14)
60 TOWN%=FNalphast(20,4,15)
```

Routine 5: Add Spaces to End of String.

Type: FUNCTION
Syntax: FNpsr(S\$,N%)
Purpose: To ensure that a string is of a
fixed length for use in
tabulated displays and
printing. Also useful for
filing to maintain record
compatibility.

BEEBUG Function/Procedure Library

Parameters: S\$ is the string to be used
N% is the length at which the string is to be returned.

Notes: The routine curtails strings exceeding the length stated in addition to extending the length where required.

Related: Used in Routine 4

Example:

```
10 X$=FNpsr("This is the String",20)
```

Routine 6: Prepare Space on Screen.

Type: PROCEDURE
Syntax: PROCcl(X%,Y%,A%,S\$)
Purpose: To clear a line on the screen with spaces or to draw a line of characters such as a dotted line.

Parameters: X% is the screen row number
Y% is the screen line number
A% is the number of characters
S\$ is the character to be shown

Notes: Can be used to erase a line or to emphasise a line.

Related: Used in Routine 4

Example:

```
10 PROCcl(4,10,20,"."):
REM Dotted line of 20 Fullstops
```

Routine 7: Yes/No Screen Prompt

Type: FUNCTION
Syntax: FNyn(S\$,X%,Y%)
Purpose: To allow the operator to confirm the accuracy of entries made.

Parameters: S\$ is a message posing the question to the operator to which will be added " : Y/N"
X% is the row at which the message is to be displayed.
Y% is the line number.

Notes: The function returns TRUE if Y or y is pressed and False if N or n.

Related: None

Example:

```
10 IF FNyn("Are Your Entries Correct",
1,24) THEN PROCcarryon ELSE PROCgoback
```

Routine 8: Double Sized Screen Print

Type: PROCEDURE
Syntax: PROCtitle(S\$)
Purpose: To display program name or title on screen in double Sized Characters.

Parameters: S\$ is a string in which the first two characters are the row, and the next two characters are the line at which the characters in the remainder of the string are to be displayed.

Notes: Only applicable in mode 7 or 135

Related : None

Example:

```
PROCtitle("0310PROGRAM * TELSTAR **")
```

Routine 9: Double Sized Line Print

Type: PROCEDURE
Syntax: PROCi
Purpose: To print each separate line of double height print.

Parameters: None

Notes: Only applicable in modes 7 and 135.

Related: Integral to Routine 8

Routine 10: Shift up Part of Screen

Type: PROCEDURE
Syntax: PROCupscreen
Purpose: To maintain messages on the screen at the top two lines and the bottom two lines, when scrolling.

Parameters: None.

Notes: The procedure returns with a full screen set.

Related: None

Example:

```
10 PROCupscreen
```

Routine 11: Clear Part of Screen

Type: PROCEDURE
 Syntax: PROCcIs
 Purpose: To allow part of the screen to be cleared and maintaining contents of lines 1, 2, 23, and 24 on display.

Parameters: None
 Notes: The procedure returns with the full screen set.
 Related: None

Example:
 10 PROCcIs

```

28000 REM SCREEN MESSAGE
28010 :
28020 DEF PROCmessage(_$)
28030 PROCpos(_$)
28040 PRINTMID$_($,5);
28050 ENDPROC
28060 :
28070 REM Cursor Positioning.
28080 :
28090 DEF PROCpos(_$)
28100 PRINTTAB(VAL(MID$_($,1,2)),VAL(MID
$_($,3,2)));
28110 ENDPROC
28120 :
28130 REM Program Pause
28140 :
28150 DEF PROCpause(_%)
28160 _%=INKEY(_%*100)
28170 ENDPROC
28180 :
28190 REM Data Input Routine
28200 :
28210 REM Entry for Number
28220 DEF FNnumin(_%,_%,``%)
28230 =FNn(3,_%+1)
28240 :
28250 REM Entry for Date
28260 :
28270 DEF FNdatein(_%,``%)
28280 =FNn(2,6)
28290 :
28300 REM Entry for Value stored as Inte
ger
28310 :
28320 DEF FNdecin(_%,_%,``%)
28330 =FNn(1,_%)
28340 :
28350 REM Entry for Integer
28360 :
28370 DEF FNintin(_%,_%,``%)
28380 =FNn(0,_%)
28390 :
28400 REM Entry for Text of Variable len
    
```

```

gth
28410 :
28420 DEF FNalphan(_%,_%,``%)
28430 =FNn(-1,_%)
28440 :
28450 REM Entry for Text of fixed length
28460 :
28470 DEF FNalplast(_%,_%,``%)
28480 =FNpsr(FNn(-2,_%),_%)
28490 :
28500 REM Main Input Routine
28510 :
28520 DEF FNn(_`%,_%)
28530 PROCc1(_`%,_`%,_`%,".")
28540 _$="":CTL=0
28550 *FX15,1
28560 REPEAT:`$=GETS:`%=ASC(`$)
28570 IF`%=127 AND _$="" THEN VDU7:`%=0
28580 IF`%=9 AND _$="" THEN CTL=-1:`%=13
28590 IF`%=127 _$=LEFT$_($,LEN$_-1):PRIN
T`$;:`%=0
28600 IF`%=45 AND LEN($_)<1 AND _`% > -1
THEN _`%=_%+1
28610 IF(`%<0 AND `%<13) AND(LEN$_=$_
OR `%<32 OR `%>126) THEN VDU 7:`%=0
28620 IF(`%>0 AND `%<13) ANDNOT(_`%<0
OR `%=32 OR `%=46 OR (`%>47 AND `%<58))
THEN VDU7:`%=0
28630 IF`%>13 PRINT`$; : _$=_$+`$
28640 UNTIL`%=13
28650 `$_=$_:IF`%<0 THEN _=$_
28660 IF LEN _$ < 1 THEN 28690
28670 IF _`%=1 _$=MID$_($,1,LEN($_)-2)+
."+RIGHT$_($,2)
28680 IF _`%=2 _$=RIGHT$_(" "+$_,6):_
$_=MID$_($,1,2)+"/"+MID$_($,3,2)+"/"+RIGH
T$_($,2)
28690 PRINTTAB(_`%,``%);$_;
28700 =VAL(`$)
28710 :
28720 REM Add spaces to Text up to fixe
d length
28730 :
    
```

BEEBUG Function/Procedure Library

```
28740 DEF FNpsr(_$,_%)  
28750 IF LEN(_$)>=% THEN=LEFT$_($,_% ) E  
LSE=_$+STRING$_($-LEN$_$), " "  
28760 :  
28770 REM Prepare Space on Screen  
28780 :  
28790 DEF PROCcl(_$,`%,_%,_$)  
28800 PRINTTAB(_$,`%`);STRING$_($,_%);T  
AB(_$,`%`);  
28810 ENDPROC  
28820 :  
28830 REM Yes/No Screen Query  
28840 :  
28850 DEF FNyn(_$,_%`,`%`):PRINTTAB(_  
`,`%`,`%`);_`,`%` :Y/N";  
28860 REPEAT:_`%=INSTR("Nyn",GET$)  
28870 UNTIL _`%` : =(_`%AND1)-1  
28880 :  
28890 REM Double Sized Screen Print  
28900 :  
28910 DEF PROCtitle(_$)  
28920 PROCpos(_$):PROCl
```

```
28930 PROCpos(_$):VDU10:PROCl  
28940 ENDPROC  
28950 :  
28960 REM Double Sized Line Print  
28970 :  
28980 DEF PROC1  
28990 PRINTCHR$141;MID$_($,5);  
29000 ENDPROC  
29010 :  
29020 REM Shift Up Part Screen  
29030 :  
29040 DEF PROCupscreen  
29050 VDU28,0,22,39,3:PRINTTAB(0,18)  
29060 VDU10,28,0,24,39,0:ENDPROC  
29070 :  
29080 REM Clear Part Screen  
29090 :  
29100 DEFPROCcls  
29110 VDU28,0,22,39,3,12,28,0,24,39,0  
29120 ENDPROC  
29130 :
```

B

Points Arising....Points Arising....Points Arising....Points Arising....

ADFS DESKTOP ENHANCEMENTS (Vol.10 No.5)

Unfortunately the following line was missed out of the listing given in the article, (although the magazine disc version was correct):

```
1630 DEF PROCnew:X%=&70:Y%=0:A%=0:IF  
(USR&FFDA AND 255)<>8 CLS:PRINT"File  
ng system is not ADFS":PROCkey ELSE  
OSCLI"MOUNT":PROCClear:S%=0
```

MASTER MOS DFS BUGS (Vol.10 No.5)

Line 100 was printed incorrectly, causing the new DFS image to crash when used with our magazine disc. The correct line is:

```
100 P%=&51F4
```

PC DISC FORMATTER (Vol.10 No.4)

As it stands, this program will not format write protected discs, but it appears to do so. Adding the following line checks for write protection:

```
1552 IF (?cmd AND &40) PRINT "Write Pr  
otected Disc!":VDU7:PROCG
```

DOUBLE SIZE CHARACTERS IN MODE 0 (Vol.10 No.5)

The program *Dblsiz2* in the above article wouldn't work on a BBC B because it included some 65C02 assembler mnemonics. Adding the following lines (which replace lines in the original) enables the program to run on a BBC model B as well.

```
1050 DIM big% 500  
1260 CLC:LDX #11:ADC #1:STA vdu%,X  
1270 LDX #21:ADC #1:STA vdu%,X  
1280 LDX #31:ADC #1:STA vdu%,X  
1300 INX:ADC #1:STA vdu%,X  
1350 INX:ADC #1:STA vdu%,X  
1400 LDA #0:STA char%  
1540 LDY #0:STY ptr%:LDY #1  
1600 LDY #0:STY ptr%:LDY #11  
1910 TXA:PHA:LDX mult%:DEX:LDA temp%  
1940 PLA:TAX:CLC:LDA store%  
2000 AND #8:STA temp%:LDA #0:STA store%
```

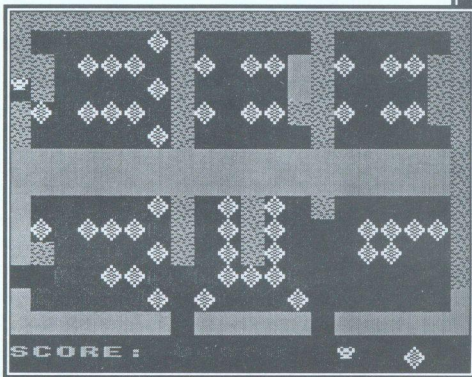
B

Pitfall Pete

In this issue we present one of our most popular games ever, by Jonathan Temple from Vol.5 No.7, to bring some Christmas cheer to your BBC.

Pitfall Pete is a 'Repton' style game in which you must collect all the diamonds on each screen, being careful not to trap yourself or the diamonds when you dislodge the many boulders.

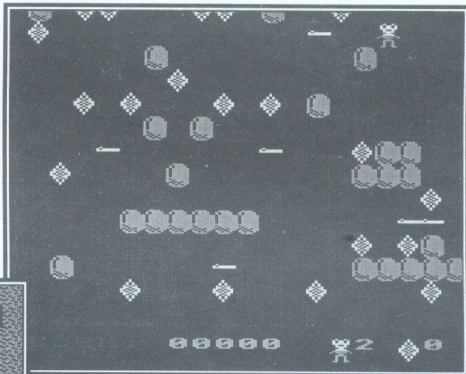
As such, the game is strong on strategy, and the screens (five of them in the listing and ten extra on the disc) require a high degree of planning and forethought before you can complete them. The game also features an easy-to-use screen designer, allowing you to try and conquer screens of your own design.



When run, the program presents an initial menu. Pressing 'P' will allow you to play the game. The keys to control Pete are 'Z', 'X', '*' and '?' for left, right, up and down. In addition you can pause in playing the game by pressing 'P', with 'C' to continue. Sound effects can be turned on and off with 'S' and 'Q'. Pressing 'R' will allow you to restart a screen, although you do lose a life. Pressing Escape will terminate the game and return you to the menu.

Once a screen has been completed, a three digit code is revealed for the next

screen. From then on you can bypass earlier screens by selecting option 'C' from the initial menu and entering the relevant code.



EDITING AND DESIGNING SCREENS

Choosing option 'E' from the menu will allow you to edit the current screen. Up to twenty screens can be designed and edited, although you will need to know the pass code for each, and select it with option 'C'. When editing, the cursor keys will move the cursor across the playing area. Features are inserted using the function keys - f0 will insert a blank, f1 earth, f2 a wall, f3 a boulder, f4 a diamond, and f5 a trapdoor. When complete, press 'E' or use Escape.

Pete usually starts a screen from the top left-hand corner, but this can be changed by using f6. The default colour scheme can also be changed using 'D' (diamonds), 'R' (rocks), or 'E' (earth) to select the feature, and then pressing numbers 1 to 7 for the new colour. The feature currently being changed is displayed beneath the playing area. An additional feature is that Ctrl-f0 can be used to clear the screen to 'earth'.

Pitfall Pete

To save an edited screen, use the 'S' option. You will be asked for a file name, and the first and last screen to save, with all nominated screens being saved together in one file. Loading previously saved screens is accomplished with the 'L' option.

On this month's magazine disc are the program from the magazine as well as ten extra screens for loading into the game. To use these extra screens, use the 'L' option and enter the filename 'PitScrn'. Then answer '1' to the 'Load From Screen' question, and the extra screens can then be played straight away.

```
10 REM Program Pitfall Pete
20 REM Version B1.0
30 REM Author J.Temple
40 REM BEEBUG December 1991
50 REM Program subject to copyright
60 :
70 ON ERROR GOTO 210
80 MODE 5:HIMEM=&4100
90 PROCinit
100 PROCsetup
110 REPEAT
120 C%=FNmenu
130 IF C%=1 PROCfile(0)
140 IF C%=2 PROCedit
150 IF C%=3 PROCfile(1)
160 IF C%=4 PROCgame
170 IF C%=5 PROCcode
180 UNTIL C%=6
190 MODE7:GOTO 240
200 :
210 IF ERR>127 CLS:PRINT'"FILING ERRO
R:"':REPORT:G%=INKEY(200):GOTO 110
220 MODE 7:PRINT'"':REPORT
230 PRINT" at line ";ERL
240 *FX 4
250 *FX 229
260 END
270 :
280 DEFPROCgame
290 Z%=3:S%=0:P%=SS:SF=0
300 REPEAT
310 PROCscreen
320 REPEAT
```

```
330 PROCkeys:PROCman
340 FOR N=1 TO 150:NEXT
350 UNTIL E%
360 IF E%=1 Z%=Z%-1:SOUND 16,2,100,2:T
IME=0:REPEAT UNTIL TIME>100
370 IF E%=2 PROCnext
380 UNTIL Z%=0 OR E%=3
390 VDU 28,4,13,15,11,12,26
400 PRINTTAB(5,12);"GAME OVER";
410 TIME=0:REPEAT UNTIL TIME>200
420 ENDPROC
430 :
440 DEFPROCkeys
450 IF INKEY-17 THEN *FX210,1
460 IF INKEY-82 THEN *FX210
470 IF INKEY-56 REPEAT UNTIL INKEY-83
OR INKEY-113
480 IF INKEY-52 E%=1
490 IF INKEY-113 E%=3
500 ENDPROC
510 :
520 DEFPROCman
530 V%=X%:W%=Y%:L%=0
540 IF INKEY-98 IF X%>0 PROCpush(-1):G
OTO580
550 IF INKEY-67 IF X%<19 PROCpush(1):G
OTO580
560 IF INKEY-73 IF Y%>0 VDU31,X%,Y%-2:
CALL R%:L%=C%:IFC%<226 OR C%=228 Y%=Y%-2
:GOTO580
570 IF INKEY-105 IF Y%<26 VDU31,X%,Y%+
2:CALL R%:L%=C%:IFC%<226 OR C%=228 Y%=Y%
+2
580 IF X%<>V% OR Y%<>W% PRINTTAB(V%,W%
);F$(0);TAB(X%,Y%);F$(6):IFL%=225 SOUND
16,-5,5,1
590 IF D% PROCfall
600 IF L%=228 SOUND17,1,100,1:Q%=Q%+1:
PROCscore(10):IF Q%=D E%=2
610 VDU31,X%,Y%-2:CALL R%
620 IF C%=226 IF Y%>1 PROCfall:D%=TRUE
:J%=X%:K%=Y%
630 ENDPROC
640 :
650 DEFPROCfall
660 IF D%=0 ENDPROC
670 IF J%<>X% OR K%<>Y% D%=0:PROCdrop(
J%,K%-2)
680 ENDPROC
```

```

690 :
700 DEFPROCpush(D%)
710 U%=X%+D%:VDU31,U%,Y%:CALL R%:L%=C%
720 IF C%<226 OR C%=228 X%=U%:ENDPROC
730 IF C%=226 IF U%>0 IF U%<19 VDU 31,
U%+D%,Y%:CALL R%:IF C%<225 SOUND16,-12,6
,1:X%=U%:PRINTTAB(U%,Y%);F$(0);TAB(U%+D%
,Y%);F$(3);:PROCdrop(U%+D%,Y%)
740 ENDPROC
750 :
760 DEFPROCdrop(A%,B%)
770 VDU 31,A%,B%:CALL R%
780 IF C%<>226 OR B%+2=28 ENDPROC
790 VDU 31,A%,B%+2:CALL R%
800 IF C%>224 IF C%<231 ENDPROC
810 IFC%=232 E%=1:ENDPROC
820 VDU 31,A%,B%
830 N%=0:REPEAT N%=N%+1
840 IF VPOS=0 C%=0 ELSE VDU 11,11:CALL
R%
850 UNTIL C%<>226
860 REPEAT
870 PRINTTAB(A%,B%-(N%-1)*2) F$(0)
880 PRINTTAB(A%,B%+2) F$(3)
890 FOR N=0 TO N% DIV3
900 SOUND 0,2,150,1:NEXT
910 B%=B%+2:VDU 31,A%,B%+2:CALL R%
920 UNTIL B%=26 OR C%=232 OR (C%>224 A
ND C%<231)
930 IFC%=232 E%=1:ENDPROC
940 IFC%=228 PROCslide(A%,B%)
950 ENDPROC
960 :
970 DEFPROCslide(A%,B%)
980 LOCAL H%
990 IF A%>0 VDU31,A%-1,B%:CALL R%:IFC%
<225 H%=-1
1000 IF A%<19 VDU31,A%+1,B%:CALL R%:IFC
%<225 H%=1
1010 IF H% PRINTTAB(A%,B%);F$(0);TAB(A%
+H%,B%);F$(3);:PROCdrop(A%+H%,B%):PROCdr
op(A%,B%-2)
1020 ENDPROC
1030 :
1040 DEFPROCscreen
1050 VDU 23;1,0;0;0;0;
1060 S=P%:PROCmine
1070 COLOUR 2:PRINTTAB(0,29);"SCORE:";T
AB(14,29);F$(6);TAB(17,29);F$(4);

```

```

1080 Q%=0:D%=0:E%=0:PROCscore(0)
1090 VDU 23;1,40;0;0;0;
1100 ENDPROC
1110 :
1120 DEFPROCscore(N%)
1130 S%=S%+N%:COLOUR 3
1140 PRINTTAB(7,29);LEFT$( "00000",5-LEN
(STR$(S%)))+STR$(S%);TAB(15,29);Z%-1;TAB
(18,29);Q%;
1150 ENDPROC
1160 :
1170 DEFPROCnext
1180 SF=SF+1:P%=P%+1:IFP%=21 P%=1
1190 VDU 28,4,15,15,11,12,26,17,2
1200 PRINTTAB(5,12);"NEXT CODE";
1210 COLOUR 1:PRINTTAB(8,14);P%(P%);
1220 FOR N%=1 TO (300+SF*100) DIV25
1230 PROCscore(25):SOUND 0,-10,5,1
1240 FOR N=1 TO 50:NEXT,
1250 RESTORE 1320:N=81:*FX 15
1260 FOR L%=1 TO 10:READ A,D:N=N+A
1270 SOUND 1,-10,N,D:SOUND 2,-5,N+48,D
1280 NEXT:REPEAT UNTIL ADVAL(-7)=15
1290 TIME=0:REPEAT UNTIL TIME>200
1300 ENDPROC
1310 :
1320 DATA 0,4,8,4,8,4,4,4,8,8,-12,8,4,8
,-12,8,8,8,-16,8
1330 :
1340 DEFPROCfile(F%)
1350 B%=0:REPEAT VDU12,17,1,B%
1360 PRINT""ENTER FILENAME:""
1370 COLOUR 3:INPUT ""F$:B%=7
1380 UNTIL F$>""
1390 C$="LOAD":IF F% C$="SAVE"
1400 X%=FNinput(C$+" FROM SCREEN")
1410 IF Y% Y%=FNinput("TO SCREEN"):IF Y
%<X% Y%=X%
1420 L$=STR$~(FNaddr(X%))
1430 C$=C$+" "+F$+" "+L$
1440 IF F% C$=C$+" "+STR$~(FNaddr(Y%)+2
89)
1450 $&700=C$:X%=0:Y%=&7:CALL &FFF7
1460 ENDPROC
1470 :
1480 DEFPROCcode
1490 CLS:PRINT""ENTER A CODE:""
1500 INPUT">--> "T%
1510 N%=0:REPEAT N%=N%+1

```

```

1520 UNTIL P%(N%)=T% OR N%=20
1530 COLOUR 2
1540 IF P%(N%)=T% SS=N%:PRINT'"SCREEN
";SS;" CHOSEN":GOTO1560
1550 PRINTTAB(4)"ILLEGAL CODE"
1560 TIME=0:REPEAT UNTIL TIME>200
1570 ENDPROC
1580 :
1590 DEFPROCclear(S)
1600 A%=FNADDR(S)
1610 FOR N%=0 TO 276 STEP 4
1620 A%!N%=&1010101:NEXT:A%?285=1
1630 ENDPROC
1640 :
1650 DEFPROCedit
1660 CLS:C%=282:S=SS:*FX4,1
1670 REPEAT PROCmine
1680 VDU 23;10,6;0;0;0;
1690 A%=FNADDR(S):PROCCOLOURS
1700 D%=0:VDU 31,10,15
1710 REPEAT G%=GET
1720 IF G%=139 IFVPOS>1 VDU11,11
1730 IF G%=138 IFVPOS<27 VDU10,10
1740 IF G%=136 IF POS>0 VDU8
1750 IF G%=137 IF POS<19 VDU9
1760 IF G%=144 PROCclear(S):D%=1
1770 IF G%=27 D%=2
1780 N%=(G% OR16)-48
1790 IF_N%>0 IF N%<8 A%?C%=N%:PROCCOLOURS
1800 G%=G% AND 223
1810 IF G%=70 D%=2
1820 IF G%=82 C%=282:PROCCOLOURS
1830 IF G%>67 AND G%<70 C%=212+G%:PROCCOLOURS
1840 IF G%>1 IF G%<8 VDU11:PRINT F$(G%-2);CHR$(8)::A%(POS-(VPOS-1)*10)=G%-2
1850 IF G%=8 X=POS:Y=VPOS-1:VDU 31,A%?283,A%?284:PRINT F$(A%?285):A%(A%?283+(A%?284)*10)=A%?285:A%?285=A%?(X+Y*10):A%(X+Y*10)=1:A%?283=X:A%?284=Y:VDU 31,X,Y:PRINT F$(6);CHR$(8);
1860 UNTIL D%:UNTIL D%=2:*FX 4
1870 VDU 23;10,32;0;0;0;
1880 ENDPROC
1890 :
1900 DEFPROCcolours
1910 X=POS:Y=VPOS
1920 VDU 19,1,A%?280;0;19,2,A%?281;0;19

```

```

,3,A%?282;0;
1930 COLOUR C%-279
1940 PRINTTAB(0,29);"CHANGING ";MID$("DIAMONDSEARTh ROCKS ",(C%-280)*8+1,8);TAB(X,Y);
1950 ENDPROC
1960 :
1970 DEFPROCmine
1980 VDU 23;10,32;0;0;0;
1990 A%=FNADDR(S):D=0
2000 VDU4,12,19,1,A%?280;0;19,2,A%?281;0;19,3,A%?282;0;
2010 FOR V%=0 TO 13
2020 FOR W%=0 TO 19:IF?A%=4 D=D+1
2030 PRINT F$(?A%);:VDU 11:A%=A%+1
2040 NEXT:PRINT:NEXT
2050 X%=A%?3:Y%=A%?4
2060 PRINTTAB(X%,Y%) F$(6)
2070 ENDPROC
2080 :
2090 DEFFNADDR(S)=M%+S*290-290
2100 :
2110 DEFFNinput(P$)
2120 COLOUR 1:PRINT'"P$+";
2130 Y%=VPOS:B%=0:COLOUR 3:REPEAT
2140 PRINTTAB(LEN(P$)+2,Y%);SPC(20);TAB(LEN(P$)+2,Y%);
2150 VDU B%:INPUT ""S:B%=7
2160 UNTIL S>0 AND S<21 AND INT(S)=S
2170 =S
2180 :
2190 DEFFNmenu
2200 VDU 4,12,20,17,1
2210 FOR N%=0 TO 30
2220 PRINTTAB(0,N%)STRING$(20,CHR$(225))
2230 NEXT:VDU 28,2,26,17,3,12,26
2240 PRINTTAB(4,5);"PITFALL PETE";
2250 COLOUR 2:PRINTTAB(4,7);STRING$(12,CHR$(228));TAB(4,8);STRING$(12,CHR$(229));
2260 RESTORE 2380
2270 FOR N%=1 TO 6:READ L$,A$
2280 COLOUR 1:PRINTTAB(4,9+N%*2);L$;
2290 COLOUR 3:PRINT A$:NEXT:COLOUR 1
2300 PRINTTAB(4,24);"CHOICE ?";
2310 *FX21
2320 REPEAT G%=GET$
2330 C%=INSTR(" lLeEsSpCcQq",G$)DIV2
2340 IFG$=CHR$(27) C%=2
2350 UNTIL C%

```

```

2360 =C%
2370 :
2380 DATA L,OAD SCREENS,E,DIT SCREEN,S,
AVE SCREENS,P,LAY GAME,C,ODE ENTRY,Q,UIT
PROGRAM
2390 :
2400 DEFPROCsetup
2410 VDU23,225,183,-4,91,239,186,111,24
5,95
2420 VDU23,226,60,94,94,191,191,191,191
,191
2430 VDU23,227,191,-1,191,223,171,87,10
6,60
2440 VDU23,228,16,16,40,40,84,108,214,1
70
2450 VDU23,229,214,108,84,40,40,16,16;
2460 VDU23,230,-1;-1;-1;-1;
2470 VDU23,231;0;0;64,191,-1
2480 VDU23,232,36,90,90,126,36,60,24,60
2490 VDU23,233,126,189,24,60,36,36,1
02
2500 RESTORE 2570:FOR A%=0 TO 6
2510 READ N%:FOR C%=1 TO N%:READ V%
2520 F$(A%)=F$(A%)+CHR$(V%):NEXT,
2530 ENVELOPE 1,133,8,4,8,3,1,1,126,0,0
,-10,126,0
2540 ENVELOPE 2,1,0,0,0,0,0,90,-1,-2,
-3,97,97
2550 ENDPROC
2560 :
2570 DATA 4,32,10,8,32,6,17,2,225,10,8,
225,6,17,2,230,10,8,230,6,17,3,226,10,8,
227,6,17,1,228,10,8,229,6,17,1,231,10,8,
32,8,17,1,232,10,8,17,3,233
2580 :
2590 DEFPROCinit
2600 DIM F$(6),P%(20)
2610 M%=&4100:R%=&70
2620 P%=R%:[OPT 2
2630 LDA #135:JSR &FFFF4
2640 TXA:ORA #96:STA &40C:RTS:]
2650 *FX 225,2
2660 *FX 229,1
2670 RESTORE 2870:PRINT"PLEASE WAIT"
2680 L%=1:REPEAT
2690 READ F%:IF F%=-1 GOTO2780
2700 A%=FNADDR(L%):A%?282=F%
2710 READ A%?283,A%?284:A%?285=1
2720 FOR N%=1 TO 24:READ A$

```

```

2730 FOR Q%=1 TO 7+4*(N%-24) STEP 2
2740 H%=EVAL("&"+MID$(A$,Q%,2))
2750 ?A%#H% DIV36:A%?1=(H%-?A%*36)DIV6
2760 A%?2=(H%-?A%*36-A%?1*6)
2770 A%=A%+3:NEXT,:L%=L%+1
2780 UNTIL F%=-1
2790 FOR B%=L% TO 20:PROCclear(B%)
2800 A%?280=3:A%?281=1:A%?282=6
2810 A%?283=1:A%?284=2:A%?285=1:NEXT
2820 R=RND(-1):FOR N%=1 TO 20
2830 P%(N%)=N%*RND(50)+100:NEXT
2840 VDU 23;10,32;0;0;0;:SS=1
2850 ENDPROC
2860 :
2870 DATA 6,16,2,13A92E99,044F329E,5556
56C2,31552B37,2B2B5B50,50566256,502C3234
,62586278,55554813,6C0C0250,50685059,4AA
5323E,2B732C0F,7F555556,56304F3E,4F2D817
F,4A59C232,2E2B3210,63557A567,68485D81,4F
2E2B97,192B9856,56565456,0E5B
2880 :
2890 DATA 1,12,24,A34F2B2D,073F9F74,9E5
15956,815B325C,3E012B38,7F2B3130,A14FA07
5,7950685C,564F2E50,9D077338,A1563150,5D
550E2B,312C3487,9C4F9D55,3363164A,385B55
61,5E7CC19C,3F622B87,0D4F565C,31563250,7
57B2B2B,2E2B342E,612C9856,565E
2900 :
2910 DATA 6,17,2,81565656,564F2D3E,9F2B
7F74,2E2B5751,2D753133,32A4A580,374F617A
,7C2B2B2C,732D9F56,5531577F,385C994F,509
72B3E,7A077432,682B5232,63312B2B,2C61342
B,4F504F73,3D562B74,8145562B,2B4F31A9,A5
7A6774,65504F2B,2B2B2C4A,2B3B
2920 :
2930 DATA 6,17,12,815D8681,7A8733A4,815
8AB33,A3519F74,2B2B6374,75642B55,5061347
3,3D2C7A32,555E7350,A02B2B2C,AC329E58,4A
4F5656,562B32AA,329D8181,6F7A5581,5281A5
82,392DA42E,8187972B,316F8281,85327451,6
42E822B,5D2D3856,4F2B56A9,2B37
2940 :
2950 DATA 6,11,2,5C623973,337A542B,7FAC
2B5B,2E585779,2B2C7C2C,56395D74,335D329D
,38642B5C,97584F3E,56507B74,514F572B,336
1322B,2B2B565C,5955589E,56972B2E,4F564F2
B,2E2C3873,39015CAC,629F7382,517E0F82,64
2E862B,2A2B2C56,56565650,0E5F
2960 :
2970 DATA -1

```

B

```

2570 IF code%<129 THEN 2640
2580 PROCclearpanel(side$)
2590 PRINT 'y$;"Change"y$;"directory"
y$;"to";r$;name$
2600 IF P%=0 AND name$<>"^" THEN ldir$=
ldir$+"."+name$
2610 IF P%=1 AND name$<>"^" THEN rdir$=
rdir$+"."+name$
2620 IF name$="^" THEN PROCparent
2630 PROCpanelinfo(side$):PROCclearpane
l(side$):PROCdisplay(side$):R%=2
2640 IF code%<133 THEN 2810
2650 PROCclearpanel(side$)
2660 PRINT'y$;"File name:"m$;name$'
2670 PRINT'y$;"Copy....2"
2680 REPEAT:key$=GET$:UNTIL INSTR("2",k
ey$)>0:REM Modify in part 2.
2690 REM Add in part 2.
2700 IF key$="2" AND NOT rpanel% THEN P
RINT'r$;"No second"r$;"panel to"r$;"c
opy to.":REPEAT UNTIL GET:PROCdisplay(si
de$)
2710 IF key$="2" AND FNsamepanels THEN
PRINT'r$;"Both panels"r$;"the same!":R
EPEAT UNTIL GET:PROCdisplay(side$)
2720 IF key$="2" AND rpanel% AND NOT FN
samepanels THEN PROCcopy
2730 REM Add 2730 to 2800 in part 2.
2810 IF_code%=67 OR code%=99 THEN PROCm
akedirectory
2820 UNTIL endflag% OR code%=27
2830 VDU26:ENDPROC
2840 :
2850 DEFFNselect
2860 LOCAL code%,char%
2870 *FX229,1
2880 *FX21,0
2890 selflag%=TRUE
2900 PRINTTAB(20*P%,R%);g$
2910 REPEAT:key%=GET
2920 IF key%=139 AND R%=2 PROCdisplay("
up")
2930 IF key%=139 AND R%<>2 THEN PRINTTA
B(20*P%,R%);b$:R%=R%-1:PRINTTAB(20*P%,R%
);g$
2940 IF key%=138 AND R%=list%(P%+6) PRO
Cdisplay("down")
2950 IF key%=138 AND R%<list%(P%+6) THE
N PRINTTAB(20*P%,R%);b$:R%=R%+1:PRINTTAB

```

```

(20*P%,R%);g$
2960 IF NOT rpanel% THEN 3000
2970 IF key%=9 THEN PROCchangepanels
2980 IF key%=137 AND P%=0 THEN PROCchan
gepanels
2990 IF key%=136 AND P%=1 THEN PROCchan
gepanels
3000 UNTIL key%=13 OR key%=27 OR key%=6
7 OR key%=99
3010 PRINTTAB(20*P%,R%);b$
3020 IF key%<>13 THEN code%=key%:GOTO 3
100
3030 VDU31,(2+20*P%),R%
3040 code%=FNscreenchar
3050 name$="":VDU9:char%=FNscreenchar
3060 REPEAT
3070 name$=name$+CHR$char%
3080 VDU9:char%=FNscreenchar
3090 UNTIL char%=32
3100 selflag%=FALSE:*FX229,0
3110 =code%
3120 :
3130 DEFPROCchangepanels
3140 PRINTTAB(20*P%,R%);b$
3150 IF P%=0 THEN P%=1:side$="right":l
row%=R%:R%=-row% ELSE P%=0:side$="left":r
row%=R%:R%=lrow%
3160 PRINTTAB(20*P%,R%);g$:ENDPROC
3170 :
3180 DEFFNscreenchar
3190 A%=135:X%=0:Y%=255
3200 !&70=USR(&FFF4):=?&71
3210 :
3220 DEFPROCparent
3230 LOCAL dir$,N%
3240 IF side$="left" THEN dir$=ldir$ EL
SE dir$=rdir$
3250 N%=LENDIR$
3260 REPEAT
3270 N%=N%-1
3280 UNTIL MID$(dir$,N%,1)=". "
3290 N%=N%-1
3300 IF side$="left" THEN ldir$=LEFT$(d
ir$,N%) ELSE rdir$=LEFT$(dir$,N%)
3310 ENDPROC
3320 :
3330 DEFFNdirname(dir$)
3340 LOCAL N%,length
3350 IF dir$="$" THEN 3390

```

```

3360 N%=LENDIR$:length=N%
3370 REPEAT:N%=N%-1:UNTIL MID$(DIR$,N%,
1)="."
3380 DIR$=RIGHT$(DIR$, (length-N%))
3390 =DIR$
3400 :
3410 DEFFN$NAMEPANELS
3420 IF RDRIVE%=LDRIVE% AND RDIR$=LDIR$
THEN =TRUE ELSE =FALSE
3430 :
3440 DEFPROC$COPY
3450 LOCAL FROM$,TO$,NEWSIDE$
3460 VDU26
3470 IF SIDE$="left" THEN FROM$=":"+STR
$LDRIVE%+"."+LDIR$ ELSE FROM$=":"+STR$RD
RIVE%+"."+RDIR$
3480 IF SIDE$="left" THEN TO$=":"+STR$R
DRIVE%+"."+RDIR$ ELSE TO$=":"+STR$LDRI
VE%+"."+LDIR$
3490 PROCWINDOW("L",10,134,132)
3500 PRINT"Copy" FROM$;".";NAME$ "to" 't
o$
3510 PRINT'R$;"Okay? (Y/N)":VDU26
3520 TEMP$=GET$:IF TEMP$="Y" OR TEMP$="
Y" THEN 3550
3530 CLS:*SRREAD 7C00 7FE7 8000 4
3540 ENDPROC
3550 PRINT'TAB(7,18)R$;CHR$136;"Copying
the file."
3560 PROCFILECOPY(FROM$,TO$,NAME$)
3570 IF SIDE$="left" THEN NEWSIDE$="rig
ht" ELSE NEWSIDE$="left"
3580 CLS:*SRREAD 7C00 7FE7 8000 4
3590 PROC$PANELINFO(NEWSIDE$):PROC$DISPLA
Y(NEWSIDE$):ENDPROC
3600 :

```

```

3610 DEFPROC$FILECOPY(FROM$,TO$,NAME$)
3620 LOCAL X%,Y%
3630 OSCLI("DIR "+FROM$):A%=FNFILEINFO(
NAME$)
3640 LENGTH%=(BLOCK%?11)*&100+BLOCK%?10
3650 X%=CBLOCK%MOD256:Y%=CBLOCK%DIV256
3660 PTR%=0:N%=0:F%=OPENINNAME$
3670 REPEAT
3680 N%=N%+1
3690 IF LENGTH%<BUFSIZ% THEN COPY%=LENG
TH% ELSE COPY%=BUFSIZ%
3700 LENGTH%=LENGTH%-COPY%
3710 CBLOCK%?0=F%:CBLOCK%!1=CBUFFER%
3720 CBLOCK%!5=COPY%:CBLOCK%!9=PTR%
3730 A%=3:CALL(&FFD1)
3740 IF LENGTH%=0 THEN CLOSE#F%
3750 OSCLI("DIR "+TO$)
3760 IF N%=1 THEN T%=OPENOUTNAME$
3770 CBLOCK%?0=T%:CBLOCK%!1=CBUFFER%
3780 CBLOCK%!5=COPY%:CBLOCK%!9=PTR%
3790 A%=1:CALL(&FFD1)
3800 PTR%=PTR%+COPY%
3810 IF LENGTH%=0 THEN CLOSE#T% ELSE OS
CLI("DIR "+FROM$)
3820 UNTIL LENGTH%=0
3830 X%=BLOCK%MOD256:Y%=BLOCK%DIV256
3840 A%=1:CALL(&FFDD):ENDPROC
3850 :
3860 REM Temporary lines
3870 DEFPROC$MAKEDIRECTORY
3880 PROCWINDOW("L",10,130,129):PRINT"t
his facility not yet""available.""...
.any key to continue."
3890 REPEAT UNTIL GET$:VDU26:CLS
3900 *SRREAD 7C00 7FE7 8000 4
3910 ENDPROC

```

CORPLAN

For serious work with Wordwise Plus

- ❖ Reviewed in BEEBUG Vol.9.No.8
- ❖ Descriptive indexing for your letters & documents
- ❖ Your own library of layout forms, letterheads etc.
- ❖ Automatic import of addresses, references, dates etc.
- ❖ CORPLAN does the layout, you just type the text!
- ❖ Resident utilities for mailmerge, label printing etc.
- ❖ For B, B+ & Master. Needs discs & Wordwise Plus
- ❖ Many special features - see free information sheet
- ❖ Pack contains disc, tutorial manual, keystrip etc.
- ❖ Price £19.50, post free UK, 14 day refund



CORPLAN Computer Systems

Three Gables, 7a Talbots Drive, Maidenhead,
Berks, SL6 4LZ. Phone or Fax (0628) 2491

*Wish something new was happening for
your BBC Micro, Master or Electron?
Something is!*

BBC Public Domain software library has over
20Mb of software available!
Send £1.50 for catalogue and sampler disc to:

**BBC PD, 18 Carlton Close, Blackrod,
Bolton, BL6 5DL.**

Make cheques payable to;
'A Blundell' or send an A5 s.a.e for more details.
(Please state disc size and format)

RISC USER

The Archimedes Magazine & Support Group

Now in its fifth year of publication, RISC User continues to enjoy the largest circulation of any magazine devoted solely to the Archimedes range of computers. It provides support for all Archimedes users at work (schools, colleges, universities, industry, government establishments) and home. Existing Beebug members, interested in the new range of Acorn micros, may either transfer their membership to the new magazine or extend their subscription to include both magazines. A joint subscription will enable you to keep completely up-to-date with all innovations and the latest information from Acorn and other suppliers on the complete range of BBC micros. RISC User has a massive amount to offer to enthusiasts and professionals at all levels.

The Archimedes Magazine & Support Group

Here are some articles and series published in the most recent issues of RISC User:

MULTIMEDIA

A survey of the new developments in the multimedia field, in which the Arc looks likely to excel.

USING RISC OS 3

First look into the new operating system for the Archimedes.

ACORN LAUNCH A5000 SYSTEM

A review of the new breed of Archimedes, the A5000.

RECOVERING LOST DATA

Recovering programs and data lost in the Desktop.

PIPEDREAM 4

A review of the latest incarnation of this combined word processor and spreadsheet.

DRAW TOOLS

A toolbox for Draw allowing the precise creation of shapes such as circles, arcs, lines and sectors, at predetermined sizes and angles very simply.

PC COMPATIBILITY

A short series which investigates the extent to which PC applications work under the PC emulator.

THE SECRET LIFE OF TEMPLATES

An article and two programs which help you unfreeze the icon bar and Filer windows, and develop a simple routine for checking the contents of template files.

EXPLOITING THE WIMP

A series on some practical aspects of WIMP programming, which is currently treating the subject of redrawing windows and icons.

USING ANSI C

A series of articles on programming Desktop applications in C.

WP/DTP

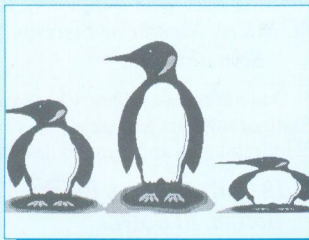
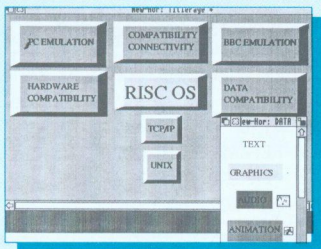
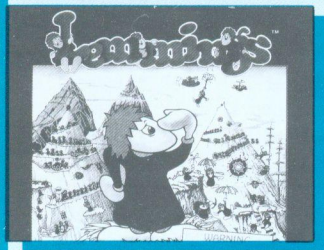
A regular column on using different DTP and WP packages. The latest article explains how to create a fold-out leaflet.

ARCADE

A round-up of the latest games for the Archimedes.

INTO THE ARC

A regular series for beginners currently explaining the various configuration settings that are possible on the Archimedes.



SUBSCRIPTION DETAILS

As a member of BEEBUG you may extend your subscription to include RISC User for only:

Destination	Additional Cost
UK, BPO & Ch Is	£ 10.50
Rest of Europe and Eire	£ 15.40
Middle East	£ 19.60
Americas and Africa	£ 21.90
Elsewhere	£ 33.00



LEARNING WITH BEEBUG

With reference to the correspondence over games in BEEBUG I think you should keep the magazine as it is. I especially like the *Function/Procedure Library* and the *First Course* series. I have learnt so much I did not know about the BBC micro in the year I have been subscribing. How about a book of the most popular articles/programs, or reprints of previous magazines? What about a First Course series based around machine code?

B.Allen

We always welcome readers' comments about the magazine. Our present intention is to keep broadly to the formula which has already proved popular, with just the occasional game (with the emphasis on thought and puzzle games) to lighten the proceedings. First Course will continue, and further instalments of the Function/Procedure Library will be appearing (though this is strongly dependent upon reader contributions).

*We have no immediate plans for a First Course type series on assembler, but readers may find useful the three previously published series **Exploring Assembler** in 11 parts from Vol.6 No.2 to Vol.7 No.2, **Using Assembler** from Vol.7 No.3 to Vol.7 No.7 and **Practical Assembler** from Vol.9 No.1 to Vol.9 No.10.*

*We have already published one book based on previous articles in BEEBUG, **File Handling for All** at £9.95, and plans for other books are currently under consideration.*

MORE RANDOM SAMPLING

Referring to the letter on the subject of Random Normal Deviates by Ron Larham in the November issue of BEEBUG (Vol.10 No.6), it seems to me that the reference to

'LOG' is surely a misprint for 'LN'. I have tested out the Box-Muller method, replacing LOG by LN, and the results are only approximately 'normal'. Moreover, the standard deviation of the values generated is 2, and not 1 as it should be. The following algorithm by Pyke and Hill is exact:

```
DEF PROC Gauss: LOCAL x,t
x=SQR(-2*LN(RND(1)))
t=2*PI*RND(1)
u1=x*SIN(t): u2=x*COS(t)
REM u1, u2 are random normal deviates
(0,1)
ENDPROC
```

This was published in *Communications of the Association for Computing Machinery* (Oct. 1965), and quotes Box & Muller in turn. It has the advantage over the Box-Muller method that the values generated are not truncated at both ends of the normal distribution.

A.G.Rimmer

My thanks to Mr.Rimmer for the interest he has shown in this subject, and for his time spent researching the original references.

FLOATING POINTS

In your Hints & Tips page in the October issue (BEEBUG Vol.10 No.5) the need to find the "closest" integer to a floating point value was discussed. The solution is to take the integer value of the floating point number with 0.5 added, i.e.:

```
N%=INT(val+0.5)
```

Trusting this is of interest.

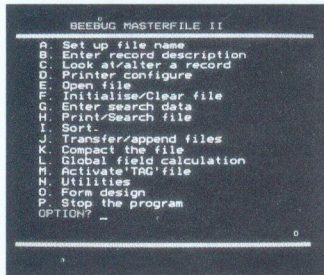
Laurie Everitt

Thanks to Mr.Everitt for reminding us of this standard approach to rounding. Ben Avison's hint provided a useful insight into the workings of the INT function in different circumstances.



BEEBUG

Software for the BBC Micro and Master series



Masterfile II

Masterfile II is the best selling general purpose database for the BBC Micro and Master 128. It has many powerful features, but is easy-to-use and ideally suited for home, business or school. It is supplied ready to run and no programming skills are required.

- ★ Menu driven for ease-of-use.
- ★ Fast 'tag' sorting.
- ★ Flexible print layouts.
- ★ Global calculations.
- ★ File size limited only by disc capacity.
- ★ File append options.
- ★ Search for a particular match.
- ★ Sorting on any field or fields.
- ★ Direct fast access to any record.
- ★ Form designer.
- ★ Label printing facility.
- ★ Record insertion.
- ★ Example database and tag files.

Normal price: **£22.48** inc VAT

Members price: **£16.86** inc VAT

Stock code 0024 40T, 0025 80T,
0081 ADFS for Master 128 only
Please add £2.50 carriage.



Command

The Command ROM is a powerful communications package that may be both menu or command driven. Command may be used with Hayes and other intelligent modems.

Text terminal - Use this to access scrolling text services such as Telecom Gold. XMODEM file transfer allows files to be sent around the world.

Viewdata terminal - A full feature terminal giving access to Prestel and other Viewdata services. Frames may be saved to disc, printed, tagged etc. and a full feature Viewdata editor is provided.

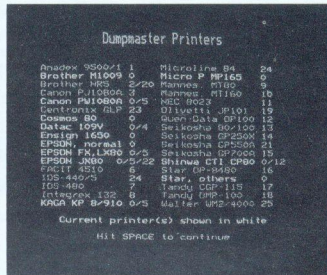
Telephone directory - Set up the name, number and log-on for services you wish to recall at a later date.

Command driven - A wide range of commands may be used in Basic to create applications for your own individual needs.

Normal price: **£39.84** inc VAT

Members price: **£29.88** inc VAT

Stock code: 0073 ROM
Please add £2.50 carriage.



Dumpmaster II ROM

Dumpmaster provides fast screen dumps using up to 8 shades from any screen mode. It includes a 'snapshot' facility to dump screens from programs while they are running. A wide range of printers are supported, including Epson compatibles and the Star LC10 colour.

Normal price: **£31.68** inc VAT

Offer price: **£19.00** inc VAT

Stock code 0053. ROM

Please add £2.50 carriage.

- Exmon II

This acclaimed ROM adds over 60 commands to manipulate and debug machine code programs. Features include:

- ★ set breakpoints
- ★ trace
- ★ set registers
- ★ full-screen RAM editor
- ★ single-step
- ★ disassembler
- ★ dual screens
- ★ search
- ★ line assembler
- ★ program relocater

Normal price: **£31.68** inc VAT

Members price: **£23.76** inc VAT

Stock code: 0004 ROM
Please add £2.50 carriage.

Quickcalc

Quickcalc is an easy-to-use disc based spreadsheet enabling you to use the calculating power of your computer without any need to program. It is ideal for personal accounts, stock control, and general financial planning.

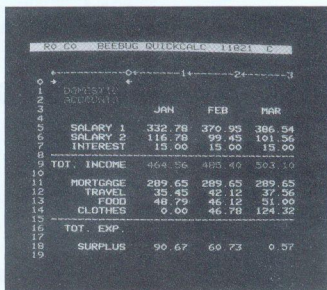
Normal price: **£18.39** inc VAT

Members price: **£13.79** inc VAT

Stock code: 0029 80T DFS
Please add £2.50 carriage.

Features include:

- ★ 20 x 50 default spreadsheet.
- ★ Load, save and print spreadsheet.
- ★ Replicate into row, column or area.
- ★ Simple histogram capability.
- ★ Individual column widths may be altered at any time.
- ★ Min, max and sum functions.
- ★ Very easy to use.



HINTS HINTS HINTS HINTS HINTS

and tips and tips and tips and tips and tips

This month we have a varied collection of hints, and don't forget, we pay for any hints published.

INVISIBLE BOOT

Paul Baron

If you don't like your !Boot files to write (for example) PAGE=&1900 then CHAIN"MENU" all over the screen, here is one method of preventing it. When *BUILdING the file (or editing it in a text processor), create the first line as XXXXX. Now use a disc sector editor to change the Xs to the following hex values:

```
1C 01 07 01 07
```

These are the VDU codes to create a zero sized text window. As it is the codes themselves that are used, nothing appears on the screen. However the first thing that the menu program (or whatever is run by the !Boot file) must do is to change the mode or text window, otherwise screen output will remain invisible for that program too.

ERROR LISTING

Eric Pope

To get a program automatically to list the line causing an error, use:

```
10 ON ERROR GOTO 100
.
.
99 END
100 ON ERROR OFF:MODE 7
110 A$="K.0L." +STR$(ERL) + " |M":$&900=A$:
X%=0:Y%=&9:CALL &FFF7
120 *FX138,0,128
130 END
```

ADFS AND DFS FORMATTED DISC

Charles Seager

If you only have a single drive, then copying from ADFS to DFS can involve an extreme amount of disc swapping. One way round this problem is to format a disc with side 0 to ADFS (M) format and side 2 to DFS format. Then the process of copying from ADFS to

DFS and vice versa is speeded up considerably, and once drive 2 has been selected from DFS (by *Drive 2), then *ADFS selects drive 0 and *DISC selects drive 2.

SINGLE KEY MERGE

Bruce Roberts

This function key definition will merge a specified Basic program onto the end of one in memory and then renumber the complete program ready for listing or running.

```
*K.0I."Merge",P$:P$="*L." +P$+" "+STR$~(
TOP-2)+CHR$13+"REN." +CHR$13:A%=138:X%=0
:F.C=0:TOLENP$:Y%=ASC(M.P$,C)):CA.&FFF4:
N.|M
```

CORRECTION TO FUNCTION KEY 10 HINT

Roger Smith

In Vol.10 No.5 the gremlins unfortunately had a go at one of our hints: "*FX" should be removed twice from the brackets half way through the hint to give "(using 202 (&CA) not 138 (&8A) as might be expected...". Our apologies.

OSBYTE 135 REVISITED

J M Shepherd

In the article "VDU and FX Calls (Part 6)" OSBYTE 135 is used to read a character at a given position on the screen. The value returned from the call is placed into Z% by Z%=USR(&FFF4), and then this must be put through some logic to extract the value returned in the X register. However as Basic stores the variable Z% at location &468, LSB first, a much quicker and shorter method of extracting the value of the X register would be to perform:

```
char=?&469
```

This is because &468 contains the accumulator, &469 the X register and &46A the Y register when Z% is set to the return value. However it should be noted that direct memory access is not a good practice in general. B

Personal Ads

BEEBUG members may advertise unwanted computer hardware and software through personal ads (including 'wants') in BEEBUG. These are completely free of charge but please keep your ad as short as possible. Although we will try to include all ads received, we reserve the right to edit or reject any if necessary. Any ads which cannot be accommodated in one issue will be held over to the next, so please advise us if you do not wish us to do this. We will accept adverts for software, but prospective purchasers should ensure that they always receive original copies including documentation to avoid any abuse of this facility.

We also accept members' Business Ads at the rate of 40p per word (inclusive of VAT) and these will be featured separately. Please send us all ads (personal and business) to MEMBERS' ADS, BEEBUG, 117 Hatfield Road, St. Albans, Herts AL1 4JS. The normal copy date for receipt of all ads will be the 5th of each month.

M128 E300, Turbo board £75, Acorn 2nd Processor unit £30, Cumana twin D S 5.25" 40/80T switchable D D with PSU £125, W.E. Master monitor stand with shelf for D D (metal) £10, Microvitec Cub 452 colour monitor £99, Delta twin joysticks £10, M512 2nd processor with essential software, memory upgrade, mouse, DOS+ ver. 1.2 & 2.1, Gem suite of software and mouse. Lots of ROMs, ROM cartridges, software, books: too many to mention please contact for more information, all software, hardware is original and come with guides & manuals. Tel. 081-684 9340 eves & w/ends.

M128 with BEEBUG Master ROM, Quad cartridge with Interword & sheet, Masterfile II disc, W.E. twin 5.25" 40/80T drive in self powered plinth, all manuals plus both reference manuals, all as new £425 o.n.o. full set of BEEBUG vol.1 to date, half in binders £100. Tel. (0203) 464256.

3.5" dual drive 80T D S £50, Peartree (MR8000) 64k battery backed cartridge for Master computer £25, Morley EPROM programmer and EPROM eraser £20 for both. Tel. (0679) 62728.

WANTED: Manual for Wordwise Plus, ring with price and particulars. Tel. (0473) 257954.

BBC B ADFS memo monitor dual disc drive Morley 20Mb SCSI hard drive, Wordwise, View Professional, Viewplot, Panasonic KXP-1080 printer desk £450, will split, Electron, Tape deck, games £75, Master co-processor £35, Linear graphics Plotmate scanner £25, Lintrac software including Plotmate ROM £20. Tel. (04867) 80632.

BBC B issue 7 with Watford DDFS, Integral 3.5" floppy drive and 32k Sideways RAM £160, Philips green screen monitor £20, BEEBUG printwise £10. Tel. (0751) 750058 day or (0653) 694173 eves.

512 co-processor upgraded to 1Mb, DOS 2.1 £130, 1 Acorn hard disc host adaptor £30, 2 Adaptec ACB4000 hard disc controllers £40 each or £70 for two. Tel. (0243) 265392.

A3000 2Mb computer with Acorn DTP £500, Master 128 with View and Viewsheet manuals etc £230, Philips CM8833 colour med. res. stereo monitor £150, Datachat 1223 modem V23 (SID etc) £35, ROMs - Exmon II £15, Viewstore £25, Printer monitor £7, BEEBUG discs - Dumpmaster £5, Masterfile £8, Quickcalc £7, View & Viewsheet manuals £5 each. Tel. 051-606 0289.

WANTED: BEEBUG Magazines Vol. 1 Nos. 1,2,8 to make my set complete. Can anyone help? Tel. (0387) 54526 eves.

7 Micro User binders boxed as new £10. Tel. (0387) 54526 eves.

Acorn 21Mb hard disc & utilities disc £200, 512 co-processor with essential software memory upgrade, including mouse, Gem Suite and all original manuals £200 o.n.o. M512 user guide and disc £5, latest Master OS ROM £15, Essential software's memopad 512 pop-up notepad £5, Hard disc compactor for 512 £5. Plus postage, all above in mint condition. Tel. (0326) 240734.

Kaga Taxan KX12 12" display monitor green screen, new in 1989, one owner only £29. Tel. 081-318 5155.

M128, twin 40/80 Cumana drive, 30Mb hard disc, 512 co-processor upgraded to 1Mb, DOS 2.1, RGB monitor, Epson printer, Morley Teletec, Comms adaptor, included ROMs, discs, manuals and magazines, upgrading forces sale £750 o.n.o. Tel. (0705) 466051.

BBC B, Microvitec 452 Cub colour monitor, also Cumana 40T D D and all connecting cables, in excellent condition £345 o.n.o. Arabic English system ALNOOR for BBC + plus complete set of Arabic alphabet keyboard covers £20, EPROM programmer MK11 for BBC from W.E. £15, Printmaster ROM with instructions £6. Please call 081-346 7228 w/days after 4pm or all day w/ends.

M128 (with upgrade OS chip), twin Cumana 40/80T switchable drive with PSU, Master internal modem, 14" colour TV/RGB monitor, Smart cartridge, Dumpmaster II ROM, plus lots of games utilities and reference books etc. £600. Tel. (0895) 672132 after 7.30pm.

M128, boxed and hardly used, with manuals and Morley ROM board £280, Acorn 80186 2nd processor with Essential 1Mb memory upgrade, DOS 2.1, Gem mouse, manual, software £160, Acorn Universal co-processor adaptor £40, Watford BBC video digitiser new and unused £65. **WANTED: A3000 system in good condition. Tel. (0952) 770633 eves, w/ends.**

M128 with 80186 co-processor including View, Word, Gem, Editor and Terminal Emulator, Mouse and all manuals, leads etc. £400, Cumana CS400S and CS400 5.25" disc drives, £135, Zenith Data Systems 14" green

monochrome VDU £50, software Mini Office II, Printer Driver, Graphics, NLQ and Dumpmaster on chip or disc, spare discs, BEEBUG magazines, books on basic programming £50, prefer to sell the lot for £600. Tel. (0626) 62782.

Will exchange Acornsoft Forth (on ROM) with manual by "Richard de-Grandis Harrison" for a copy of "Maths Workshop's 'Symbolic Calculus' with documentation". Tel. (0708) 855294.

HELP! instructions & software for Master to use "OmniReader" urgently required, will pay costs of copy and postage. Tel. (0462) 672383.

WANTED: Comms link hardware & software for down-loading files from Psion Organiser LZ to Master. Tel. (0462) 672383.

BBC B issue 7 Acorn DFS, 80T Watford DD with PSU, Acorn 6502 2nd processor with DNFS, Hi-Basic ROMs and manual, Microvitec Cub 895 colour monitor, Acorn cassette data recorder, pair of BBC joysticks, pair Voltmace joysticks, pair Voltmace joysticks with splitter box and driver program, Watford speech synthesiser with ROM and manual, Polarisising filter for Microvitec monitor, over 35 titles all in original boxes with manuals including software on ROM, disc and cassette, also 10 books, over 40 issues of BEEBUG, 20 issues of RISC User and more than 30 issues of Acorn User plus other assorted magazines. Phone for a list or make me a reasonable offer. Tel. (0277) 654343 between 5 and 10pm.

40+ BBC B tapes, all original & boxed, incl. Elite, Revs, Grand Prix, Scrabble, Sabre Wolf, Snapper, Hopper, Challenger and many many more, any reasonable offer accepted. Tel. 081-599 0442.

512 board with mouse and Gem software plus the Shibumi Problem Solver package £80, Play it again Sam 14 £5, 30 hour BASIC book £5, Master 4 slot switchable Care cartridge £8, EdiKit ROM £3. Tel. (0752) 896077.

WANTED: Master 128 manual and welcome disc, will swap for Master Compact (needs and o.s.). Tel. (0622) 756071.

512 board in Watford co-processor adaptor with mouse, DOS and manuals, hardly used £175 o.n.o. W.E. Le Modem with software and manual £45. Tel. (0724) 720675.

BBC B ADFS twin Cumana drives, Wordwise+, View Professional, BBC Graphics

ROM, Panasonic printer, Morley 20Mb SCSI hard disc drive computer desk £350 will split, also Viewplot disc £10, Master 65C102 co-processor board and support disc £50, Linear Graphics scanner Lintrack disc and Plotmate ROM £15 each, Cumana Graphics tablet £15, finally various books and tape based software £2 per item. Tel. (04867) 80632.

M128 £200, DS/DD 5.25" 40/80T DD with PSU £60, Interword ROM £20, W.E. Quest Paint £20, W.E. Wapping Editor £30, Spellmaster £15, Acornsoft Logo £20, Overview £20, BEEBUG C and Stand Alone Generator £35, lots of manuals covering many aspects of BBC computing. Tel. (0680) 61668.

Games for BBC, Master and Electron including Exile, Elite, Sams, Nevryon, Holed Out & E-Type, prices from £1, ROMs for BBC, Master & Electron, Advanced control panel and Advanced File Manager by PRES, brand new, £20 each or £35 for both, Quickshot 2 joystick with conversion program for BBC and Master 128 £5, new Advanced Ref. Manuals £10, Master 128 for High Flyers book £6. Tel. (061-225) 9169.

M128 discs and ROM cartridge, all 3 reference manuals and discs, Cub med. res. colour monitor, Viglen 40/80T DS drive, modem 1000, Viglen ext. ROM cartridge system (+6 cartridges) £450, ROMs, Lisp, MasterROM, Interword, Viewstore, AMX Pagemaker, all with manuals, plus many discs inc. speech, Viewspell and Keytogether with 8 vols. of BEEBUG magazine all for £100. Tel. (0924) 374566.

BBC B DFS with solidisk shadow 4Mhz sideways ROM RAM board and solidisk realtime clock, printer, 30 discs of games, ROMs & utilities, 80T disc drive, £220 o.n.o. Tel. (0565) 777 256.

1 Acorn hard disc host adaptor BBC £30, 2 Adaptec ACB-4000 hard disc controller £40 each, 1 512 board for BBC or BBC Master with Essential software, upgrade to 1Mb RAM plus books and software £130, 1 EPROM blower & eraser £25, 1 computer desk with master 128, twin drives 40/80, Microvitec colour monitor, modem, joysticks, Marconi Tracker Ball, loads of books and magazines £400 you collect (Bognor Regis) Tel. (0243) 265392 eves and w/ends.

BBC B 8271 DFS, 80T drive, speech upgrade, extras £150, Ferguson TX TV/RCB monitor £40, Watford shadow RAM board £20, volumes 1-6 BEEBUG, 1-4 Micro User, 4 volumes Acorn User from Jan 1984, all bound, offers? Tel. (0937) 835744.

M128 40/80T disc drive, ROM cartridges, printer, carrying case, software including Interword, Spreadsheet, Intergraph, manuals, magazines and discs £275 or exchange for Amiga compatible RGB monitor or VCR. Tel. (0462) 676761.

WANTED: ATPL board for the BBC B + 128, preferably with battery backup. Tel. 081-555 9069.

M128, twin 40/80T drives in Viglen (separate keyboard) console, Taxan 810 dot matrix printer, Hyperdriver, Advanced File Manager, BEEBUG modem, View, Viewsheets, Advanced Control Panel, Advanced File Manager, Printmaster, Masterfile II, various minor games

and educational software, manuals, some BEEBUG magazine discs, blank discs, offers around £350, about one third of the cost. Perhaps monitor also. Tel. (0590) 673929.

M128 with Interword, Spellmaster, Intersheet & other software £300, dual 80T drives £75, Star NLI10 printer £100, all boxed with manuals £450 the lot. Tel. 081-536 1238 eves.

WANTED: Early version of Ventura Publisher (say 1.01 for running on expanded M512. Tel. (0238) 510851 eves.

WANTED: Master reference manuals. Tel. (0786) 61501 eves and w/ends.

BBC B issue 7, Morley data recorder, Watford DDFS disc interface, DSSD 40/80T DD, manuals and software £170, Taxan colour monitor £60, Epson RX80 printer £60, all with manuals etc. and as new, test gear, Maplin 'Gold' MF100 function counter/freq-meter 100MHz perfect condition £48. Tel. 081-864 0309.

BBC Education Spelling it out, disc and book for BBC model B, B+, Master 128 £6, Crossword disc and instructions for BBC B £3.50, B&W monitor, Sinclair professional Series S-12MM £10, also assorted books. Please call for further information. Tel. 081-346 7228 w/days after 4pm.

M512 system selling separately, M128 £150, 512 co-processor with mouse, Gem software & manuals £100, dual 40/80T drives with PSU in plinth £100, Microvitec 1451 colour monitor £100, Panasonic KX1081 printer hardly used £80, Viewstore, Spellcheck II, ref. manuals 1&2, Dabs 512 guide & Dabs guides to Viewsheets, Viewstore and Master Operating System, book on assembly language programming, disc storage box. Please phone for list & asking prices. Tel. 061-427 2144.

M128 with 80186 512 co-processor including View, Word, Gem Editor and Terminal Emulator, mouse and all manuals, leads etc. £400, Cumana CS4005 and CS400 5.25" disc drives £135, Zenith data systems 14" green monochrome VDU £50, software- NiQ Office II, Printer Driver, Graphics, Mini and Dumpmaster on chip or disc, spare discs, BEEBUG magazines, books on basic programming £50, prefer to sell the lot for £600. Tel. (0632) 62782.

M128 with Turbo co-processor and MOS 3.50, Internal BEEBUG modem, 512 co-processor upgraded to run externally with 1Mb, Morley 40Mb hard disc drive, Microvitec 1431 monitor, Acorn teletext adaptor, Quest Tracker ball, Watford Video Digitiser, Hybrid Music 5000 system with Music 4000 keyboard and Music 1000, Watford user port splitter, Software includes View Professional, Overview, Wapping Editor, Masterfile, Dumpmaster, Master ROM, BEEBUG C and ADE+ macro assembler, would cost over £2000 to be new. Offers around £850 the lot or will split (buyer collect or pay carriage). Tel. (0460) 54887.

BBC B fitted in Oak PC type case (with separate keyboard), twin TEAC 40/80T disc drives/independent power supply, ADPS upgrade, ATPL sideways ROM board, Watford shadow 32k RAM card, AMX mouse, ROM software including: Interword, Intersheet, all user guides, handbooks and cables provided, complete set of BEEBUG magazines to current issue included, plus Quendata daisy wheel

printer (with spare ribbons), the lot for £325 o.n.o. Tel. (0753) 860690.

M128 with 30Mb hard disc, 80T 5.25" DS DD with PSU, View printer driver generator and View manual, offers around £600, buyer pays postage or collects. Tel. (0226) 340421.

M512, twin Cumana 5.25" drives, Apple green monitor, ROM cartridge, joysticks, Dabs & ADV. user guides, Ref. manual, software etc. £430. Tel. (0207) 504137.

M512 co-processor, Dabs 512 Technical & User guides with support discs (5.25 & 3.5), Gem & Ability - £100 o.n.o. BEEBUG C ROMs unused £20, Acorn GXR ROM £12, Diagram II ROM £15, Comstar Comms ROM £12, Image discs £12, Fleet St Editor (BBC B) £12, View Spellcheck £10, all software boxed with all documentation, sale due to upgrade. Tel. (0222) 461609 eves or w/ends.

M512 (Gem, mouse etc.) £350, dual Cumana 40/80T disc drive (CD8005) £100, Acorn Teletext adaptor (& ATS ROM) £35, Demon modem and ROM £35, green screen monitor £15, Overview £30, Viewstore £15, Viewsheets £15, ADI £15, Printmaster £15, Disc Doctor £15, complete set of BEEBUG (to date) £45, plus many other items. Tel. (0928) 564412.

BEEBUG from vol.2 No.7 to vol.10 No.5 over 70 copies in all, offers? Tel. (0395) 263638.

Archimedes 440, 4Mb RAM, 53Mb hard disc, 5.25" disc drive interface and Taxan 770+ multisync monitor, all boxed with manuals, as new £1,550. Tel. 081-445 7875.

Tandata modem Td1400, 1200/75bps operation with manual, just back from £70 overhaul, offers or swap for colour monitor for beeb. Tel. (0928) 722454.

EPROMs M27128 AFI, PCM12.5v SGS-Thomson brand new in anti-static strip £2 each or £25 for strip of 15. M128 £199, 80186 co-processor £99, 65C02 co-processor £45, Demon modem £35, B&W monitor £35, colour monitor £99, disc drive 80T £65, CC spell checker ROM £24, ROM cartridge £3, metal stand £12. Tel. (0234) 856070.

BBC B with DFS, 40/80T disc drive, loads of discs, also comes with Interword, art package, joysticks and mouse £200 o.n.o. Tel. 081-668 2595.

Star LC24-10 printer, with sheet feeder, almost new £230 o.n.o. Tel. (0582) 699483.

Archimedes 440 colour v.g. condition £1500 o.n.o. sale due to upgrade to 540, unused boxed Acorn DTP and PC Emulator included, many useful public domain utilities, printer available if required, some free training available as I am a lecturer in I.T. Tel. 081-368 0258 eves.

Archimedes 440 4Mb RAM, 53Mb hard drive 5.25" disc drive interface, Taxan 770+ multisync monitor £1500 all boxed with manuals, as new. Tel. 081-445 7875 eves.

BBC B issue 4, DFS, 40T single disc drive, green 7" monitor, spare EPROMs, all original packaging, welcome tapes etc. £180. Tel. (0634) 724156.

M128, joysticks, manuals, Viglen dual 40/80T drive, Interword, many games £325. Tel. (043871) 5496.

BEEBUG MEMBERSHIP

Send applications for membership renewals, membership queries and orders for back issues to the address below. All membership fees, including overseas, should be in pounds sterling drawn (for cheques) on a UK bank. Members may also subscribe to RISC User at a special reduced rate.

BEEBUG SUBSCRIPTION RATES

£18.40	1 year (10 issues) UK, BFPO, Ch.1
£27.50	Rest of Europe & Eire
£33.50	Middle East
£36.50	Americas & Africa
£39.50	Elsewhere

BEEBUG & RISC USER

£27.50
£41.50
£50.50
£55.50
£59.50

BACK ISSUE PRICES (per Issue) 1 July 1991

Volume	Magazine	5"Disc	3.5"Disc
6	£1.00	£3.00	£3.00
7	£1.10	£3.50	£3.50
8	£1.30	£4.00	£4.00
9	£1.60	£4.75	£4.75
10	£1.90	£4.75	£4.75
Binders	£4.20		

All overseas items are sent airmail. We will accept official UK orders for subscriptions and back issues, but please note that there will be a £1 handling charge for orders under £10 which require an invoice. Note that there is no VAT in magazines.

POST AND PACKING

Please add the cost of p&p when ordering individual items. See table opposite.

Destination	First Item	Second Item
UK, BFPO + Ch.1	£ 1.00	£ 0.50
Europe + Eire	£ 1.60	£ 0.80
Elsewhere	£ 2.40	£ 1.20

BEEBUG
 117 Hatfield Road, St.Albans, Herts AL1 4JS
 Tel. St.Albans (0727) 40303, FAX: (0727) 860263
 Manned Mon-Fri 9am-5pm
 (24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by RISC Developments Ltd.

Editor: Mike Williams
 Assistant Editor: Kristina Lucas
 Technical Assistant: Mark Moxon
 Production Assistant: Shella Stoneman
 Advertising: Sarah Shrive
 Managing Editor: Sheridan Williams

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, RISC Developments Limited.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

We are always seeking good quality articles and programs for publication in BEEBUG. All contributions used are paid for at up to £50 per page, but please give us warning of anything substantial that you intend to write. A leaflet 'Notes to Contributors' is available on receipt of an A5 (or larger) SAE.

Please submit your contributions on disc or cassette in machine readable form, using "View", "Wordwise" or other means, but please ensure an adequate written description is also included. If you use cassette, please include a backup copy at 300 baud.

In all communication, please quote your membership number.

RISC Developments Ltd (c) 1991

Printed by Arlon Printers (0923) 268328 ISSN - 0283 - 7561

Magazine Disc

December 1991
DISC CONTENTS

OP-ART PATTERNS - an entertaining program which produces a wide variety of dazzling optical art patterns

SELF NUMBERING GRAPH PROGRAM - a program which allows graphs to be drawn using "nice numbers", where the scale and numbering of the axes are chosen automatically to fit the scale of the graph.

A DISC ORGANISER FOR ADFS (1) - a handy utility which allows you to catalogue two ADFS discs simultaneously and copy files from one to the other.

PATTERNS FROM A KEYBOARD (2) - the application from last month for producing and printing text patterns from the keyboard, with added reverse feed and report options.

TELETEXT AND THE FUNCTION KEYS - a useful utility which sets up the function keys, so that all the teletext codes can be typed in from the keyboard.

BEEBUG WORKSHOP: SIMULATION MODELLING (3) another example of simulation modelling Event 2, which adds the factor of customer tolerance.

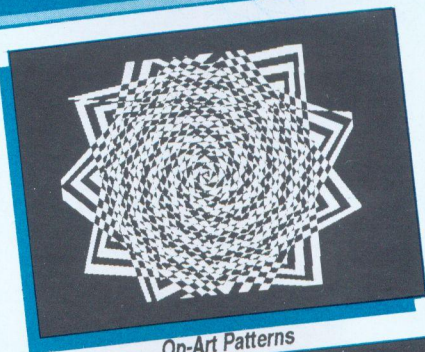
FUNCTION/PROCEDURE LIBRARY (7) - a new installment of 11 routines for use in producing screen displays.

PITFALL PETE - an updated version of this most popular "Repton" style game in which you must collect all the diamonds on each screen without getting trapped by a dislodged boulder.

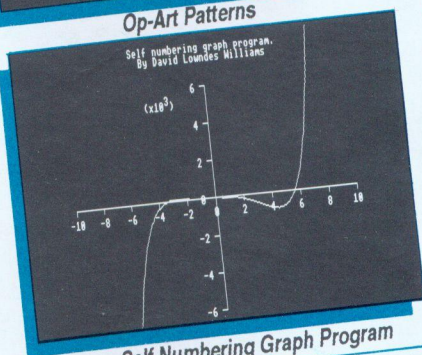
MAGSCAN DATA - bibliography for this issue

CHRISTMAS BONUS!
10 SCREENS FOR PITFALL PETE - ten additional screens to load into the game.

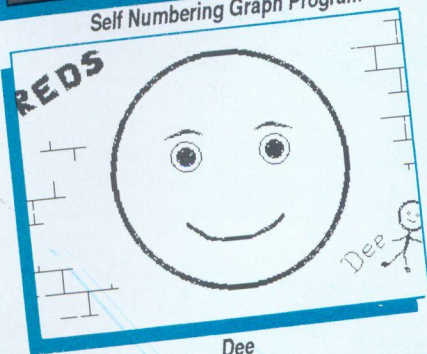
DEE - a chatty friendly face on the screen with whom you can have a weird and wonderful conversation.



Op-Art Patterns



Self Numbering Graph Program



Dee

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + £1 P&P (50p FOR EACH ADDITIONAL ITEM)
Back issues (5.25" disc since Vol.3 No.1, 3.5" disc since Vol.5 No.1) available at the same prices.

DISC (5.25" or 3.5") SUBSCRIPTION RATES
6 months (5 issues)
12 months (10 issues)

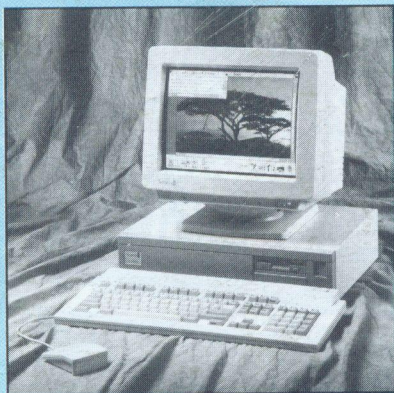
UK ONLY
£25.50
£50.00

OVERSEAS
£30.00
£56.00

Prices are inclusive of VAT and postage as applicable. Sterling only please.

RISC Developments, 117 Hatfield Road, St.Albans, Herts AL1 4JS

The New Acorn A5000



A5000 Features

- RISC OS Version 3
- ARM 3 For Unbelievable Speed
- 1.6 Mb New Format Floppy Drive
- 40 Mb IDE Hard Drive
- Acorn Multi-Scan Monitor

The New Learning Curve Pack

- New Multi-tasking PC Emulator
- Genesis 2 Database
- 1st Word Plus Wordprocessor
- Acorn DTP
- Lemmings and Pacmania Games
- Audio Training Tape
- Optional 300 dpi Ink Jet Printer

Acorn have just announced the release of the new A5000 computer. It is the latest in the Archimedes range and features the new version of RISC OS (not currently available on any other machine).

Based around the ARM 3 processor running at 25 MHz, this machine goes like lightning.

With a price of £1499 (plus VAT) for the A5000 with ARM 3, Multi-Scan Monitor, 2 Mb RAM and a 40 Mb IDE hard drive this machine really is excellent value.

A brochure on the A5000 is available free on request, please phone or write.

If you have been considering upgrading your system, there has never been a better time. However, the A5000 is in very short supply and so if you wish to be one of the first in the country with this new system, please use the form below to reserve yours now. Orders will be processed strictly in order of receipt. Should you need to change or cancel your order at any time until the system is despatched this will of course be no problem. Cheques will not be cashed and cards not debited until your system is sent.

Beebug are one of Acorn's largest dealers and have significant quantities of the A5000 on order. You may rest assured of our commitment to the Acorn marketplace and to our customers. Our full technical support team will be available to assist you with any problems that you may have in coming to grips with this new computer.

Voted 'Dealer Of The Year 1990' by A & B Computing (now Archimedes World) readers for customer service.

BEEBUG A5000 Computer Reservation

Please Reserve For Me (delete as appropriate)

Price INC VAT

- | | |
|--|----------|
| A) One 2 Mb A5000 with 40 Mb Drive & Multi-Scan Monitor | £1761.33 |
| B) One 2 Mb A5000 Learning Curve With 40 Mb Drive & Multi-Scan Monitor | £1799.00 |
| C) One system B (as above) with Acorn Ink Jet 300 dpi Printer | £2075.00 |

I wish to trade in the following equipment, please send me a quotation _____

Cheque enclosed, value £ _____ Debit My Access / Visa / Switch Card (No. as below)

Card No _____ / _____ / _____ / _____ Expiry Date _____ / _____

Please Deliver The System (NOTE: Add £9.00 courier charge) _____ I Will Collect from showroom _____

NAME _____ MEMBERSHIP NO _____ SIGNED _____

DAY TEL NO _____ ADDRESS (If Different) _____

SEND TO: A5000 Reservation, BEEBUG Ltd, 117 Hatfield Rd, St. Albans, Herts AL1 4JS