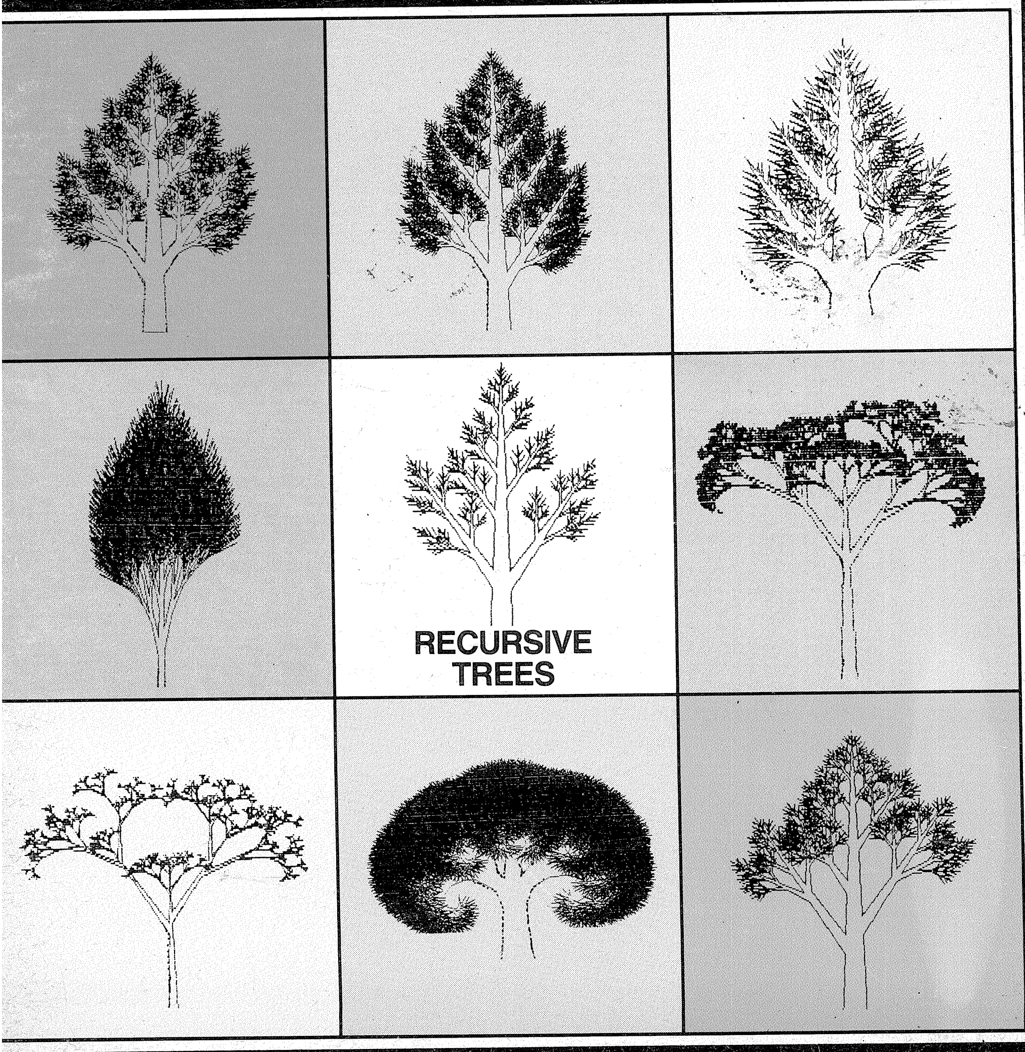
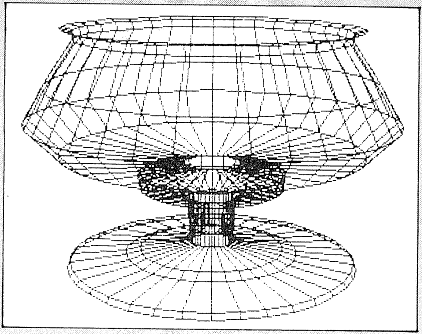
OLUME 4 NUMBER 5 OCTOBER 1985 PRICE £1.20

BEEBUG

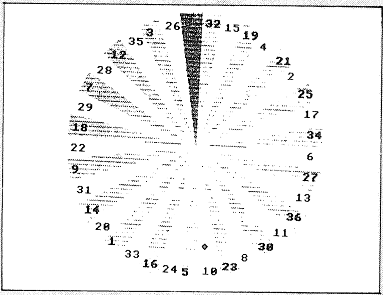
FOR THE BBC MICRO



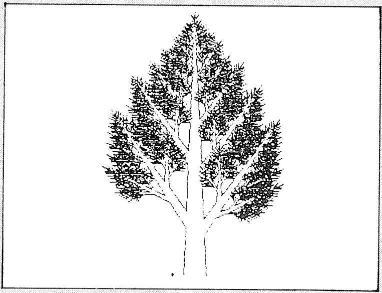
RECURSIVE TREES



Glentop's 3D Graphics System

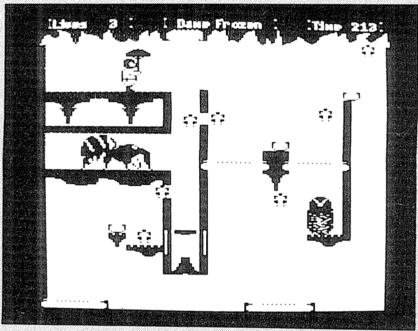


Roulette



Recursive Trees

Boffin



BEEBUG

VOLUME 4 NUMBER 5
OCTOBER 1985

GENERAL CONTENTS

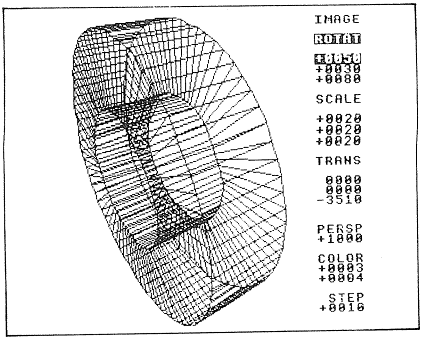
3 Editorial Jottings
4 Postbag
5 BEEBUGSOFT Forum
6 News
7 Recursive Trees
10 Glentop's 3D Graphics System
12 Dynamic Memory Window
15 Boffin – Review and Competition
16 Programming with Wordwise Plus (Part 1)
19 Hints for Wordwise Plus Users
19 Wordwise Help and Prize Competition
20 First Course
Print Formatting (Part 1)
22 Points Arising
23 EPROM Programmer Project (Part 2)
27 Loan Repayments
28 Double Density Disc Filing Systems
Compared
32 BEEBUG Workshop
Text Compression
34 Adventure Games
35 Advanced Disc User Guide Reviewed
36 Further Disc Menu Extensions
38 Data Structures (Part 2)
42 Interfacing the Beeb
Eight Books Reviewed
45 Roulette

PROGRAMS

7 Recursive Trees
12 Dynamic Memory Window
16 Wordwise Plus Segment Programs
23 EPROM Programmer Driver
27 Loan Repayments
32 Workshop Procedures
36 Disc Menu Extensions
38 Data Structures – examples
45 Roulette Game

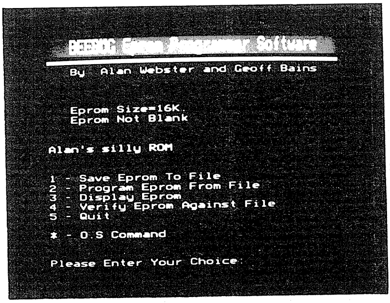
HINTS, TIPS & INFO

14 Music System Line Feeds
14 Z80 Filenames
35 Wordwise Plus Hints
VDU Codes
Sound
Filename Printing
Label Duplication
49 Local Points of Interest
49 Trouble with *FX3
49 Quick Random



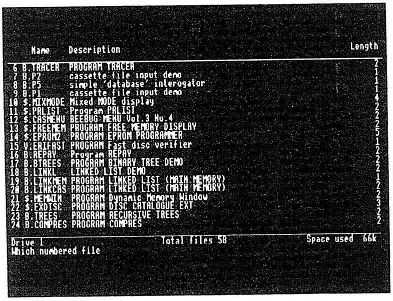
```
IMAGE

ROTAT

+0050
+0030
+0080

SCALE

+0020
+0020
+0020

TRANS

 0000
 0000
-3510

PERSP
+1000

COLOR
+0003
+0004

 STEP
+0016
```

Glentop's 3D Graphics System



```
 BEEBUG Eprom Programmer Software
 By  Alan Webster and Geoff Bains


 Eprom Size=16K.
 Eprom Not Blank

Alan's silly ROM

1 - Save Eprom To File
2 - Program Eprom From File
3 - Display Eprom
4 - Verify Eprom Against File
5 - Quit

* - O.S Command

Please Enter Your Choice:
```

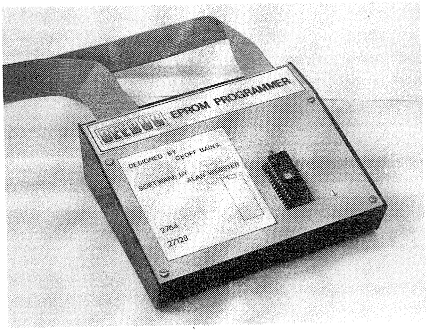
EPROM Programmer



```
    Name     Description                                              Length
 6 B.TRACER  PROGRAM TRACER                                                2
 7 B.P2      cassette file input demo                                      1
 8 B.P5      simple 'database' interogator                                 1
 9 B.P1      cassette file input demo                                      1
10 $.MIXMODE Mixed MODE display                                            4
11 $.PRLIST  Program PRLIST                                                2
12 $.CASMENU BEEBUG MENU Vol.3 No.4                                        2
13 $.FREEMEM PROGRAM FREE MEMORY DISPLAY                                   2
14 $.EPROM2  PROGRAM EPROM PROGRAMMER                                      5
15 V.ERIFAST PROGRAM Fast disc verifier                                    1
16 B.REPAY   Program REPAY                                                 2
17 B.BTREES  PROGRAM BINARY TREE DEMO                                      2
18 B.LINKL   LINKED LIST DEMO                                              2
19 B.LINKMEM PROGRAM LINKED LIST (MAIN MEMORY)                             1
20 B.LINKCAS PROGRAM LINKED LIST (MAIN MEMORY)                             2
21 $.MEMWIN  PROGRAM Dynamic Memory Window                                 2
22 $.EXDISC  PROGRAM DISC CATALOGUE EXT                                    3
23 B.TREES   PROGRAM RECURSIVE TREES                                       2
24 B.COMPRES PROGRAM COMPRES                                               2
Drive 1                      Total files 58                  Space used  66k
Which numbered file
```

Further Disc Menu Extensions

EPROM Programmer Project



EDITORIAL JOTTINGS

Last month saw the first article in our new regular series for newcomers to computing on the Beeb – First Course. Despite Acorn's much-publicized troubles, new BBC micro owners are still to be found in as large numbers as ever, and BEEBUG intends to continue to support all Beeb owners, regardless of experience, as earnestly as we have in the past.

New and established members alike will be interested in another new magazine feature starting this month. The software arm of BEEBUG, has been given a voice in the form of the BEEBUGSOFT Forum. A page of the magazine has been devoted to bringing you news of new BEEBUGSOFT products and help with the old ones.

The recent Acorn User show was well attended and provided an opportunity for members to see all these BEEBUGSOFT products demonstrated and to try them out for themselves. There is a similar opportunity to view the software and meet the people behind BEEBUG at the Micro User Show at the Horticultural Halls in London between November 14th and 17th.

This month's magazine cassette/disc contains a couple of extra items. The full disc benchmark program is there for members interested in how the results published in the Double Density Disc Systems review, in this issue, were obtained. The program runs on single density systems too, so you may like to test out your own DFS if you have one.

The second extra on the cassette/disc is a bit unusual, for BEEBUG at any rate. We have included one of the demonstration programs from Glentop's 3D Graphics Development System. This self-contained program will show you just the kind of sophisticated three dimensional animation that this package is capable of. See the back cover for the full details of all the programs on the magazine cassette/disc.

PROGRAM CLASSIFICATION

All programs in the magazine, and on the magazine cassette/disc, are marked with the symbols shown below. An uncrossed symbol indicates full working, a single line through a symbol shows partial working (normally some modifications will be required), and a cross through a symbol indicates a program that will not work on that system.

Basic I	I	Electron	◎
Basic II	II	Disc	💾
Tube	⊖	Cassette	📼
Model B+	+		

POSTBAG

POSTBAG

Logically speaking

Thanks to Peter Lewis for some interesting tutorials on the use of logic (see BEEBUG Vol.4 Nos.2 & 3). Here is a quirk that puzzled me:

```
    FOR J=Ø TO 5:PRINT
J,(J MOD 2):NEXT
```

This worked OK but not this extension:

```
    FOR J=Ø TO 5:PRINT J,
(J MOD 2);:IF (J MOD2) PR
INT " odd" ELSE PRINT " e
ven":NEXT
```

I changed the IF statement to:

```
    IF NOT(J MOD 2) PRINT
" even" ELSE PRINT " odd"
:NEXT
```

and then to:

```
    IF (J MOD 2) PRINT "
odd":NEXT ELSE PRINT " ev
en":NEXT
```

With this, I finally discovered that the IF must include its own NEXT as well as the ELSE.

More tutorials please.
Dennis Kemp

As Dennis Kemp found out, you can easily run into problems if you have NEXT in an IF-THEN-ELSE statement. The same can happen with REPEAT-UNTIL. It is best to avoid this situation altogether by using a different programming technique.

For example, in the above set type$(Ø)=" even" and type$(1)=" odd". The IF statement that caused all the trouble can then be replaced by:

```
PRINT type$(J MOD 2):NEXT
```

A Point of View

Users of View 1.4 may be interested to know that they can obtain View 2.1 in exchange for £1Ø and the 1.4 ROM from Acornsoft at Cambridge Technopark, 645 Newmarket Rd, Cambridge CB5 8PD. The two revised manuals "Into View" and "View Guide" are £2.5Ø each. However, View 1.4 users already have two good manuals. All we need is a note of the differences. The details (with thanks to BEEBUG) are as follows:

PRINT, SHEETS and SCREEN may be used on text in memory without saving it first. Just issue the command without a file name.

A new stored command, LJ (left justify), works in a similar fashion to RJ (right justify).

A new command FOLD can be used with the CHANGE command. With FOLD on (FOLD 1) characters being changed retain their original case. With FOLD off (FOLD Ø) characters to be changed take on the case supplied by the user (e.g. you can now change 'BASIC' to 'Basic', impossible before.
Joyce Diment

Clarss Distinction

The new Computer Concepts Speech ROM is a beautiful chip. Within minutes of installing it, I was able to produce quite a long sentence which was absolutely intelligible and with "expression". But then I thought I had found twc glaring omissions, the sound for "AW" as in "awful" or "law", and the sound for "AH" as in "task". They give for these, respectively, "OR" and "AR", and actually give examples "sor" for "saw" and "clarss" for "class".

I am not especially for clarss distinction but this is orful! However, I find on closer listening, that they do indeed pronounce "AR" as "AH", leaving one obliged to use an "AH" sound for "are". One can go too fah!

Regarding clarity, I realised that as I knew what I intended, that was what I heard when listening to the result. So I made a recording of some longer sentences and asked a group of blind, or nearly blind, people to listen. The general verdict was that they needed to hear it at least three times before the meaning was clear. I was rather crestfallen as I thought it was pretty good. Maybe the speed is just a little too fast.
M.B.Dearlove

We are hoping to review this new Computer Concepts ROM in the near future. The point about knowing what to expect is a good one which we will take up in our review.

BEEBUG SOFT FORUM

Welcome to the first BEEBUGSOFT Forum. We hope to make this page a regular feature in BEEBUG magazine to give us, at BEEBUGSOFT, a chance to keep you, the members, up to date on software developments.

We will also use this page to answer some of the questions on BEEBUGSOFT products that we are most frequently asked by members, and to give some hints and tips about using BEEBUGSOFT products. If you wish to raise any points concerning our products, ask any questions, or make any contributions that you feel would be of interest to other members, please write to BEEBUGSOFT Forum at the Editorial address.

We are always pleased to receive suggestions and ideas, and as you will have seen from last month's price list, we are happy to evaluate members programs and consider them for inclusion in the next BEEBUGSOFT catalogue.

WHO ARE BEEBUGSOFT?

BEEBUG Publications and BEEBUGSOFT are entirely separate entities, with separate staff. We are however, located in the same offices and often work closely together. Our aims are similar: to produce, respectively, the magazine and the software that will be of interest to you, the members.

WORDEASE

At the Acorn User exhibition we launched our new WordEase program, a utility to assist with, and add to, Wordwise Plus. If you have any queries on WordEase, or suggestions for any further extensions to Wordwise Plus, we'd be pleased to hear from you.

PAINTBOX II OR PAINTMASTER?

Our popular drawing package, Paintbox II, has been renamed 'Paintmaster'. A company called Quantel has a prior registration of the name Paintbox, and has requested that we rename our program to avoid possible confusion. Paintmaster is identical in every way, except name, to Paintbox II.

NEW BEEBUG DISCS

We have just increased our range of BEEBUG blank discs. We can now offer 4 types of disc to meet your exact requirements:

ss dd 48TPI ds dd 48TPI
ss dd 96TPI ds dd 96TPI

Unfortunately we have also had to pass on a small increase in price. Full details are on the inside back cover of BEEBUG.

EXMON II HINT

Exmon II is able to assemble directly from disc using files created on a wordprocessor such as Wordwise, giving a number of advantages, including greater available memory. This useful facility is covered in the Exmon manual (see page 26) but many users seem unaware of it.

QUITTING MASTERFILE

When using Masterfile II it is essential that you use option "P" to quit the program, as described in the manual. If you press the Break key you may cause damage to your data because files will not have been updated and closed in a correct manner.

LICENCES

All BEEBUGSOFT products are subject to copyright. Unauthorised copying, apart from the production of one backup copy only for use by the purchaser, is strictly prohibited. However, we operate a licensing scheme to allow schools and colleges to have access to, and to produce, multiple copies of our programs at special rates. Further details are available from the Software Manager at the St. Albans address.

COMING SOON

We will shortly be increasing the amount of software and hardware available from BEEBUGSOFT. Watch this space for further information.

NEW TELEPHONE NUMBER

Please note our new telephone number for order queries and for technical enquiries is:

St. Albans (Ø727) 4Ø3Ø3.

All telephone orders should still be placed on Ø49481 6666.

News News News News News News.........

Aries the Ram

Aries Computers, the first to produce a shadow RAM board for the Beeb, has now released a new version offering 32K extra RAM. This is 2ØK screen memory and 12K sideways RAM, along the same lines as the B+, and includes a sideways ROM socket to take the controlling ROM. The B-32 board costs £92 from Aries on Ø223-862614.

Music Countdown

Hybrid Technology, designer of the Music 5ØØ, has previewed further models to extend the range, for a November release. The Music 4ØØ is the long-awaited keyboard, the Music 2ØØ is an interface to the industry standard MIDI specification and the Music 1ØØ is a monitor amplifier offering 4 watts per channel. The music control language, Ample, to look after all of these, is to appear in new clothing as the 'Ample Nucleus' in ROM. Reviews of all coming soon. Hybrid also has an audio tape of the Music 5ØØ in action for potential users. The tape costs £1.95 from Hybrid on Ø223-31691Ø.

Keynotes

Two new numeric keypads have arrived on the Beeb scene. Keypad from Commercial products (Ø293-3Ø174) costs £44.95 and connects in parallel with the main keyboard and so is totally compatible with all software. It offers the numbers Ø-9, decimal point and Return. The Kenpad costs £6Ø from K. Blanchard (Ø4446-4129Ø), one of the authors of Acornsoft's Termulator, and is designed for use with that ROM. It plugs into the User Port and offers Ø-9, point, Return, and four special function keys.

Multi-Tasking

Multi-Basic is a sideways ROM that adds multi-tasking ability to your Beeb. Up to 8 background tasks, written in standard BBC Basic, can be operative at one time and triggered by time or logical conditions. The ROM also includes several I/O control commands for interfacing enthusiasts. Multi-Basic costs £34.95 from CMS on Ø371-5666.

The Disc Drive's Friend

BS-DOS is the latest DFS for the Beeb from CUC. BS-DOS costs £39.5Ø and works alongside any Acorn-type DFS to provide several enhancements. Double sided disc drives are now configured as single drives, across the two surfaces, with up to 256 files. BS-DOS is compatible with all second processors (including Torch) and supports all the normal OS calls. Further details from CUC on Ø1-311 2555.

Space Bits

A self-contained, simple to use satellite data receiving and decoding package is available for the BBC micro. Astrid will receive, decode, store and display data from UoSAT 1 and 2. Astrid costs £149 from MM Microwave on Ø751-75455.

Bilbo II

The infamous Hobbit adventure from Melbourne House (Ø1-94Ø 6Ø64) is now available in a much expanded form on disc for the Beeb. The new Hobbit has more locations and over 5Ø graphic screens and costs £17.95.

More Teletext

Morley Electronics has released a teletext adaptor for the Beeb. The Morley adaptor connects to the User Port and, along with the inevitable sideways ROM, enables your Beeb to receive all teletext information and to down load programs. The tuning of the receiver is entirely under software control. The adaptor with ROM and power supply costs £1ØØ.

The Case for the Beeb

Oak Computers has launched a new and impressive looking cabinet for your BBC micro. Moulded in light grey plastic, the 'Personal Computer' range has a detached keyboard and a main cabinet which can house the main circuit board, power supply, disc drives, second processor, modem, etc. Prices start from £1ØØ. Further details from Oak on Ø274-614167.

Recursive Trees

William Godwin uses the technique of recursion to produce a veritable forest of delights.

In the world of home micros the BBC micro stands out as having one of the most powerful recursion facilities. This program exploits this technique to the full to produce a variety of tree designs.

To create a tree, the various parameters that define the tree's features are first edited. These are explained in full in the program notes section, but briefly they comprise the tree's height width and density of branches. To select a parameter, you move the flashing cursor across the screen using the left and right cursor keys. Then you can either type in a new value for the parameter by first pressing the space bar, or use the up and down cursor keys to change the current digit of the value displayed.

For example, to change the number 1ØØ to the number 2ØØ, you could place the cursor anywhere underneath the number, press the space bar and enter the new value, 2ØØ. Alternatively, you could place the cursor under the digit 1 and press the cursor up key to alter the digit 1 to a 2.

Each time you alter a parameter, the basic shape of the tree is drawn. This is not an exact copy of how the tree will finally look; it will be much more complex than this. It is just a rough guide.

When you are satisfied with your parameters, pressing the Return key will then start the drawing of the tree. If you get fed up with waiting for a certain tree to be drawn, as you will with one that involves a high level of recursion, pressing the Delete key will halt the drawing of the tree and allow you to return to the 'edit mode'.

The trees can take a long time to be drawn. The time taken is largely dependant on the value for 'TWIG'. Small values, such as 8, give very detailed trees that take a long time. Try larger values until you are satisfied with the general shape of the tree.

The trees produced by the program can be dumped to any Epson-compatible printer by pressing the Tab key. This initiates the screen dump routine in PROCgemini. If you have a printer that is not Epson-compatible you can insert your own screen dump routine in the procedure, PROCgemini and call it appropriately from line 218Ø. Alternatively you can call one of the various screen dumps in ROM from this line (*GDUMP for Printmaster, for example) and leave out PROCgemini (and line 13Ø) altogether.

RECURSION

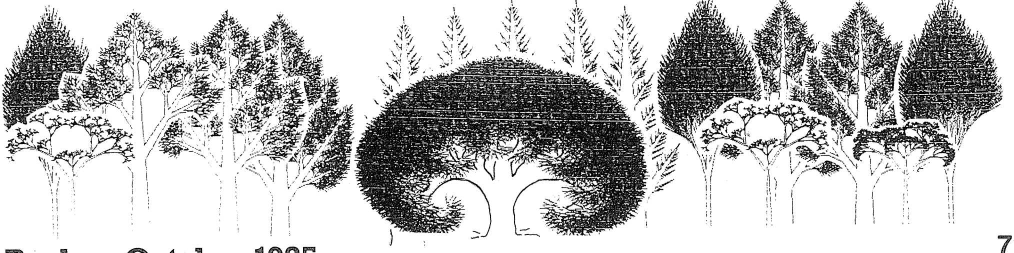
A recursive routine is one that calls itself. Obviously such a state of affairs could go on forever in an endless loop. The usual implementation of recursion in BBC Basic, therefore, is a function or a procedure that calls itself only if a particular condition is met. The condition in the case of this program is the length of the current branch being drawn. The drawing process - splitting the current branch recursively - is repeated until the branch is smaller than TWIG.

PROGRAM NOTES

The size and shape of the tree is determined by six parameters:

TWIG This gives the size of the smallest detail. The tree is drawn by drawing a smaller branch on each branch until the twig size is reached

TREE This gives the height of the main tree trunk



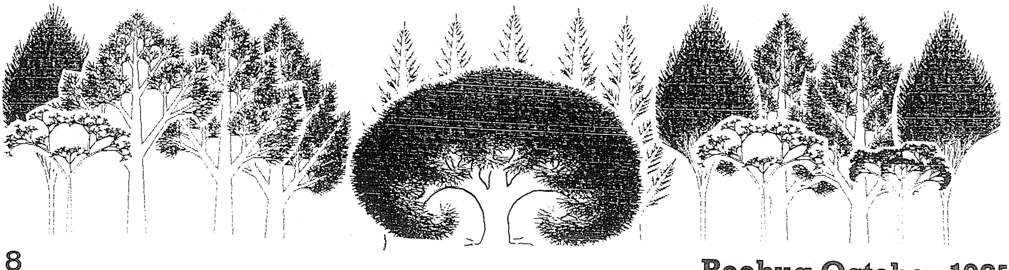
ANGLE This is the angle at which new branches grow. A new branch would grow at right angles to the previous one if an angle of 9Ø degrees was stated

LIMB This value governs the thickness of a limb (branch)

HIGH This determines approximately how high up a new branch will grow

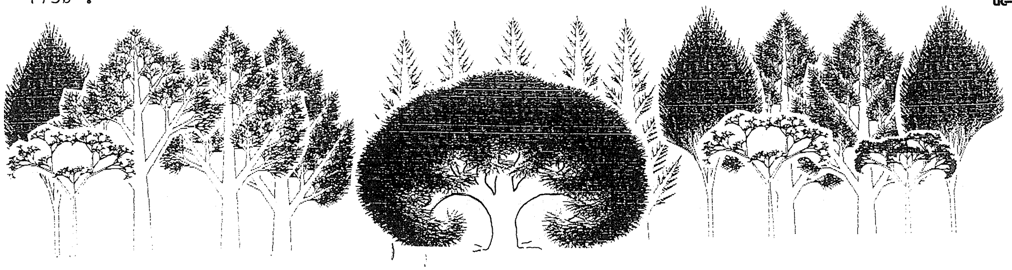
LONG This gives the relative length of a limb. The maximum value is 999, but high values usually cause too much recursion

```
   1Ø REM PROGRAM TREE
   2Ø REM VERSION BØ.2
   3Ø REM AUTHOR  W.GODWIN
   4Ø REM BEEBUG  OCTOBER 1985
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ ON ERROR GOTO21Ø
  11Ø MODE1:PROCintro
  12Ø N%=5:DIM param(N%),code%&86
  13Ø IF code%?&85<>&6Ø PROCgemini
  14Ø F%=6:@%=F%
  15Ø FOR J%=Ø TON%:READ param(J%):NEXT
  16Ø J%=Ø:REPEAT PROCedit
  17Ø PROCtree(tall,twig,64Ø,8,trunk,H,u
p,down)
  18Ø *FX15 1
  19Ø IFGET=9 PROCprint
  2ØØ UNTIL FALSE
  21Ø ON ERROR OFF:*FX4
  22Ø MODE7:REPORT:PRINT" at line ";ERL
  23Ø END
  24Ø :
 1ØØØ DEF PROCedit:*FX4 1
 1Ø1Ø K%=1:Y%=F%-1:VDU22,1:IF FNtest THE
N21Ø
 1Ø2Ø PROCdisplay
 1Ø3Ø REPEAT PRINTTAB(F%*J%+Y%,31);
 1Ø4Ø A%=GET
 1Ø5Ø IFA%=32PRINTTAB(F%*J%+1,31)">>";:I
NPUTTAB(Ø,29)"VALUE",K:PROCalter(K-param
(J%))
 1Ø6Ø IFA%=138PROCalter(-K%)
 1Ø7Ø IFA%=139PROCalter(K%)
 1Ø8Ø IFA%=136PROCleft
 1Ø9Ø IFA%=137PROCright
 11ØØ UNTILA%=13:VDU22,Ø:ENDPROC
 111Ø :
 112Ø DEF PROCalter(X%)param(J%)=param(J
%)+X%
 1125 IFFNtest param(J%)=param(J%)-X%:EN
DPROC
 113Ø CLS:PROCdisplay:ENDPROC
 114Ø :
 115Ø DEF PROCleft
 116Ø IFK%<1ØØØK%=K%*1Ø:Y%=Y%-1:ENDPROC
 117Ø IFJ%>ØJ%=J%-1:K%=1:Y%=F%-1
 118Ø ENDPROC
 119Ø :
 12ØØ DEF PROCright
 121Ø IFK%>1K%=K%DIV1Ø:Y%=Y%+1:ENDPROC
 122Ø J%=(J%+1)MOD(N%+1):K%=1:Y%=F%-1:EN
DPROC
 123Ø :
 124Ø DEF PROCdisplay
 125Ø PROCtree(tall,tall*tall-1,64Ø,8,tr
unk,H,up,down)
 126Ø PRINTTAB(Ø,3Ø)"  TWIG  TREE ANGLE
 LIMB  HIGH  LONG"
 127Ø FORI%=ØTON%:COLOURI%MOD2+1:PRINTpa
ram(I%),;:NEXT
 128Ø ENDPROC
 129Ø :
 13ØØ DEF FNtest
 131Ø IFparam(Ø)<2THEN=TRUE ELSEtwig=par
am(Ø)*param(Ø)
 132Ø theta=PI-RADparam(2):phi=PI-RADpar
am(3)-theta:CT=COStheta:ST=SINtheta:CP=C
OSphi:SP=SINphi
 133Ø H=param(4)*.ØØ1:up=param(5)*.ØØ1:K
=up*ST
 134Ø down=K/SP:D=K*(CT/ST+CP/SP):E=1-H-
D
 135Ø tall=param(1):trunk=tall/TAN((thet
a+phi)*.5)
 136Ø IFtall<Ø OR E<Ø OR D<Ø OR H<Ø THEN
=TRUE
 137Ø IF5Ø*LNup>LN(twig/tall/tall)THEN=T
RUE
 138Ø IF5Ø*LNdown>LN(twig/tall/tall)THEN
=TRUE
 139Ø =FALSE
 14ØØ :
 141Ø DEF PROCtree(tall,twig,P,Q,trunk,H
,up,down)
```



```
1420 P=P-trunk:MOVEP,Q:PROCuptree(trunk
,tall):PROCdowntree(trunk,-tall)
1430 ENDPROC
1440 :
1450 DEF PROCuptree(X,Y):LOCALU,V,K,F
1460 K=H-RND(1)*(H*(H<E)+E*(E<H)):F=1-D
-K
1470 IFX*X+Y*Y<twig PROCline(X,Y):ENDPR
OC
1480 PROCline(K*X,K*Y):PROCturn(CT,ST):
PROCuptree(up*U,up*V)
1490 PROCturn(CP,-SP):PROCdowntree(down
*U,down*V):PROCuptree(F*X,F*Y)
1500 ENDPROC
1510 :
1520 DEF PROCdowntree(X,Y):LOCALU,V,K,F
1530 K=H-RND(1)*(H*(H<E)+E*(E<H)):F=1-D
-K
1540 IFX*X+Y*Y<twig PROCline(X,Y):ENDPR
OC
1550 PROCdowntree(F*X,F*Y):PROCturn(CP,
SP):PROCuptree(down*U,down*V)
1560 PROCturn(CT,-ST):PROCdowntree(up*U
,up*V):PROCline(K*X,K*Y)
1570 ENDPROC
1580 :
1590 DEF PROCturn(C,S):U=X*C-Y*S:V=X*S+
Y*C:ENDPROC
1600 :
1610 DEF PROCline(X,Y)
1620 IFINKEY(-90)twig=1E6
1630 P=P+X:Q=Q+Y:DRAWP,Q:ENDPROC
1640 DEF PROCintro
1650 COLOUR2
1660 PRINT'"       TREE DESIGNER"
1670 PRINT'"Use the ARROW KEYS to alter
 parameters"
1680 PRINT'"    or SPACE  to enter a va
lue;"
1690 PRINT'" Press RETURN to draw the T
REE"
1700 PRINT'" (then DELETE to escape)"
1710 PRINT'"    or TAB    to send to pr
inter."
1720 PRINT'" Press SPACE to continue."'
:IFGET
1730 ENDPROC
1740 DATA90,900,150,4,100,650
1750 :
```

```
1760 DEF PROCgemini
1770 COL=&70:ROWS=&71:LOC=&72:STO=&74:W
R=&FFEE:FX=&FFF4
1780 FORI%=0TO2STEP2:P%=code%:[OPTI%
1790 LDA#2:JSR WR
1800 LDA#27:JSR PR:LDA#65:JSR PR:LDA#8:
JSR PR
1810 LDA#&84:JSR FX:STX LOC:STY LOC+1
1820 LDA#32:STA ROWS
1830 .BEGIN
1840 LDA#80:STA COL
1850 LDA#27:JSR PR:LDA#76:JSR PR:LDA#12
8:JSR PR:LDA#2:JSR PR
1860 .BLOCK
1870 LDY#7
1880 .FETCH
1890 LDA(LOC),Y:STA STO,Y
1900 DEY:BPL FETCH
1910 LDY#8
1920 .SEND
1930 LDX#7
1940 .PINS
1950 ASL STO,X:ROR A
1960 DEX:BPL PINS
1970 JSR PR
1980 DEY:BNE SEND
1990 LDA LOC:CLC:ADC#8:STA LOC
2000 BCC NOC
2010 INC LOC+1
2020 .NOC
2030 DEC COL:BNE BLOCK
2040 LDA#13:JSR PR
2050 DEC ROWS:BNE BEGIN
2060 LDA#27:JSR PR:LDA#48:JSR PR
2070 LDA#3:JSR WR
2080 RTS
2090 .PR
2100 PHA:LDA#1:JSR WR
2110 PLA:JSR WR:RTS
2120 ]NEXTI%
2130 PRINT~P%:ENDPROC
2140 :
2150 DEF PROCprint:*FX3 10
2160 FORI%=0TON%:PRINTparam(I%),;:NEXT:
PRINT
2170 *FX3
2180 CALLcode%:ENDPROC
```



Glentop's 3D Graphic System

Geoff Bains takes a look at a new versatile graphics development system that promises to get more from your Beeb's screen than you ever thought possible.

Product : Graphics Development System
Supplier: Glentop Publishers,
          Freepost, Barnet,
          Herts.
          Ø1-441 413Ø
Price   : £25.ØØ

There are many graphics systems on the market for the BBC micro. These will allow you to create pictures of varying complexity on your Beeb's screen and display them. What all these packages lack, however, is any real application for the resulting pictures. All you can do is to save them to cassette or disc and load them in again at a later date.

The Graphics Development System from Glentop Publishers, however, is a different story. This unique package allows you to create wireframe representations of 3D objects on the screen, manipulate them in a variety of ways and then to use the resulting images in Basic or assembler programs to produce some stunning animated images.

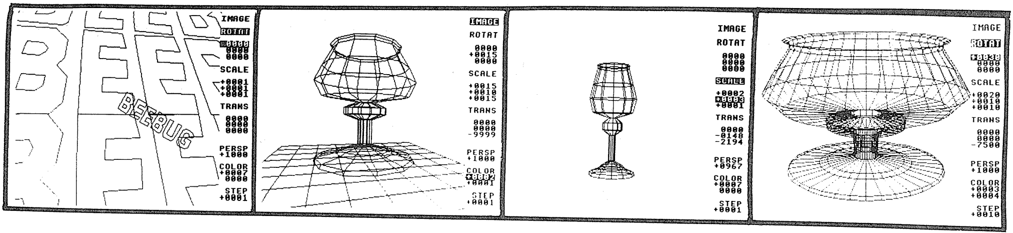
The graphics creation programs are a little different from many on the market. This package is exclusively concerned with wireframe models. There is no block filling, shading, or even much in the way of colour. The models are defined as a series of three-dimensional co-ordinates, representing each point of the wireframe, entered into an editor. As well as the co-ordinates each point is also defined as a draw or move operation with the colour and type (solid or dotted) of line used.

The data tables produced by the editor can be saved onto disc and reloaded for further editing. Unfortunately there is no option to insert extra points into the middle of a data table; these can only go onto the end. This does mean that alterations can take up unnecessary extra table entries.

Once an object's data is entered it can be viewed on the screen. The program displays a perspective projection of the wireframe and allows you to alter the viewpoint and perspective, and to perform rotations, scalings, and translations in three dimensions. The various parameters are displayed alongside the image and are altered using the cursor keys.

That's all very well but it doesn't go very much further than the 3D Rotation program in BEEBUG Vol.1 No.1Ø. However, the Graphics Development System doesn't stop there. Shapes can be rotated about any of the three axes to form surfaces. The program to do this is very versatile, allowing either the lines parallel or perpendicular to the axis of rotation only to be created, or both, and the angle through which the rotation is performed and the step size to be altered at will. The creation of such 'profiles' is, like all the operations in the Graphics Development System, extremely fast.

Any wireframe model can be used as a 'macro' and stretched, translated, rotated, enlarged in different ways with each result saved. All these new models



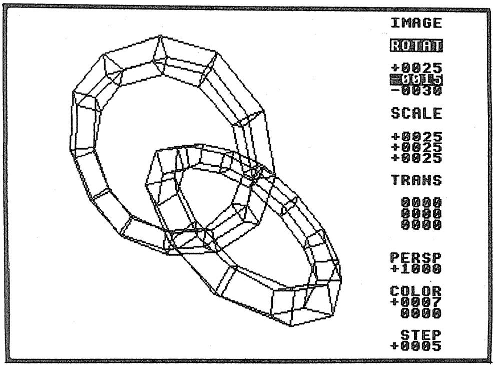
```
IMAGE
ROTAT
0000
0000
0000
SCALE
+0001
+0001
+0001
TRANS
0000
0000
0000
PERSP
+1000
COLOR
+0007
0000
STEP
+0001
```

```
IMAGE
ROTAT
0000
+0015
0000
SCALE
+0015
+0010
+0015
TRANS
0000
0000
-9999
PERSP
+1000
COLOR
+0002
+0001
STEP
+0001
```

```
IMAGE
ROTAT
0000
0000
0000
SCALE
+0002
+0003
+0001
TRANS
0000
-0148
-2194
PERSP
+0967
COLOR
+0007
0000
STEP
+0001
```

```
IMAGE
ROTAT
+0030
0000
0000
SCALE
+0020
+0010
+0010
TRANS
0000
0000
-7500
PERSP
+1000
COLOR
+0003
+0004
STEP
+0010
```

can be then linked together to form a new creation. In this way complex models can be built up with the minimum of data entry. The interlocking rings shown here, for example, were created by manipulating a simple diamond shape (five data entries only!). This was turned about one axis to give a profile and the resulting ring used as a macro. This was translated and rotated to form the two rings which were then linked together to form the whole picture.



```
IMAGE
ROTAT
+0025
-0015
-0030
SCALE
+0025
+0025
+0025
TRANS
 0000
 0000
 0000
PERSP
+1000
COLOR
+0007
 0000
STEP
+0005
```

The only limit to this model creation is the Beeb's memory. The Graphics Development System operates perfectly with the 65Ø2 second processor, however, so larger models are available with this connected. The package also has a disc and printer dump option allowing you to save your masterpieces in a more immediate form.

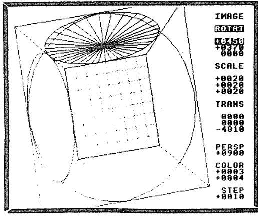
The model-creation section of the Graphics Development System is very effective but it is somewhat eclipsed by the other half of the package - the programs to help you make some use of the models you have created. These consist of two types. Firstly there are the machine code routines that can operate on the data that represents the 3D co-ordinates of an object. These are all bundled together in one group in a variety of formats for use with different systems - cassette, disc, or second processor - and in different memory positions.

To access these routines there is provided the beginnings of a Basic program with procedures already defined to access the machine code. To display the image of an object, then, you just load the machine code, load the data, and run the Basic program suitably amended to include calls to the procedures required (PROCgcol, PROCtrans, PROCrot, PROCproject, etc.).

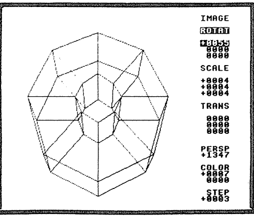
Objects can be animated in this way and several demonstrations of this are provided on the disc. However, better animation can be achieved by resorting to assembler to access the 3D routines rather than Basic. This again is fairly simple as it is really just a case of calling up the routines when needed. However, really fast animation can be achieved by using another feature of the Graphics Development System.

The Beeb's OS can plot lines on the screen at the rate of about 9,ØØØ pixels per second. The Graphics Development System contains its own line drawing routines that will draw at a rate of 3Ø,ØØØ pixels a second. Although it has some restrictions (pixel co-ordinates, GCOLØ type plotting only, no windows or redefinable origin, and it will not work with second processor) the speed difference can be staggering. A marvellous Elite-type space craft performs a complex turning manoeuvre without a suggestion of flicker in one of the demonstrations. (We have included this demo on this month's magazine cassette/disc so you can see the effect to the full).

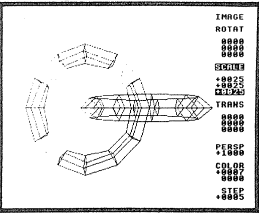
All this takes some getting to grips with. The manual is adequate but better tutorials could be given. However, if you are looking for a package that will really take your Beeb to the limits of graphics and animation for a reasonable price, then the Graphics Development System delivers the goods.



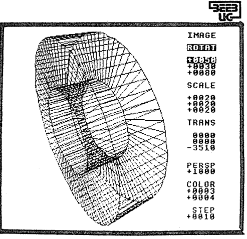
```
IMAGE
ROTAT
+0450
+0370
 0000
SCALE
+0020
+0020
+0020
TRANS
 0000
 0000
-4810
PERSP
+0900
COLOR
+0003
+0004
STEP
+0010
```



```
IMAGE
ROTAT
+0055
 0000
 0000
SCALE
+0004
+0004
+0004
TRANS
 0000
 0000
 0000
PERSP
+1347
COLOR
+0007
 0000
STEP
+0003
```



```
IMAGE
ROTAT
 0000
 0000
 0000
SCALE
+0025
+0025
+0025
TRANS
 0000
 0000
 0000
PERSP
+1000
COLOR
+0007
 0000
STEP
+0005
```



```
IMAGE
ROTAT
+0050
+0030
+0080
SCALE
+0020
+0020
+0020
TRANS
 0000
 0000
-3510
PERSP
+1000
COLOR
+0003
+0004
STEP
+0010
```

Dynamic memory display

The BBC micro's event system provides Sebastian Lazareno with a dynamic and fast moving window onto the Beeb's inner workings.

Many readers will appreciate the value of a static memory dump, giving a snapshot of the micro's memory, but such a display does not show changes in RAM as they occur. This program provides a dynamic memory window, allowing you to observe the Beeb's memory in real time while using the keyboard, running other programs (including ROMs), or just 'resting'. The program can also display the contents of any sideways ROM, and is particularly useful for monitoring sideways RAM.

PROGRAM ENTRY

Save the program before you run it, since an error may cause the computer to hang. Cassette users should alter the value of 'code' in line 160 to &C00.

USING THE PROGRAM

When you RUN the Basic program you will be asked for the start address of the display (i.e. to display zero page, enter 0). The default display start is at PAGE.

The main routine extends through page 9 without intruding into page 10 (Toolkit workspace). It displays the contents of 48 bytes of memory in 'dump' format at the top of a mode 7 screen. You can scroll through memory using the vertical cursor keys with Shift (slow) or Ctrl-Shift (fast). The routine is called 50 times a second by the 'start of vertical sync' event. According to the Advanced User Guide (p.288) such routines should not last more than about 2 milliseconds. This one takes about 5.2 milliseconds, but problems are avoided as the full routine is called by only a proportion of the events generated.

The number of events per complete execution is initially set to 5 but can be altered using Shift along with the horizontal cursor keys. The ratio is shown at the bottom of the display, together with the number of events since the last execution and the display start address. The whole routine can be turned off using function key f7 and turned on using f8. This disables and enables the event generation.

POSSIBLE USES

1.) ROMS. Look at locations starting at &8000 and type *HELP: by stopping the display with Shift+Ctrl you can explore any of your active ROMs. With a small modification to the program (see below) the contents of any sideways ROM or RAM can be continuously displayed.

2.) Entering a line of Basic. Look at locations &700 onward (the Basic keyboard buffer), enter a line of Basic and see the line being tokenized when you press Return.

3.) Keyboard entry. Still looking at &700 onward, use the Copy key to enter a few characters. Note that the characters entering the buffer are the ones you copied, and this is still so if Shift or Ctrl are pressed with Copy. Now look at &3E0 onward (the keyboard buffer). Note that when you Copy a character, the code entering the buffer is that for the Copy key, rather than the character copied, and that pressing Shift or Ctrl together with Copy causes a different code to enter the buffer. Now look at locations &EC and &ED (which contain 'current keys pressed' information) and see that pressing Copy produces yet another code (the 'INKEY' code), which is not affected by pressing Shift and Ctrl. These observations provide some clues as to how the Beeb processes keyboard input, and suggest further questions.

PROGRAM MODIFICATIONS

The program displays both Teletext character sets, i.e. ASCII 32-126 and 160-255, so many displayed characters are ambiguous. To display only the normal set delete line 670.

Sideways ROMs/RAM can be selected with the following modifications. Change 'lock' to 'rom' in lines 390 and 410 and insert the following lines:

```
 150 rom=base+2:key=base+3:lock=base-1
 500 .start LDA rom:STA&FE30
 505 LDA#0:STA&308
1015 LDA&F4:STA&FE30
```

To inspect your RAM (or ROM), set the display to start at &8ØØØ and press Shift with the horizontal cursor keys to select the required ROM.

As it stands, the routine monopolizes the event vector. The program could be modified to save the previous contents of the vector and exit via a JMP so that any other user routines using this event can take their turn.

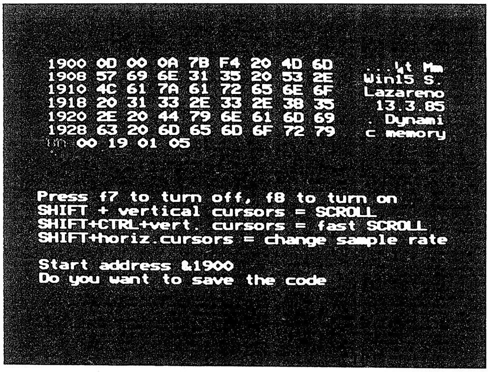
PROGRAM NOTES

Lines 22Ø-8ØØ: checks number of event, keys pressed, sets up pointers, text window.

Lines 81Ø-112Ø: the display routine.

Lines 113Ø-122Ø: sets up soft keys. Clears buffer if necessary using equivalent of ON ERROR to detect full soft key buffer. Starts routine by invoking f8.

```
   1Ø REM PROGRAM MEMWIN
   2Ø REM VERSION BØ.7
   3Ø REM AUTHOR  S.LAZARENO
   4Ø REM BEEBUG  OCT 85
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ *FX13,4
  11Ø MODE7
  12Ø PRINT'"Press f7 to turn off, f8 to
 turn on"'"SHIFT + vertical cursors = SC
ROLL"'"SHIFT+CTRL+vert. cursors = fast S
CROLL"'"SHIFT+horiz.cursors = change sam
ple rate"
  13Ø ON ERROR GOTO 16ØØ
  14Ø base=&8A:ad=base:INPUT"Start addre
ss &"S$:!ad=EVAL("&"+S$)
  15Ø key=base+2:lock=base+3
  16Ø line=base+4:code=&8CØ:?key=Ø:?lock
=5
  17Ø REM EVENTV points to routine
  18Ø ?&22Ø=code MOD &1ØØ:?&221=code DIV
 &1ØØ
  19Ø FORI%=ØTO2STEP2
  2ØØ P%=code
  21Ø [OPT I%:PHP:PHA:TXA:PHA:TYA:PHA
  22Ø LDX key:INX      \Increment
  23Ø STX key:CPX lock  \key,
  24Ø BEQ scrollcheck
  25Ø LDX #24Ø:JMP end  \exit if < lock
  26Ø .scrollcheck      \is SHIFT or
  27Ø LDA &25A:TAX      \SHIFT LOCK
  28Ø AND #8:BNE ctrl   \pressed?
  29Ø TXA:AND #32
  3ØØ BNE start         \If so, check
  31Ø .ctrl TXA:AND #&4Ø\for CTRL,set
  32Ø LSR A:BNE scan    \acc. to alter
  33Ø LDA #8            \address pointer
```



```
1900 0D 00 0A 7B F4 20 4D 6D   ...¼t Mm
1908 57 69 6E 31 35 20 53 2E   Win15 S.
1910 4C 61 7A 61 72 65 6E 6F   Lazareno
1918 20 31 33 2E 33 2E 38 35    13.3.85
1920 2E 20 44 79 6E 61 6D 69   . Dynami
1928 63 20 6D 65 6D 6F 72 79   c memory
8n 00 19 01 05


Press f7 to turn off, f8 to turn on
SHIFT + vertical cursors = SCROLL
SHIFT+CTRL+vert. cursors = fast SCROLL
SHIFT+horiz.cursors = change sample rate

Start address &1900
Do you want to save the code
```

```
  34Ø .scan LDX &EC      \Check whether
  35Ø BEQ skip           \a cursor
  36Ø PHA:SEC:TXA        \key has been
  37Ø SBC #128:TAX:PLA   \pressed,
  38Ø CPX #25:BNE right \if so,
  39Ø DEC lock:JMP start\take
  4ØØ .right CPX #121    \appropriate
  41Ø BNE up:INC lock    \action.
  42Ø JMP start
  43Ø .up CPX #57:BNE down
  44Ø CLC:ADC ad:STA ad
  45Ø BCC skip:INC ad+1
  46Ø .skip JMP start
  47Ø .down CPX #41:BNE start
  48Ø EOR #&FF:ADC ad::STA ad
  49Ø BCS start:DEC ad+1
  5ØØ .start LDA #Ø:STA &3Ø8
  51Ø LDA #24:STA &3Ø9   \Equivalent
  52Ø LDA #39:STA &3ØA   \to
  53Ø LDA #8:STA &3ØB    \VDU28,Ø,24,39,8
  54Ø ORA &DØ:STA &DØ    \Text window set
  55Ø LDA #1:STA line
  56Ø LDY #Ø:LDX #Ø:STX key\Print addr.
  57Ø .loopad:JSR printad  \and contents
  58Ø .loop:LDA (ad),Y     \of next 8
  59Ø JSR printacc         \locations,
  6ØØ LDA #32:JSR prinx    \first in
  61Ø INY:ASL line:BCC loop\hex,
  62Ø ROL line:LDA #131:JSRprinx:INX
  63Ø TYA:SEC:SBC #8:TAY
  64Ø .loop2:LDA (ad),Y    \then ASCII
  65Ø CMP #32:BCC dot      \if possible.
  66Ø CMP #127:BCC pound
  67Ø CMP #16Ø:BCS next    \ Remove this
 line to prevent display of chars 16Ø -
255
  68Ø .dot LDA #ASC".":BNE next
  69Ø .pound:CMP#96:BNE hash \MODE 7
  7ØØ LDA#35:BNE next        \display
  71Ø .hash CMP#35:BNE under \of
  72Ø LDA#95:BNE next        \these
  73Ø .under:CMP#95:BNE next \is
  74Ø LDA #96                \strange
 75Ø .next JSR prinx
 76Ø INY:ASL line        \Display 8
 77Ø BCC loop2:ROL line  \locations
 78Ø CPX #24Ø:BCS end    \per line,
 79Ø JMP loopad          \6 lines.
 8ØØ \
 81Ø .printhex:AND #&F   \Print
 82Ø CLC:ADC #&3Ø:CMP #&3A\digit in
 83Ø BCC prinx:ADC #6    \hex
 84Ø .prinx:STA &7CØØ,X  \increment
 85Ø INX:RTS             \screen point
er
 86Ø \
 87Ø .printacc:PHA       \Print
 88Ø ROR A:ROR A         \contents of
 89Ø ROR A:ROR A         \accumulator
 9ØØ JSR printhex:PLA    \in hex.
 91Ø JSR printhex:RTS
 92Ø \
 93Ø .printad:LDA #134:JSR prinx
 94Ø TYA:CLC:ADC ad:PHA  \Print
 95Ø LDA ad+1:ADC #Ø     \2 byte
 96Ø JSR printacc        \address,
 97Ø PLA:JSR printacc    \increment
 98Ø LDA #135:JSR prinx  \screen
 99Ø RTS                 \pointer.
1ØØØ \
1Ø1Ø .end
1Ø2Ø LDA #133:JSR prinx  \This routine
1Ø3Ø LDA #base           \executed
1Ø4Ø JSR printacc        \at every
1Ø5Ø LDA #135:JSR prinx  \call: prints
1Ø6Ø LDY #Ø              \the start
1Ø7Ø .loop:LDA base,Y    \address
1Ø8Ø JSR printacc        \of the
1Ø9Ø LDA #32:JSR prinx   \display, and
11ØØ INY:CPY #4:BNE loop \contents of
111Ø PLA:TAY:PLA:TAX     \lock & key
112Ø PLA:PLP:RTS         \then exits.
113Ø .invoke
114Ø OPT FNonerror
115Ø OPT FNturnoff       \Set f7
116Ø OPT FNturnon        \Set f8
117Ø OPT FNbreak         \Set f1Ø
118Ø LDA line:STA&2Ø2    \ON ERROR
119Ø LDAline+1:STA&2Ø3   \OFF
12ØØ LDA#138:LDX#Ø       \Invoke f8 to
121Ø LDY#136:JSR&FFF4    \turn on rout
ine
122Ø RTS:]
123Ø NEXT
124Ø CALLinvoke:END
125Ø DEF FNonerror:pass=P%
126Ø FORopt=ØTO2STEP2:P%=pass
127Ø [OPT opt
128Ø LDA&2Ø2:STA line
129Ø LDA&2Ø3:STA line+1
13ØØ LDA #(onerr MOD 256):STA &2Ø2
131Ø LDA #(onerr DIV 256):STA &2Ø3
132Ø BNE setup
133Ø .onerr:LDA#18:JSR&FFF4
134Ø .setup:]
135Ø NEXT:=I%
136Ø :
137Ø DEFFNturnoff:str$="K.7*FX13,4 M":=
FNoscli
138Ø :
139Ø DEFFNturnon:str$="K.8MO.7:V.31,Ø,8
:*FX14,4 M":=FNoscli
14ØØ :
141Ø DEFFNbreak:str$="K.1ØO. M?&22Ø=&"+
STR$~(code MOD256)+":?&221=&"+STR$~(code
 DIV256)+" M":=FNoscli
142Ø :
143Ø DEFFNoscli
144Ø pass=P%:FORopt=Ø TO2 STEP2
145Ø P%=pass
146Ø [OPT opt
147Ø LDX#(string MOD256)
148Ø LDY#(string DIV256)
149Ø BNEoscli
15ØØ .string
151Ø ]
152Ø $P%=str$
153Ø P%=P%+LEN$P%+1
154Ø [OPT opt
155Ø .oscli JSR&FFF7
156Ø ]
157Ø NEXT
158Ø =I%
159Ø :
16ØØ ON ERROR OFF:MODE 7
161Ø IF ERR=17 END
162Ø REPORT:PRINT" at line ";ERL
163Ø END
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS

MUSIC SYSTEM LINE FEEDS - Allan Woods

Island Logic's Music System was said in our review (BEEBUG Vol.4 No.1) to produce an extra linefeed at each line on printing out music (if the printer was configured in the normal way). The @ key switches off this effect but is not documented.

Z8Ø FILENAMES - Michael Colley

Filenames starting with character 152 (conceal display in mode 7) operate perfectly on the unadorned Beeb. They are not recognised with a Z8Ø second processor.

BOFFIN

Boffin is an exciting new game from Addictive Games. Geoff Bains, BEEBUG's resident 'boffin', dons his white coat and reports on the game and on an exclusive BEEBUG competition.

Boffin is the second release for the BBC micro from Addictive Games and seems set to follow 'Football Manager' as a chart topper. The boffin of the title is a cute little mad professor armed only with an umbrella and an insatiable desire to wipe out horseshoes and collect pieces of laboratory apparatus. The horseshoes and the bits of glassware are scattered around the inevitable caverns. You must guide your boffin around the caverns, destroying the unlucky horseshoes and picking up bonuses in the form of the apparatus.

The caverns are filled with the usual array of platforms, barbed wire, perilous drops and trampolines. The latter are especially effective, giving a beautiful boing noise and actually flexing when they're jumped on. The umbrella is not just carried for show. It will stop the professor's fall and can even be used to pick up the objects.

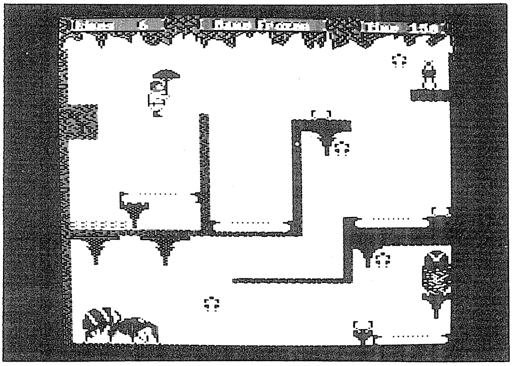
Also in the caverns, however, is a superb assortment of suitably evil creatures - gigantic spiders, giant puffers, manta bats, etc. The movement of the spider is excellent, given the size of it, and the game is better than most when considering playability. The collision detection is very good too. You don't die when you are still inches away from your foe as happens in many other games. This is especially important with the spider as you have to get pretty close to that, at times, to win.

There is a high score table which records the number of the game currently being played and the game comes in two versions. Boffin I is Beeb only (B and B+) and boasts 45 caverns. Boffin II has only (only!) 25 caverns but it is compatible with both the Beeb and Electron.

Overall, the graphics are good, fast and smooth and the whole theme makes it very enjoyable. The mad professor with his umbrella is a great idea, and as an example of the Manic Miner type of caverns game, Boffin shows a wonderful sense of humour and is strongly recommended.

MPETITION COMPETITION COMPETITION COMPETITION COMPETITIO

To celebrate the launch of Boffin, Addictive Games are running a competition exclusively for BEEBUG members. Each month Addictive are giving £50 to the highest scorer in Boffin II. To make things even easier for members there is a special price on the game as well. For BEEBUG members the excellent Boffin costs only £8.95, that's £1 off the normal price. Send your high scores to Addictive at:



BEEBUG Competition,
Addictive Games,
7a, Richmond Hill,
Bournemouth,
BH2 6HE.

You must quote your BEEBUG membership number and include a statement from a witness. Should you be the top scorer, you will be expected to reproduce a similar score at Addictive's offices.

Each month, the winner of the Boffin High Scores competition will be invited to Addictive Games to challenge Paul O'Mally, the author of Boffin, at his own game. The first reader skilful enough to beat him will win £250 worth of peripherals and add-ons, of his choice, for his computer. The results of the competition will be published in future issues of BEEBUG.

Wordwise Plus

In the first of three articles, Stephen Ibbs, an enthusiast for the new Wordwise Plus, introduces the whole idea of writing Wordwise Plus programs.

There are now over 20,000 Wordwise Plus ROMs in use. Perhaps the most exciting aspect of this ROM is the special programming language built into it. This enables routines to be written to control the word processor. In addition to the main text area there are ten 'segment' areas each of which can store text, or routines which are called by pressing Shift and one of the red function keys. The excellent reference manual gives a few examples but more complex routines can be easily written once you have a basic grasp of the language. This short series of three articles is intended to provide that initial understanding. Some useful hints will be given, and the text will be illustrated with worked examples.

It is first necessary to understand that the commands of the language can either be written as a program, in a segment, or typed directly from the main menu preceded by a colon. Thus if we wanted to move the cursor to the bottom of the text it could be done by executing the following in a segment:

```
SELECT TEXT
CURSOR BOTTOM
DISPLAY
```

or by typing :CURSOR BOTTOM <Return> from the menu. Any colon command must end with a <Return> but this will not normally be shown in these articles. Similarly you will not normally be reminded that a colon command must be typed from the menu page.

The ability to issue commands from the menu is very useful when you are writing routines. You have absolute control and can check the effect of each command as it is issued.

HINT: Try issuing commands (except REPEAT loops etc) one at a time from the menu if you are having trouble fault-finding.

Probably the most often-used command is CURSOR, followed by another instruction. Place some sample text into the main area, go back to the menu and type:

```
:CURSOR BOTTOM
:CURSOR UP 2
:CURSOR LEFT 5
```

Go back to the text, and you will see just how precisely the cursor position can be controlled. When moving it to a particular position on a line, the CURSOR AT command is used, thus CURSOR AT 10 will place the flashing cursor underneath the tenth character. The only, but important, problem is with CURSOR AT 39. Because of the 'wrap-around' effect that stops words being split over lines, a blank area may be left on the right hand edge of a line on the screen. CURSOR AT 39 places the cursor under the last actual character (which may of course be a space or a Return character) and not necessarily on the rightmost character position on the screen.

HINT: CURSOR AT 39 places the flashing cursor under the last character, whether at position 39 or not.

As a very simple first example, let us count how many characters a particular line has. To do this we will use the GCT$ command (Get Character from Text). The character 'got' will be the one above the cursor, and the cursor will then move one position to the right. Never forget that final part. This, at different times, can be both invaluable and a nuisance.

Enter segment 0 and type in the following lines (ending each with Return):

```
SELECT TEXT
CURSOR AT 0
A%=0
REPEAT
A$=GCT$
A%=A%+1
UNTIL A$=CHR$13
DISPLAY
```

Go back to the main text area and clear it. Type in a line of 34 characters then press Return and then a few more characters. Place the cursor anywhere on the upper line, and press Shift-fØ to run the segment program. After a brief pause you will see the cursor move to the start of the 2nd line. If you now (from the menu) type :PRINT A% the answer 35 (the count includes the Return at the end of the line) will be displayed.

What has happened? The cursor has started from the beginning of the line and repeatedly placed a GCT$ character into the variable, A$, and increased A% by 1. You can't tell it to stop the loop when it finds a Return character, as such, but by finding Return's ASCII code of 13. Hence the line, UNTIL A$=CHR$13. If you want to test this, type :PRINT ASC(A$) from the menu and you will see the number 13 displayed. Similarly f1 (green embedded start) is CHR$2 and f2 (white embedded end) is CHR$7. So if you want to count characters until the cursor encounters the start of an embedded command, it can be easily done by changing the line to:

```
    UNTIL A$=CHR$2.
```

HINT: In situations where you can't use |R for the Return try CHR$13. Similarly use CHR$2 for f1 instead of |G and CHR$7 for f2 instead of |W

When the routine finds the Return character it adds one more to A% (giving 35) then stops. Finally the DISPLAY puts us back into edit mode.

The technique of collecting characters using REPEAT-UNTIL loops is relatively slow, but is used time and again in routines. Try inserting a Z into the upper line of text, and altering the appropriate line to UNTIL A$="Z" (note firstly the quotation marks, and secondly that if you use a lower case z it will not work. To trap either you would put UNTIL A$="Z" OR A$="z").

Whilst the above example worked, it would be extremely tedious to have to type a Return at the end of every line to be counted. Fortunately there is a little known memory location that can spring to our aid. ?&7E holds the current horizontal cursor position. Place the cursor anywhere on a text line, then type (from the menu):

```
:CURSOR AT 39
:PRINT ?&7E
```

and the amount returned will be the actual final character position, which may not necessarily be 39. This memory location is extremely useful and is used time and again. However, it is worth noting that this is not a very 'legal' method. There is no guarantee that this location will have the same function in future issues of Wordwise Plus.

HINT: Calling ?&7E memory location will return a value equivalent to the horizontal position of the cursor.

If you want to see the routine running, insert the command DISPLAY after the A$=GCT$ line. You will see the cursor moving along as it goes through the REPEAT loop, stopping when it finds the Return. To slow it down even more add the line IF GET THEN just above the A%=A%+1. This will 'single-step' the routine. Press the space bar each time the screen freezes.

HINT: when developing routines, the lines
DISPLAY
IF GET THEN
will allow examination of the cursor position and the routine's effect so far, before continuing.

The simple routine discussed so far could be used to transfer text very slowly from the main area to, say, segment 7 by adding a couple more lines so that the routine looks like this:

```
SELECT TEXT
REPEAT
A$=GCT$
SELECT SEGMENT 7
TYPE A$
SELECT TEXT
UNTIL EOT
CURSOR TOP
DISPLAY
```

Note the EOT command - 'End of Text', to end the REPEAT loop - and the CURSOR TOP command, which restores you to the top of the text in the main area.

There is, however, another command associated with getting characters from text, GLT$ (Get Line from Text). This will place as many characters as possible into a variable up to either the first Return encountered, or until the variable has 255 characters in it. This means that the transfer of text mentioned above can be achieved in a fraction of the time using the following routine:

```
SELECT TEXT
REPEAT
A$=GLT$
SELECT SEGMENT 7
TYPE A$
SELECT TEXT
UNTIL EOT
DISPLAY
END
```

If you run this routine, you will find the resulting text in segment 7 is a little strange - all the Returns from the original are missing from the copy. This is because GLT$ stops at a Return (or after 255 characters). We can account for this by inserting the following after the TYPE A$ line:

```
IF LEN(A$)<255 THEN TYPE "|R"
```

If GLT$ stops before 255 it is because of a Return. A Return is inserted in the copy (with the |R) if the length of A$ is less than 255. The only case where this might not be accurate would be the very final line of the text. If this was important the following could be inserted after the UNTIL EOT line:

```
SELECT SEGMENT 7
IF LEN(A$)<255 THEN DELETE LEFT
SELECT TEXT
```

HINT: When using GLT$, remember that it does not collect the Return character which may need to be inserted separately.

As it stands, the transfer will commence from the current cursor position. If you want the entire text copied, insert as the 2nd line:

```
CURSOR TOP
```

so now the routine would look like this:

```
SELECT TEXT
CURSOR TOP
REPEAT
A$=GLT$
SELECT SEGMENT 7
TYPE A$
IF LEN(A$)<255 THEN TYPE "|R"
SELECT TEXT
UNTIL EOT
SELECT SEGMENT 7
IF LEN(A$)<255 THEN DELETE LEFT
SELECT TEXT
CURSOR TOP
A$=""
DISPLAY
END
```

The A$="" line illustrates good practice - to empty any string variables used. This is important, particularly with the GLT$ command. There are only just over 600 bytes of memory available for variables, and this amount can be very quickly filled, in which case the GLT$ command won't work, and the error message NO $ ROOM will appear. If you experience this, then you must clear the offending variables by typing, say, :D$="" from the menu.

HINT: Always try to clear variables, particularly $ variables, after use, to keep memory space empty.

Finally, for this month, let's use the ?&7E location information to delete a complete line, with the following routine:

```
SELECT TEXT
CURSOR AT 39
DELETE LEFT ?&7E
DELETE AT
DISPLAY
```

The cursor is placed on the rightmost character and ?&7E returns a value of, say, 34 so then 34 characters are deleted to the left of the cursor. The character above the cursor is still there, so this is removed with DELETE AT.

As an exercise for next month, you may like to alter this routine a little. How could the routine be developed so that, after deleting the text, the routine checks to ensure that the two lines now joined (above and below the line deleted) are separated by one space? Without this modification you will find that sometimes the lines are joined together with either no space, or two spaces. Think about that one for now. I will give you the answer in the next issue.

— HINTS — for Wordwise Plus users

During the course of development of BEEBUGSOFT's Wordease, many useful hints and tips were discovered. We bring some of these together here, for expert and beginner alike.

VDU CODES

Wordwise Plus accepts VDU codes in a very similar way to Basic. They may be strung together in the normal way and used to print out characters on the screen that could not otherwise be produced. Try the following segment program.

```
    CLS
    DOTHIS
    VDU 129,157,131,141
    PRINT "Double height in colour."
    TIMES 2
    IF GET THEN END
```

This will clear the screen and print a double height banner in yellow on a red background.

VDU 31 can be used to very good effect in a similar manner to position text on the screen. VDU31 is equivalent to the Basic TAB(x,y). For example:

```
    CLS
    X%=1
    REPEAT
    VDU31,X%*3,X%*2,134
    PRINT "hello"
    X%=X%+2
    UNTIL X%>10
    IF GET THEN END
```

However, be very careful not to execute a VDU 22 (select screen mode) command from Wordwise Plus. This can cause all your text and programs to be lost.

SOUND

Sounds can be produced from within Wordwise Plus using the VDU7 (beep) command. The pitch and duration of the note can be changed using *FX commands, 213 and 214. For example, to produce a one-second high-pitched note use:

```
    *FX213,200
    *FX214,20
    VDU7
```

To control the pitch or the duration with a variable, OSCLI must be used:

```
    N%=0
    REPEAT
    OSCLI"FX213,"+str$(N%)
    *FX214,2
    VDU7
    N%=N%+5
    UNTIL N%>250
```

The voice and the amplitude/envelope of the note can also be changed using *FX211 and *FX212, respectively. However, note that you cannot alter envelopes from within Wordwise Plus so if you select an envelope (using *FX212) then the envelope must be defined from Basic before entering Wordwise Plus.

FILENAME PRINTING

To automatically print the last-used filename within your text, use the following embedded command:

```
    <f1> PS F$ <f2>
```

Where <f1> is the green code produced by pressing function key 1. This will print the filename when the text is either previewed or printed.

LABEL DUPLICATION

It is easy to duplicate label or procedure names when writing a long segment program, with disastrous results. To avoid a clash, test a proposed name before using as follows:

Press Escape to leave you in the menu of the segment containing your development program

Type :GOTO newname

If the error message, 'No such label' is produced then the name is okay. Otherwise it has already been used.

WORDWISE HELP AND PRIZE COMPETITION

The Wordwise Plus language is a fascinating one to use, and we hope that these hints and the series of articles that we are currently running tempt you to experiment with it.

To assist you in your explorations we have set up a panel of 'experts' who will try to answer any problems you have with Wordwise Plus. We will feature a selection of your questions (with answers!) in coming issues, and we are offering a prize of £50 for the most novel Wordwise Plus program submitted before November 30th.

1st course

Print Formatting (Part 1)

Correctly formatting numbers and text on the screen often causes problems for newcomers to programming. In the first of two articles on this subject, J. Pike looks at the help provided by BBC Basic.

Whether it is on the screen or to a printer, the correct formatting of data can often seem to be a hit or miss affair. The solution would seem to be the use of the special variable @% but a first reading of the User Guide about this can be most off-putting. However, the use of @% has much to offer when properly understood.

The explanation of format control using @% on pages 7Ø and 325 of The User Guide gives a good introduction to the facilities available, but it is incomplete. You will be surprised by some of the unexpected, and usually undesirable, formats which crop up. In addition determining the @% value to give a particular format can be very difficult.

USING @%

The format control variable @% is a special variable that controls the printing (or display) of numbers. @% is a four byte integer similar to the 'resident' integers A% to Z%. Its value remains unchanged by running a program or editing, but it is reset to a default value of &9ØA when Break is pressed. (In Basic I this default value of @% is &AØA. However, this acts the same as @%=&9ØA, the first A being interpreted as 9 because it is not in the permitted range of 1 to 9).

Byte 4 (B4) STR$	Byte 3 (B3) Type	Byte 2 (B2) Digits	Byte 1 (B1) Field width
00 Default	00 General	01-09 Digits	00-FF Width
01 @%	01 Exponential	01-09 Decimals +1	
	02 Fixed	00-09 Decimals	

Each of the four bytes that make up @% controls a particular function of print formatting. These are shown in the table and will be considered individually.

BYTE 1

The B1 byte determines the field width for printing a number. The print line is divided into fields B1 characters wide into which successive numbers are printed. Numbers and strings have different rules governing where they are printed.

Numbers are printed right justified in the next unused field, whereas strings ignore the fields and are printed consecutively. For example, with a field width of 5 (@%=&5):

```
PRINT 1 2 3
PRINT "1" "2" "3"
    |    |    |
    1    2    3
123
```

(| denotes the (invisible) field boundaries).

Numbers can be made to print like strings by using a semi-colon as a separator. However, the first number is still printed to be right justified in the first field. A semi-colon has no effect on the strings.

```
PRINT 1;2;3
PRINT "1";"2";"3"
    |    |    |
    123
123
```

Similarly, strings can be forced into the next free field with a comma as a separator, though they are still left justified. This has no effect on the numbers.

```
PRINT 1,2,3
PRINT "1","2","3"
    |    |    |
    1    2    3
1    2    3
```

All the numbers can be made to print as do strings by preceding the whole line with a semi-colon. Any other separator following (like a comma) will cancel this effect. Again, the semi-colon has no effect on the strings but the comma forces the last string into the next field.

```
PRINT ;1 2 3,4
PRINT ;"1" "2" "3","4"
    |    |    |
123      4
123  4
```

A comma preceding all the items to be printed is ignored by both numbers and strings.

```
PRINT ,1 2 3
PRINT ,"1" "2" "3"
    |    |    |
    1    2    3
123
```

Problems arise when a number uses more characters in its representation than there are in the field width. Then the field is completely filled and the extra characters overflow into the next field. Thus even one too many characters causes all the subsequent printing to be one field further right. This can cause havoc when setting out numbers in tables.

One way of preventing this is to use the TAB command. TAB(X) positions the cursor X characters along the current line and accounts for the Return when the line exceeds the screen width. Numbers following TAB(X) are printed starting at position X but can be made to print in the next field by following TAB(X) with a comma. Problems can arise with TAB(X) when X<POS. It is better then to use TAB(X,Y) which moves directly to position X,Y. It then re-allocates the print fields starting at the current position and, unlike TAB(X), prints the next number right justified in the first field.

BYTE 4

The B4 byte is of no interest unless the STR$ function is to be used. STR$ converts a number into an equivalent string of characters. If the B4 byte of @% is set (B4=1), then the string created will follow the format currently defined by the other bytes of @%. If B4 is zero then the string created will follow the default format regardless of the current value of the other @% bytes.

BYTE 3

B3 controls the format in which a number is printed. A General, Exponential or Fixed format is used depending on whether B3 is Ø, 1 or 2. Values of B3 greater than 2 give a general format.

GENERAL E.g.: 1Ø.

The general format aims to give a reasonable representation of any number. This is the default format. Numbers below Ø.1 are printed in exponential form.

EXPONENTIAL E.g.: 1E1.

The number is expressed as a multiple of a power of ten. The first number (the 'mantissa') is the multiple. The number following the E (the 'exponent') is the power of ten. 2.3E-1 is therefore equivalent to Ø.23.

FIXED E.g.: 1Ø.ØØ.

The number is printed with a specified number of decimal places and with the decimal point in a fixed position.

BYTE 2

The B2 byte controls the accuracy to which the number is printed. In General and Exponential formats, this byte controls the maximum number of significant figures to which a number is printed. In General format the number of significant figures is the same as the total number of digits in the number. In Exponential format B2 is similarly equal to the number of digits in the mantissa. In General or Exponential format values outside the range 1 to 9 default to 9.

Although B2 influences the length of the number, it is with the accuracy that this byte is primarily concerned. When B2 is larger than the number of digits needed to represent the number then less digits are printed. For example, for all values of B2 greater than 2, the number 12.6 is printed as 12.6 in General format and 1.26E1 in Exponential format. When B2 is less than the number of significant figures the number is truncated. If B2=2, then 12.6 is printed as 13 or 1.3E1, with the least significant figure being rounded as appropriate. If B2=1 then it is not possible to represent 12.6 in General format, and it is then printed as 1E1 in General format and 1.E1 in Exponential format.

For numbers that are either very large or very small, the General format reverts to Exponential. This can cause the number of characters used to represent the number to be larger than anticipated. For example, with the default format (@%=&9ØA), we might expect a maximum of 11 characters for any number (9 digits plus the decimal point and negative sign). However, a very small negative number will revert to Exponential format and be printed with many more (9 significant figures, a negative sign, the 'E', the exponent and the sign of the exponent!). It can easily take up 15 characters.

This can have serious implications when controlling the positioning of numbers using the B1 field width byte in, say, a table. Numbers can easily overflow their fields.

To cause, say, Ø.Ø1 to be printed as Ø.Ø1 and not as 1E-2 we use the Fixed format. In Fixed format, B2 controls the number of decimal places. Unlike General and Exponential formats, however, trailing zeros are not suppressed in Fixed format. With B2=2 the number 1 is printed as 1.ØØ. If the number requires more than 9 decimal places the number of decimal places reduces until there are 9 digits, and the number prints in the default format. As for the General format, very large numbers revert to Exponential format, using a maximum of 14 characters. The range of B2 in Fixed format is Ø to 9. B2=Ø causes numbers to be followed by a decimal point but no decimal digits!

The correct use of @% will enable you to deal with just about any situation requiring the tidy presentation of numbers. However, there is a need for a printing format which is more flexible, easier to specify, and which cannot overflow its allotted space. In part 2, we shall look at a format with these and other useful properties, similar to the PRINT USING command to be found in other versions of Basic.

POINTS ARISING POINTS ARISING POINTS ARISING POINTS

VIEW PRINTER DRIVER GENERATOR (BEEBUG Vol. 4 No.3)

As a result of renumbering, the line numbers of the '£' patch printed in the magazine are out by 1Ø and should be numbered from 12Ø2 to 12Ø8 and not as printed. The coding itself is quite correct.

PRETTY LIST (BEEBUG Vol.3 No.9)

Mr.N.Smith of Stoke-on-Trent has found that when this utility is used with assembler programs containing comments following a "/", the text drifts to the right of the page or screen. This apparently results when the utility finds a character which corresponds to a Basic token. Mr.Smith offers the following to prevent this:

```
  2Ø5 IF C%=&5C THEN PROCbslash
    :
 167Ø DEF PROCbslash
 168Ø REPEAT:C%=?I%
 169Ø IF C%=&5C AND ?(I%-4)<>&D PRINT'SPC5;:FOR J%=1 TO S%:PRINT SPC1;:NEXT J%
 17ØØ IF C%<&8Ø PRINT CHR$(C%); ELSE A%=C%:CALL token
 171Ø I%=I%+1:UNTIL C%=&D
 172Ø ENDPROC
```

ASSEMBLER ARITHMETIC (BEEBUG Vol.3 No.8)

John Bank of Ang's Hotel, Brunei (yes, BEEBUG goes everywhere) has discovered that this program does not correctly handle a zero result if the destination variable is real, nor does it detect division by zero. The problems occur in the two sign routines sig1 and sig2, which do not recognise the special format of a floating point zero (five zero bytes). It can be corrected with the following changes:

```
 2Ø8Ø .sig1:LDX#4:LDA#Ø:.detzero:ORAfac1,X:DEX:BPLdetzero:TAX:BEQzero
 2Ø85 LDA#8Ø:EORfac1-2:AND#&8Ø:EORfac1+1:STAfac1+1:.zero:RTS
 2Ø9Ø .sig2:LDX#4:LDA#Ø:.detz2:ORAfac2,X:DEX:BPLdetz2:TAX:BEQzero
 2Ø95 LDA#&8Ø:EORfac2-2:AND#&8Ø:EORfac2+1:STAfac2+1:RTS
```

Eprom Programmer Project (Part 2)

Alan Webster and Geoff Bains continue our project on constructing an EPROM programmer by describing the essential software to program your own EPROMs.

When you have finished building the EPROM programmer hardware (described in the previous issue of BEEBUG) you will need the following program to use it to program your own EPROMs. If you have any sideways software (that you have been using in sideways RAM, for example) then you can 'blow' an EPROM with this straight away. Otherwise you can use the programmer to copy or study existing ROMs. However, note that all commercial ROMs are copyright and should not be copied. We will be looking at writing your own sideways software in the next issue.

As the program is fairly lengthy and it requires 16k of memory space to store an EPROM in RAM, disc users will need to set PAGE to &1200 for this program. Type PAGE=&1200 before you CHAIN the program.

The program is extremely user friendly. However, make sure that the programmer is plugged in correctly before you start. Accompanying the main menu, there will be a 'test state' printed. The 'test state' will inform you of an EPROM's presence in the programmer, if it is blank or not, and the size and name of any non-blank EPROM present. This test state is continuously updated as long as the menu is displayed.

The main menu provides six options:

1. Move the present EPROM contents into memory and save it to cassette or disc under the specified filename.
2. Load the specified file into memory and blow it into EPROM. This takes a little over 7 minutes for an 8k and 14.5 minutes for a 16k EPROM.
3. Display the EPROM in a 'dump' format. Press the Shift Key to continue displaying the EPROM's contents. To exit: press Escape.
4. Verify the present EPROM against a file on cassette or disc. This is most useful to check a programmed EPROM or a saved EPROM file.
5. Leave the EPROM programmer software.

*. Execute a command line (any * command).

To copy an existing ROM or EPROM, then, all you have to do is to insert the ROM to be copied, select option 1 to save the contents to a cassette or disc file, insert a blank EPROM of the correct size, and select option 2, entering the filename used, when prompted. You may like to verify the resulting EPROM against the file saved. Use option 4 for this, entering the filename when asked.

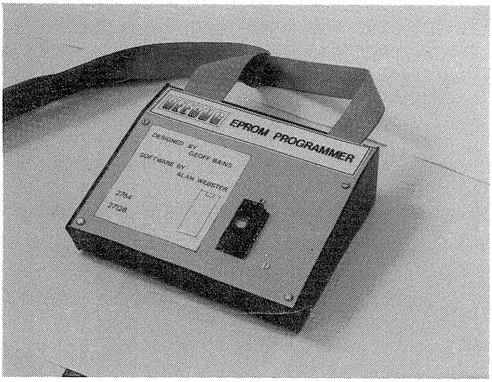
To inspect a ROM or EPROM, insert it carefully in the programmer, select option 3 and study the dump on the screen at leisure. Shift displays the next screenful of the ROM and Escape will return you to the main menu.

That completes the facilities to produce your own EPROMs and to study others. In the next issue of BEEBUG we will look at how to write the sideways ROM programs to fill them with.

<u>PROGRAM NOTES</u>

Lines 140 to 160 form the main program loop. The main procedures are described below in order of appearance.

PROCinit	Set up variables and define break key to de-select 21 volts.
PROCmenu	De-select 21 volts, print menu and wait for input.
PROCtest	Test size and state of EPROM. 0=not present, 1=blank and 2=not blank.
PROCstar	Issue * command.
PROCquit	Select mode 7 and end.
PROCassemble	Set up the machine code (see later for a full description).
PROCsave	Save EPROM to file.
PROCersize	No EPROM to save.
PROCdispsave(x)	If x=0 then move EPROM into memory and execute PROCsave. If x=1 then display EPROM contents.



PROCprogram Program EPROM from file (see later for full explanation).
PROCverify Verify EPROM against file.
PROCromt Display title of EPROM.

ASSEMBLER NOTES

PROCassemble sets up the following variables:

DDRA/DDRB Data Direction Registers at &FE63 / &FE62.
PCR Peripheral Control Register at &FE6C.
IORA/IORB Input / Output Registers at &FE61 and &FE60.
bytes/bytes2 Test State of EPROM buffer.
LO8/HI6 14 bit address stores.

The Input / Output Registers are the two locations (for ports A and B of the User VIA) through which data is passed as input or output, from or to, the programmer. The Data Direction Registers control which direction this data is to go. The Peripheral Control Register is a general register which has just four states which concern us here, as follows:

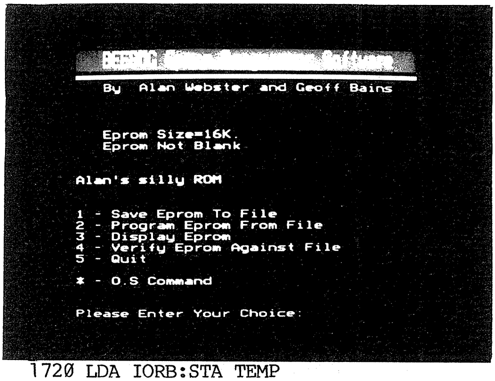
CA2 hi,CB2 lo	1 1 Ø Ø 1 1 1 Ø	= &CE
CA2 hi,CB2 hi	1 1 1 Ø 1 1 1 Ø	= &EE
CA2 lo,CB2 hi	1 1 1 Ø 1 1 Ø Ø	= &EC
CA2 lo,CB2 lo	1 1 Ø Ø 1 1 Ø Ø	= &CC

CA2 is connected to the EPROM's NOT program line and CB2 to the latch of the latch chip (see Aug/Sept BEEBUG for more information on these two lines).

The assembler code can be broken down into the following subroutines:

1650-1730 Read one byte of data.
1740-1780 Read title of EPROM.
1790-1820 Read bytes Ø-3 and &2000-&2003. Store in 'bytes' and 'bytes2'.
bytes=Ø EPROM not present.
bytes=&FFFFFFFF EPROM blank
bytes=bytes2 8k EPROM.
bytes<>bytes2 16k EPROM.
1830-2090 Display/Move EPROM to memory.
?&76=1 EPROM is read and displayed.
?&76=Ø EPROM is read and moved down into memory.
2250-2430 Program EPROM from the data in memory.
2440-2450 Wait routine. Provides a delay of 5Øms to program EPROM.

```
   1Ø REM PROGRAM EPROM PROGRAMMER
   2Ø REM VERSION B1.7
   3Ø REM AUTHOR  Alan Webster
   4Ø REM BEEBUG  OCTOBER 1985
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
  1ØØ MODE7
  11Ø PROCassemble
  12Ø PROCinit
  13Ø ON ERROR GOTO 141Ø
  14Ø REPEAT
  15Ø PROCmenu
  16Ø UNTIL Ø
  17Ø END
  18Ø :
 1ØØØ DEFPROCinit
 1Ø1Ø OSCOM=&7BØØ:V%=Ø
 1Ø2Ø *K.1Ø O.|M?&FE61=-1|M
 1Ø3Ø ENDPROC
 1Ø4Ø :
 1Ø5Ø DEFPROCmenu:VDU23,1,Ø;Ø;Ø;Ø;26
 1Ø6Ø ?&FE61=255:H%=21:*FX21
 1Ø7Ø PROCtitle:PROCtest:PRINT':*FX2ØØ
 1Ø8Ø PRINTCHR$131"1 - Save Eprom To Fil
e"'CHR$131"2 - Program Eprom From File"'
CHR$131"3 - Display Eprom"
 1Ø9Ø PRINTCHR$131"4 - Verify Eprom Agai
nst File"'CHR$131"5 - Quit"
 11ØØ PRINT'CHR$134"* - O.S Command"''
 111Ø PRINTCHR$133"Please Enter Your Cho
ice:";
 112Ø G$=INKEY$(5):H%=H%+1
 113Ø IF G$="" PROCtest:GOTO112Ø
 114Ø G=ASC(G$)-48:IFG=-6 G=6
 115Ø IF G<1ORG>6 PROCtest:GOTO112Ø
 116Ø VDU31,26,16:PRINTG$'
 117Ø ON G GOTO 118Ø,119Ø,12ØØ,121Ø,122Ø
,123Ø
 118Ø PROCdispsave(Ø):ENDPROC
 119Ø PROCprogram:ENDPROC
 12ØØ PROCdispsave(1):ENDPROC
 121Ø PROCverify:ENDPROC
 122Ø PROCquit:ENDPROC
 123Ø PROCstar:ENDPROC
 124Ø :
 125Ø DEFPROCtest
 126Ø CALL mctest:IF !bytes=-1 test=1 EL
SE IF !bytes=Ø test=Ø ELSE test=2
 127Ø size=Ø:IF !bytes=-1 OR !bytes=Ø TH
EN 129Ø
 128Ø IF !bytes2=!bytes size=&2Ø ELSE si
ze=&4Ø
 129Ø PRINTTAB(3,2)CHR$134"Eprom Size=";
size/4;"K. "'TAB(3,3)CHR$134"Eprom ";
 13ØØ IF test=Ø PRINT"Not Present";ELSE
IF test=1 PRINT"Blank";ELSE IF test=2 PR
INT"Not Blank";
 131Ø PRINTSPC(1Ø);:IF H%>2Ø H%=Ø:PROCro
mt
 132Ø ENDPROC
 133Ø :
 134Ø DEFPROCstar
 135Ø VDU42:INPUTLINE""$OSCOM:X%=OSCOM M
OD 256
 136Ø Y%=OSCOM DIV 256:CALL &FFF7
 137Ø PROCkey:*FX2ØØ 1
 138Ø IFHY=ASC("*") PRINT:GOTO135Ø
 139Ø ENDPROC
 14ØØ :
 141Ø ON ERROR OFF
 142Ø CLOSE#Ø:IF ERR=17 THEN 12Ø
 143Ø MODE7:*FX2ØØ
 144Ø REPORT:PRINT" at line ";ERL
 145Ø END
 146Ø :
 147Ø DEFPROCtitle:CLS
 148Ø FORF%=1TO2:PRINTTAB(Ø,F%)CHR$129CH
R$157CHR$141CHR$131"BEEBUG Eprom Program
mer Software   "CHR$156:NEXT:PRINTTAB(Ø,
3)CHR$147;STRING$(38,CHR$172)
 149Ø PRINTTAB(3)CHR$133"By  Alan Webste
r and Geoff Bains"
 15ØØ VDU28,Ø,24,39,6
 151Ø ENDPROC
 152Ø :
 153Ø DEFPROCquit
 154Ø VDU22,7:END
 155Ø :
 156Ø DEFPROCassemble
 157Ø DDRA=&FE63:DDRB=&FE62
 158Ø PCR=&FE6C:bytes=&8Ø:bytes2=&84
 159Ø IORB=&FE6Ø:IORA=&FE61
 16ØØ LO8=&7Ø:HI6=&71
 161Ø TEMP=&73
 162Ø DISPWIDTH=32:Esize=&88
 163Ø FOR PASS=Ø TO 2 STEP2
 164Ø P%=&71ØØ:[OPT PASS
 165Ø .read:LDA #&FF
 166Ø STA DDRA:LDA #&CE
 167Ø STA PCR:LDA#255:STA DDRB
 168Ø LDA#192:STA IORA:LDA LO8:STA IORB
 169Ø LDA #&EE:STA PCR
 17ØØ LDA HI6:AND #&3F:ORA #&4Ø
 171Ø STA IORA:LDA #Ø:STA DDRB
```



[illegible]

By Alan Webster and Geoff Bains

```
    Eprom Size=16K.
    Eprom Not Blank

Alan's silly ROM

1 - Save Eprom To File
2 - Program Eprom From File
3 - Display Eprom
4 - Verify Eprom Against File
5 - Quit

* - O.S Command

Please Enter Your Choice:
```

```
 172Ø LDA IORB:STA TEMP
 173Ø RTS
 174Ø .title:LDA#Ø:STA HI6:LDA#9:STA LO8
 175Ø LDY#38:LDA#32:.ylp:JSR&FFEE:DEY:CP
Y#Ø:BNE ylp
 176Ø LDA#31:JSR&FFEE:LDA#1:JSR&FFEE:LDA
#5:JSR&FFEE
 177Ø LDY#Ø:.rdlp3:JSR read:BEQ endt:JSR
&FFEE:INC LO8:INY:CPY#38:BNE rdlp3
 178Ø .endt:RTS
 179Ø .mctest:LDA#Ø:STA LO8:STA HI6
 18ØØ LDY#Ø:.rdlp:JSR read:STA bytes,Y:I
NC LO8
 181Ø INY:CPY#4:BNE rdlp:LDA#Ø:STA LO8:L
DA#&2Ø:STA HI6
 182Ø LDY#Ø:.rdlp2:JSR read:STA bytes2,Y
:INC LO8:INY:CPY#4:BNE rdlp2:RTS
 183Ø .getdata:LDA #Ø
 184Ø STA &8C:STA &8D:LDA#DISPWIDTH:STA&
8B
 185Ø LDA &76:BEQ getlp
 186Ø LDA#12
 187Ø JSR &FFEE:LDA#14:JSR &FFEE
 188Ø .getlp:CLC:LDA Esize:CMP &8D:BEQ r
eadend
 189Ø LDA &76:BEQ over
 19ØØ LDA&8B:CMP#DISPWIDTH:BEQ num
 191Ø .bck:JSR decit
 192Ø .over
 193Ø LDA &8C:STA LO8
 194Ø LDA &8D:STA HI6
 195Ø JSR read
 196Ø LDA &76:BNE dsp
 197Ø LDY #Ø:CLC:LDA &8D:ADC #&3Ø:STA &7
8
 198Ø LDA &8C:STA &77:LDA TEMP:STA (&77)
,Y
 199Ø JMP ok2
 2ØØØ .dsp:LDA TEMP
 2Ø1Ø CMP #127:BCS dot
 2Ø2Ø CMP #32:BCS ok
 2Ø3Ø .dot:LDA #46
 2Ø4Ø .ok:JSR &FFEE:.ok2
 2Ø5Ø BIT&FF:BMI escp:CLC
 2Ø6Ø LDA &8C:ADC #1:STA &8C
 2Ø7Ø LDA &8D:ADC #Ø:STA &8D
 2Ø8Ø CLC:CMP #&4Ø:BNE getlp
 2Ø9Ø .readend:LDA#15:JSR&FFEE:RTS
 21ØØ .num:JSR&FFE7:LDA #13Ø:JSR&FFEE
 211Ø LDA #38:JSR &FFEE
 212Ø LDA &8D:JSR shift
 213Ø LDA &8D:AND#&F:JSR disp
 214Ø LDA &8C:JSR shift
 215Ø LDA &8C:AND#&F:JSR disp
 216Ø LDA #134:JSR &FFEE:JMP bck
 217Ø .shift:AND#&FØ:LSRA:LSRA
 218Ø LSRA:LSRA
 219Ø .disp:CLC:CMP#1Ø:BCC nmb
 22ØØ CLC:ADC#7:.nmb:ADC#48
 221Ø JSR &FFEE:.escp:RTS
 222Ø .decit:SEC:LDA&8B:SBC#1:STA&8B
 223Ø CMP#Ø:BNE ret:LDA#DISPWIDTH:STA&8B
 224Ø .ret:RTS
 225Ø .program:LDA #Ø:STA &8C:STA &8D
 226Ø LDA #255:STA DDRA:STA DDRB
 227Ø LDA #128:STA IORA:LDA#5Ø:STA&74:ST
A&75:JSR wait
 228Ø .prglp:LDA #&CE:STA PCR:LDA#134:JS
R&FFEE
 229Ø LDA #38:JSR &FFEE:LDA &8D:JSR shif
t:LDA &8D:AND#&F:JSR disp:LDA &8C:JSR sh
ift:LDA &8C:AND#&F:JSR disp:LDA#13:JSR&F
FEE
 23ØØ LDA &8C:STA IORB
 231Ø LDA #&EE:STA PCR
 232Ø LDA &8D:AND #&3F:ORA #&8Ø
 233Ø STA IORA:CLC
 234Ø LDA &8D:ADC #&3Ø
 235Ø STA &78:LDA &8C:STA &77
 236Ø LDA #Ø:STA &74:LDA #49:STA &75
 237Ø LDY #Ø:LDA (&77),Y:STA IORB
 238Ø LDA #&EC:STA PCR:JSR wait
 239Ø LDA #&EE:STA PCR:CLC
 24ØØ LDA &8C:ADC #1:STA &8C
 241Ø LDA &8D:ADC #Ø:STA &8D
 242Ø CMP Esize:BNE prglp
 243Ø RTS
 244Ø .wait:DEC &74:BNE wait
 245Ø DEC &75:BNE wait:RTS
 246Ø ]:NEXT:ENDPROC
 247Ø :
 248Ø DEFPROCsave:IF size=Ø PROCersize:E
NDPROC
 249Ø INPUT'"Filename: "F$
 25ØØ A$="save "+F$+" 3ØØØ +"
 251Ø G=&1ØØ*size:A$=A$+STR$~G
 252Ø $OSCOM=A$:X%=OSCOM MOD 256
 253Ø Y%=OSCOM DIV 256:CALL &FFF7
 254Ø ENDPROC
 255Ø :
 256Ø DEFPROCersize
 257Ø PRINT"?? No Eprom To Save ??":VDU7
:PROCkey
 258Ø ENDPROC
```

```
 259Ø :
 26ØØ DEFPROCdispsave(L%)
 261Ø ?&76=L%
 262Ø ?Esize=size:CALL getdata
 263Ø IF L%=Ø AND V%=1 ENDPROC
 264Ø IF L%=Ø PROCsave:ENDPROC
 265Ø PROCkey
 266Ø ENDPROC
 267Ø :
 268Ø DEFPROCprogram
 269Ø CALL mctest:IF !bytes<>-1 VDU7:END
PROC
 27ØØ VDU12:PRINT'CHR$131"Filename :";:I
NPUT""F$:PRINT'':*FX2ØØ 1
 271Ø A%=5:X%=Ø:Y%=&73
 272Ø F%=&72ØØ:$F%=F$
 273Ø !&73ØØ=F%:CALL &FFDD
 274Ø lngth=!&73ØA AND &FFFF
 275Ø IF lngth>&4ØØØ PRINT"File too long
";:PROCkey:ENDPROC
 276Ø IF lngth>&2ØØØ size=&4Ø ELSE size=
&2Ø
 277Ø IF lngth=Ø PRINT"File Empty";:PROC
key:ENDPROC
 278Ø D$="LOAD "+F$+" 3ØØØ"
 279Ø $OSCOM=D$:X%=Ø:Y%=OSCOM DIV 256
 28ØØ CALL &FFF7:?Esize=size
 281Ø PRINTCHR$13Ø"Ready To Go (Y/N):"'
 282Ø REPEAT:S=(GET AND 223)
 283Ø UNTILS=89 OR S=78
 284Ø IF S=78 ENDPROC
 285Ø CALL program
 286Ø ENDPROC
 287Ø :
 288Ø DEFPROCkey
 289Ø PRINT''CHR$131"Press Any Key To Re
turn To Menu";:HY=GET
 29ØØ ENDPROC
 291Ø :
 292Ø DEFPROCverify
 293Ø V%=1:CLS:PROCdispsave(Ø)
 294Ø PRINT'CHR$13Ø"Filename:";:INPUT""F
$
 295Ø B%=OPENIN(F$):E%=Ø
 296Ø FORS%=1TO (size*&1ØØ):PRINTCHR$13;
"&";~(S%-1);
 297Ø IF BGET#B%<> S%?&2FFF E%=E%+1:PRIN
TCHR$133"Verify Error at &";~(S%-1)
 298Ø NEXT:CLOSE#Ø:V%=Ø
 299Ø IF E%=Ø PRINT''CHR$134"No Errors."
 ELSE PRINT''CHR$129;E%;" Error(s)."
 3ØØØ PROCkey
 3Ø1Ø ENDPROC
 3Ø2Ø :
 3Ø3Ø DEFPROCromt
 3Ø4Ø IF test<>2 PRINTTAB(1,5)STRING$(38
,CHR$32);:ENDPROC
 3Ø5Ø VDU31,1,5
 3Ø6Ø CALL title
 3Ø7Ø ENDPROC
```

SUPPLEMENT

INDEX

Advertising in BEEBUG	18
Bulletin Boards-corrections	18
Classified Ads	11
Competition Results	7
Discounts	6, 8, 10
Events	7
High Scores	7

ADVERTISERS

Addictive Games	9
BEEBUGSOFT	2–5
Cumana	24
Datapen	19
Datastar Systems	14
Electronequip	17
GCC (Cambridge) Ltd	16
J & O Software	16
Microware	12–13
Opus	20–23
Promicro	10
Speedysoft	15
Statistical Services Centre	18

BEEBUGSOFT UPGRADES

We offer a full service for members who wish to upgrade their Beebugsoft products from cassette to disc, disc to Rom etc., or to the latest version (e.g. Masterfile I to Masterfile II). To take advantage of this service please return your original program along with your order and remittance, deducting the following amounts from the price. (The amounts shown should be deducted from the full price, before calculating members discount). Send all orders to the usual High Wycombe address.

Product		Discount
● Masterfile I (disc) to Masterfile II (disc)		£9.50
● Masterfile I (cass) to Masterfile II (disc)		£5.00
● Exmon I (cass) to Exmon II (Rom)		£5.00
● Exmon I (Rom) to Exmon II (Rom)		£13.50
● Sleuth (Rom) to Sleuth 1.06 (Rom)	*[see below]	£19.00
● Spellcheck I (disc) to Spellcheck II (Rom + disc)		£9.50
● Paintbox (disc) to Paintmaster [Paintbox II] (disc)		£6.00
● Paintbox (cass) to Paintmaster [Paintbox II] (disc)		£5.00
● Design (cass) to Design (disc)		£5.00
● Dumpmaster (cass) to Dumpmaster (disc)		£5.00
● Hershey Characters (cass) to Hershey Characters (disc)		£5.00
● Quickcalc (cass) to Quickcalc (disc)		£6.50
● Sprites (cass) to Sprites (disc)		£5.00
● Starter Pack (cass) to Starter Pack (disc)		£6.50
● Teletext (cass) to Teletext (disc)		£5.00

SLEUTH – An Enhanced Version

The Sleuth upgrade is for owners of versions prior to 1.06 who wish to purchase the enhanced Sleuth which now also features:

1. The ability to enter Sleuth directly from any stage of your own program.
   e.g. IF D% = 10 THEN *SLEUTH or ON ERROR *SLEUTH.
2. The new command 'OUT' to exit Sleuth and continue running your program from the current line.
3. Dual screen workspace located in sideways Ram.
4. Variable update now includes arrays.
5. Aries B–20 and Watford 32k Ramcard compatibility.

BEEBUG SOFT

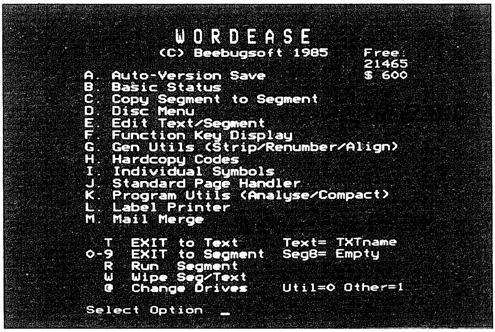
*See above for details on how to upgrade to the new Sleuth.

$W_4 O_1 R_1 D_2 E_1 A_1 S_1 E_1$

THE WORDWISE PLUS UTILITIES DISC

WORDWISE PLUS IS A WORDPROCESSOR WITH GREAT POTENTIAL — FULFIL THAT POTENTIAL WITH WORDEASE.

WORDEASE CONTAINS A HOST OF UTILITIES ALLOWING YOU TO STREAMLINE WORDPROCESSING WITH WORDWISE PLUS. MORE THAN THIS IT PROVIDES A NUMBER OF SPECIAL APPLICATIONS — SUCH AS MAIL MERGING AND LABEL PRINTING, IN ADDITION TO PROGRAMMING UTILITIES, SUCH AS THE VARIABLE LISTER AND POWERFUL COMPACTING ROUTINES, WHICH SQUASH REMS, SPACES, LABELS AND KEYWORDS. SUPPLIED ON A DUAL 40/80 TRACK DISC.



```
                 W O R D E A S E
            (C) Beebugsoft 1985      Free:
                                     21465
     A. Auto-Version Save            $ 600
     B. Basic Status
     C. Copy Segment to Segment
     D. Disc Menu
     E. Edit Text/Segment
     F. Function Key Display
     G. Gen Utils (Strip/Renumber/Align)
     H. Hardcopy Codes
     I. Individual Symbols
     J. Standard Page Handler
     K. Program Utils (Analyse/Compact)
     L. Label Printer
     M. Mail Merge

        T   EXIT to Text        Text= TXTname
     0-9   EXIT to Segment     Seg8= Empty
        R   Run  Segment
        W   Wipe Seg/Text
        @   Change Drives       Util=0 Other=1

     Select Option _
```

A_1 *AUTOMATICALLY SAVES YOUR TEXT WITH SELF-INCREMENTING VERSION NUMBER.*

B_3 *REPORTS ON THE STATUS OF EACH SEGMENT, SHOWING CONTENTS AND MEMORY USAGE.*

C_3 *COPY A FILE FROM SEGMENT TO SEGMENT, ALLOWING 'OVERWRITE', 'TO CURSOR' OR 'MARKED SECTION ONLY'.*

D_2 *AUTOMATICALLY READS DISC TO PROVIDE A DETAILED MENU OF YOUR FILES. PRESS A KEY TO LOAD FILE OF YOUR CHOICE.*

E_1 *TEXT EDITING AID — GIVES A WINDOW ONTO TEXT AREA WITH INSTANT SEARCH AND REPLACE OPTIONS.*

F_4 *DISPLAY AND EDIT THE CONTENTS OF THE FUNCTION KEYS.*

G_2 *A SET OF GENERAL UTILITIES INCLUDING AN EMBEDDED COMMAND REMOVER, AN INDENT FACILITY, A SECTION RENUMBERING ROUTINE AND AN ALPHA SORT.*

H_4 *AUTOMATICALLY ADD PRINTER CODES TO YOUR TEXT AT THE PRESS OF A KEY: UNDERLINE, ENLARGE, CONDENSE, ETC.*

I_1 *PREDEFINED SYMBOLS (ARROWS, COPYRIGHT, SCISSORS ETC.) AUTOMATICALLY ADDED TO YOUR TEXT.*

J_8 *AUTOMATICALLY MAINTAIN AND LOAD A LIBRARY OF STANDARD PAGE SETTING OR LETTERS.*

K_5 *PROGRAMMING UTILITIES (FOR WORDWISE PLUS PROGRAMMERS) INCLUDING PROGRAM SQUASHER, VARIABLES DISPLAY AND PROGRAM DE-CRUNCHER.*

L_1 *FLEXIBLE AND EASY TO USE MULTI-FORMAT LABEL PRINTER.*

M_3 *POWERFUL MAIL MERGE ROUTINES FOR SENDING PERSONALISED STANDARD LETTERS.*

WORDEASE FROM BEEBUGSOFT — £21.50 inc. VAT and POSTAGE

Please send me .... WordEase @ £21.50

Name

Address

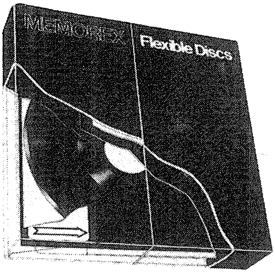
Debit my Access/Barclaycard with £

No. □□□□□□□□□□□□□□□□

Telephone credit card orders 0494 816666 24hrs.
Telephone for further information 0727 40303.
BEEBUGSOFT, PO Box 109, High Wycombe, Bucks. HP10 8NP

BEEBUG SOFT

MEMOREX DISCS
FROM BEEBUGSOFT



Memorex discs are extremely reliable and ideal for use with the BBC Micro. They have performed well in use at Beebug and we are pleased to recommend them and offer them at a reduced price.

3481 — Memorex 5.25 inch single sided, soft sectored, double density, 48 tpi disc, with hub ring. Suitable for 35 or 40 track systems.
BOX OF 10 — save £9.31. Normal price £24.81. Members Price £15.50.

3501 — Memorex 5.25 inch double sided, soft sectored, double density, 96 tpi disc, with hub ring. Suitable for 77 or 80 track systems.
BOX OF 10 — save £16.01. Normal price £41.51. Members Price £25.50.

Prices include VAT but not postage. Please add £0.60 for the first box and £0.40 for each subsequent box.

Overseas orders (including Eire): Calculate U.K. price, including postage, then deduct 15% VAT and add £2.00 per box.

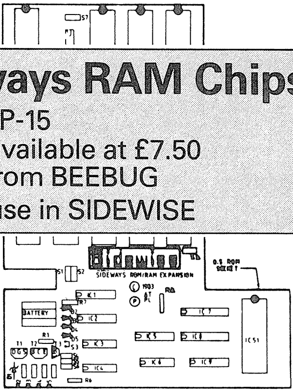
Please send a cheque with order and membership number to, BEEBUGSOFT, PO Box 109, High Wycombe, Bucks HP10 8NP

ATPL's SIDEWAYS ROM EXPANSION BOARD
Normal Price £44.70 Members Price £39.00

- Simply plugs into the BBC Micro
- No soldering necessary
- Increases the sideways ROM capacity to 16
- Fully buffered - allows all sockets to be used
- Complete with full and detailed instruction booklet
- Accepts 16K RAM in special sockets
- Battery back up facility for RAM (parts available directly from ATPL at extra cost)
- As used at BEEBUG
- Reviewed in BEEBUG vol. 2 number 6

Sideways RAM Chips

8K 6264 LP-15
are now available at £7.50
inc. VAT from BEEBUG
Ideal for use in SIDEWISE



HOW TO ORDER

Please send your order with a cheque/ postal order made payable to BEEBUG, and enclose your membership number. Overseas members, please send the same amount, this will cover the extra, post but not VAT.

The address for SIDEWAYS is:
BEEBUGSOFT, PO Box 109, High Wycombe, Bucks. HP10 8NP

BEEBUG
SOFT

NEW COMPUTER CONCEPTS ROMS FROM BEEBUGSOFT

For a full specification of all the ROMS please send A5 SAE to the address below. Mark your envelope "ROM SPEC"

NEW INTER-CHART
THE CHART & GRAPH
PLOTTING ROM

This ROM-LINK compatible program allows practically any data to be presented in graphic form. Three basic types of graph are supported—the pie chart, bar graph and line graph. INTER-CHART is especially tailored to INTER-SHEET and so can very easily use spreadsheet data for plotting charts and graphs. RRP £36.80
Members price **£29.90**

DISC DOCTOR — The general utility Rom for disc users.
RRP £33.35
Members Price **£29.00**

INTER SHEET — A spreadsheet Rom allowing 40,80 or 105 column screen display and compatible with Wordwise/View. The first of a 'Rom-Link' suite of products which will integrate together allowing data to be passed from one program to another. RRP £56.35
Members Price **£52.35**

PRINTMASTER — for Epson and Star printers — Allows easy access to your printer's special features. Specify Epson or Star version. RRP £33.35
Members Price **£29.00**

WORDWISE — The orginal 8k wordprocessor. RRP £46.00
Members Price **£36.00**

WORDWISE PLUS — The improved word processor on 16k Eprom, incorporating its own programming language. RRP £56.35
Members price **£53.90**

ACCELERATOR— A comprehensive 24k Rom (2 Roms) Compiler. Supports full floating point arithmetic and will compile most Basic programs with only slight alterations. RRP £64.40
Members Price **£59.90**

GRAPHICS ROM — A host of routines for anyone interested in extended graphics on the BBC Micro. RRP £33.95
Members Price **£29.00**

SPEECH ROM — An advanced speech system using phonemes to allow any spoken word to be produced. This Rom requires the Acorn Speech processor to be present in your machine (A part of the Acorn Speech System. If required this chip is available at an extra cost of £10.) RRP £33.35 Members Price **£31.00**

UPGRADES from WORDWISE to WORDWISE PLUS — return your old Wordwise manual and Eprom by recorded delivery for upgrade. (Damaged Eproms enc. extra £5) UK **£21.85** Airmail to Europe **£26** Elsewhere **£28**
UPGRADES TO END ON 18/10/85 — Please note that Computer Concepts will no longer provide upgrades from Wordwise to Wordwise-Plus after 18th Oct. 1985.

COMPUTER CONCEPTS ROM ORDER FORM

Please send me ____________________ price ______________

____________________ ______________

____________________ ______________

TOTAL ENCLOSED ______________

Credit Card orders may be telephoned to (0727) 40303

Prices include post, packing and VAT, overseas members send the same amount to include extra post but not VAT. Please make cheques payable to BEEBUGSOFT.

Send to: **"Rom Offer" Beebugsoft, PO Box 50, Holywell Hill, St. Albans, Herts. AL1 3YS**

To avoid delay please do not send this to the High Wycombe address, please do not combine orders for other BEEBUGSOFT products with this order.

Name ____________________ *Membership No.* ____________________

Address ____

____

Access/Visa Card No. □□□□ □□□□ □□□□ □□□□ *Expiry date*

Discounts

BEEBUG has arranged discounts for members at a number of retail outlets who supply computer books, software, hardware and services. We are continually negotiating further discounts.

It is advisable to telephone before placing an order, to check availability. Members should simply quote their membership number with their order, though members taking discount will not necessarily be given credit card facilities, (you must check this). We are not acting for these companies, nor receiving payment from them, and cannot be held responsible for their services.

AKHTER computer group,
28/29 Burnt Mill, Harlow,
Essex CM20 2HJ.
Tel: 0279 443521

5% discount to members for cash or cheque with order.

ALLM Books,
21 Beechcroft Road,
Bushey, Herts.
Tel: 0923 30150

10% discount for members of BEEBUG and ELBUG the Small Computer Program Index and other database and library software.

Appleby, Myers and Clarke,
60 Market Street,
Watford, Herts.
Tel: 0923 31960

Free postage on books.

AVC Software,
PO Box 415, Harborne,
Birmingham B17 9TT

25% off 'Learning with the Beeb', and 'Learning with the Elk' discount price: £1.50 for BEEBUG and ELBUG.

Calancraft Ltd.,
134a St. John's Road,
Woking, Surrey GU21 1PS
Tel: 0932 65284

From 5% to 15% on microcomputer equipment, workstations, disc drive cases & PSU's and software.

Computer Holidays,
37 University Road,
Highfield, Southampton
Tel: 0703 558621

£5 discount for BEEBUG and ELBUG.

Danum Computer Systems,
17 East Laith Gate,
Doncaster,
S. Yorkshire DN1 1JG
and Computer Room,
Unit 12, Market Hall,
Barnsley, S. Yorks.

Discount prices on request.

East Riding Computers,
24 Sherbuttgate Road
South,
Pocklington,
York YO4 2YB
Tel: 07592 4387

Discount prices to members on request.

K. Edward,
54 Greyham Court,
Dalclaverhouse,
Dundee DD4 9DQ

5% discount VDU Support Stand kits.

fFolders,
10 Weston Lane,
Bridgetown,
Totnes, Devon.

15% discount on function key labels.

FUREY Enterprise Products,
7 Manor Park,
Froxfield, Wilts. SN8 3LF
Tel: 0488 83006

10% Discount.
BBkey Function key booklet — £2
BBtype word processor — £23
View/Wordwise Reformatter — £2

Globe Book Services,
Houndmills,
Basingstoke,
Hants. RG21 2XS

10% Discount on 'Assembly Language Programming for the BBC Micro' by I. Birnbaum.

→8

MUSIC COMPETITION RESULTS

The music competition of BEEBUG Vol.4 No.2 brought in so big a response that we only just managed to listen to all the entries in time to announce the results this month. The standard of all the entries was very high and a great number of you had obviously worked long into several nights to complete your programs. The task of the competition was to write a Basic program to reproduce a piece of music, making the most of the four channels and envelopes that the BBC micro provides.

Some staggering entries appeared at the BEEBUG office but when it arrived it was obvious to all that the winner was to be Nigel Hambly's version of Strauss's Blue Danube. This epic work (the program and data is in several parts taking up about 80K of disc space!) plays an accurate rendition of this famous waltz and even includes some excellent graphics. The first prize of an ATPL Symphony keyboard, BEEBUGSOFT Murom and Ian Waugh's book will soon be on their way to Mr. Hambly.

Second prize (and only just pipped at the post) goes to Melvyn Wright for his 'music machine' that plays Blaze Away March by Holzmann. The music machine is a fairground organ with all the necessary instruments recreated. One highly novel and ingenious trick used in this program is to switch the cassette motor relay on and off as a rhythm track! A copy of Murom and Ian Waugh's book go to Mr. Wright.

New High Scores BEEBUG

Supplier	Game	Score	Player
BEEBUG MAG	Cosmonaut	4040	R.Koten
BEEBUG MAG	Truffle Hunt	6189	R.Koten
FIREBIRD	Duck	55200	E.Denning
H SOFT	Penguin	163730	J.Dowling
MICRO POWER	Castle Quest	2210	S.Rawnsley
PSION	Saloon Sally	56380	R.Koten
SUPERIOR SOFT	Jack & Beanstalk	11430	R.Koten

Events

Compec '85	Olympia, London	12-15 Nov
Micro User	Horticultural Halls Westminster	14-17 Nov
Computers in Education	Barbican Centre	22-25 Jan 1986

Hermes Computer Products,
23 Durnford Close,
Norden, Rochdale OL12 7RX

10% discount + free p.&p. on a graphics design pad.

Infomania,
Carey Place, Watford,
Herts. WD1 2LR
Tel: 0923 55122

15% discount on courses to BEEBUG members. Also discount membership to their club.

ITM Ltd.,
E3 New Enterprises,
S.W. Brunswick Dock,
Liverpool L3 4AR

Rotaview Professional videotext system. Usually £70 + VAT + pp. Now with £5 discount.

Kingsley TV Services,
40 Shields Road,
Newcastle Upon Tyne
Tel: 0632 650653

£10 discount off their range of Monitors and TV/ Monitors.

Microage,
135 Hale Lane,
Edgware, Middx. HA8 9QP

5% discount off some disc drives.

Microguides,
St. James House,
105-113 Broadway,
London W13 9BL

10% discount for f3 or more of their keyboard overlays.

Microtest Ltd.,
18 Normandy Way, Bodmin,
Cornwall PL31 1EX
Tel: 0208 3171

10% discount on Software, and 5% discount on Hardware.

Microware,
637 Holloway Road,
London N19

10% discount on DDFS. Other discount prices as shown in Microware advert.

Micro Computer World,
1 Lane Close, London
NW2 6QZ
Tel: 01-452 0893
Write for free list.

Discount on Software for BBC & Electron. Discount on Hardware Joysticks and Accessories.

Micro Supplies.
Oxford House, Faldouet,
St. Martin, Jersey, C.I.
Tel: 0534 55743

5% discount on diskettes.

Mine of Information,
(Mail Order),
1 Francis Avenue,
St. Albans,
Herts. AL3 6BL
Tel: 0727 52801

Computer Books 5% discount.

Off Records,
Computer House,
58 Battersea Rise,
Clapham Junction,
London SW11 1HH

25% off diskette purchases.

Opus Supplies,
158 Camberwell Road,
London SE5 0EE

20% discount for members on business desk only.

Oxford Computer Supplies,
PO Box 133,
Oxford OX4 3NB
Tel: 0865 716137

7.5% discount on discs, printer paper & word-processing paper.

P.M. Communications,
47 Wickentree Lane,
Failsworth,
Manchester M35 9AY
Tel: 061 688 7444

Large discounts on disc drives, diskettes, and power supplies.

Shinecrest,
Unit 14, Dicker Mill,
Hertford, Herts. SG13 7AE
Tel: 0992 51977

10% off their home computer work station to ELBUG and BEEBUG members.

Simonsoft,
25 Tatham Road, Abingdon,
Oxon. OX14 1QB

5% to members of BEEBUG on Sprites v.2, Superfruit, Castaway. 5% to members of ELBUG Sprites v.2, Superfruit.

A. Spencer,
74 Dovers Park,
Bathford,
Bath BA1 7UE

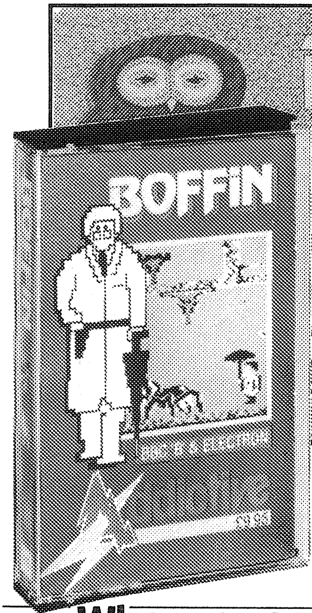
25% to members for Journal Indexing System and Objective Test System.

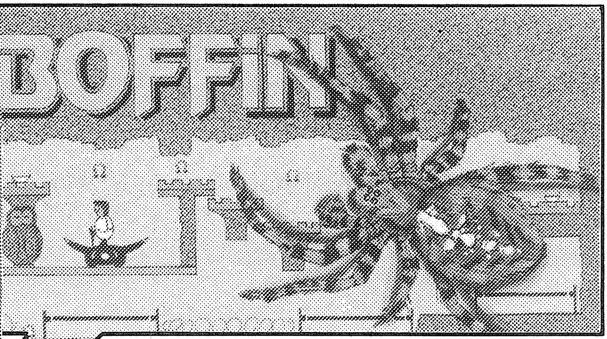
→10

Can YOU avoid the fangs of a deadly spider?

▷ Addictive launch BOFFIN ◁

A Mad Professor on the Loose





Boffin is a challenging, action-packed game requiring a combination of cunning and slick keyboard control. Beware the Giant Puffer–Blimps in your way and give the massive Tarantulas a wide berth! Land on the giant hydraulic rams and they will slam you into deadly neutron magnets embedded in the cave roof! Manta-bats are useful allies and the trampolines will help you gain height – but remember to use your umbrella to break a deadly fall!

Available BBC B and Electron £9.95

ALSO AVAILABLE

STRINGER
Stringer is an arcade adventure game in which you are a press photographer set the task of snatching a shot of a Film Star in an hotel. The game is tantalisingly playable, graphically excellent and, of course – Addictive!
Available . . . £8.95.

FOOTBALL MANAGER
The All Time Classic
Football Manager . . . Spectrum 48K £8.95 . . . Commodore 64 £8.95 . . . Amstrad CPC 464 £8.95 . . . BBC B £8.95 . . . Electron £8.95 . . . Dragon £7.95 . . . Vic 20 £7.95 . . . ZX81 £7.95.

When you open an Addictive Game, there's a new world of discovery waiting for you.

To: Addictive Games Ltd. 7a Richmond Hill, Bournemouth, Dorset BH2 6HE

Please send me:

□ cassette(s) for Boffin for ....(computer)

□ cassette(s) for Stringer for ....(computer)

□ cassette(s) of Football Manager for ....(computer)

I enclose cheque/postal order for* £ ....

Please debit my Access card No. ....

*Delete as applicable

Name ....

Address ....

....

Postcode .... Tel: ....

Access orders can be telephoned through on (0202) 296404

For overseas orders, please add £1.50.

Addictive

● Addictive Games ●
7a Richmond Hill, Bournemouth BH2 6HE
Telephone: 0202 · 296404.

Total Business Service,
29 Hollow Way Lane,
Amersham, Bucks.

A free program with every two purchased.

Universal Stands,
43 Pearce Avenue,
Parkstone, Poole,
Dorset BH14 8EG
Tel: 0202 740147

5% discount to members on their Universal Stand.

Wholesale Supplies Ltd.
Woking.
Tel: 04862 62020

Juki 6100 daisywheel printer. Model 6100. Brand new, fully guaranteed, free delivery UK mainland, £369 inc. VAT.

Zero Software,
29 St. Michaels Close,
N. Waltham,
Hampshire RG25 2BP

£1.95 off "EASYCALC" package.

Classified Ads

Rates are 15p per word for members' personal ads. Members' business ads cost 25p per word - and if you are selling any item in quantities rather than one-off, then that counts as business. From 1st May 1985 you must add 15% VAT to the cost of your ad. In either case send cheque with order to PERSONAL ADS, PO BOX 5Ø, ST ALBANS, HERTS. It is essential to include your membership number. Please note that we cannot arrange for box numbers.

Nutcrackers: Tape 2 Tape, Tape 2 Disc, Disc 2 Tape and Disc 2 Disc: Only £5! The best and cheapest package available. 'How to move your s/w to disc'. Extensive text with hints and suggestions + cassette with 9 programs to help you: £5.- Nutcracker + Howto together £9. We sell many other utilities: SAE. Note BBC only (sorry!). R-SOFT, 22 Marriotts Close, Felmersham MK43 7HD, Beds, Tel: Ø234 78173Ø.

'D-Master', the most powerful disc analyser and backup program. You need either an Acorn DFS or DNFS, or Watford DFS 1.3 onwards. It is better than other programs selling for more than £1Ø-£15! Price: £7.ØØ; for nutcracker owners: £3. R-Soft, 22 Marriotts Close, Felmersham, Beds MK43 7HD. Phone (Ø234) 78173Ø.

CP/M user group(s) disc libraries, 4ØØ+ volumes with 8ØØØ+ items £1.5Ø / vol., Acorn Z8Ø, Torch and other formats. Also format translation service. SAE or phone. R.Smith, 138 Holtye Road, East Grinstead, Sussex RH19 3EA. (Ø342) 313883.

MARKS AND STATISTICS for teachers and lecturers. Revolutionizes the handling of marks. Simple to use spreadsheet format. 33Ø students per file. Up to 3ØØ subjects. Sorts, analyses, normalizes, calculates means, applies weighting factors. Prints histograms and lists. Model B. O.S.1.2. Full documentation. Disc (4Ø or 8Ø track) Y16. In-Form, 73 Woodfield Park, Colinton, Edinburgh EH13 ØRA.

Sanyo DM8112CX monitor. 12 inch Hi-res monochrome. 6 months old. £55 ono. Dover 84Ø685.

List in colour in Mode 7. Coloured line numbers, 1Ø definable strings. £6 cassette, £7 disc. PAJ Taylor, 23 Limes Ave., Alfreton, Derbys. DE5 7DW.

Wanted. Phloopy cartridges. Thomas, 25 Wykewane, Malvern, Worcs. (Ø6845) 6348Ø. Evenings.

CP/M user group(s) disc libraries, 4ØØ+ volumes with 8ØØØ+ items £1.5Ø / vol., Acorn Z8Ø, Torch and other formats. Also format translation service. SAE or phone. R.Smith, 138 Holtye Road, East Grinstead, Sussex RH19 3EA. (Ø342) 313883.

Acorn DFS inc.8271 plus Cumana 4Ø track single sided 1ØØk full height drive £15Ø. Phone Ø788-2859 Evenings.

SALE! One only each item, most in original packaging: DISCS: Acheton (2-disc adv) £14; Magic Mushrooms £8; Swag £6; Mr-Ee £7; Beebugsoft's Paintbox £1Ø; Watford's Disc Executor (T-D) £6; Disc-Data (Database) £12; Debug / Copy-all £6. View PDG £1Ø; Watford FX-8Ø PD £6. Cassettes: Hunchback; Sealord; Galaxy Wars; Graphs and Charts; Early Learning; £2 each. Fortress; Frak!; Franklin's Tomb (adv); £3 each. ROMs: View 2.1 £32; ADE v1.Ø £3Ø; STARdataBASE £5Ø; Printmaster (Epson / Kaga) £18. Also: Watford MK2 ROM board £2Ø; UDM DDFS 2.Ø £5Ø. Phone Ø843-291167 after 6pm.

Wordwise Plus, as new £33. Quickcalc (disc) £7, Vu-Calc £5. G.Brown. Tel. Ø1-341-2187 (evenings).

Convert your 8Ø track 5.25" drive to 4Ø/8Ø switchable. Simply plug existing drive cable into small box provided, and supplied cable into your BBC. No soldering, drilling etc. Guaranteed. £18 includes p&p. Magus Electronics, 33 Pool Lane, Winterley, Sandbach, Cheshire.

BBC Graphics planning sheets. A4 pads of 8Ø sheets £1.99 plus 5Øp pence p&p each to YES, 2 Fleckney Rd., Kilby, Leicester LE8 1TB. Send SAE for sample.

Printmaster ROM £2Ø.ØØ. Computer Books (Cryer on graphics and Latham on disc drives £3 each) inc. postage. Ring Cardiff 48 56 84.

Listings-BBC, quick confidential service, any length, by return, £1.75 including p&p. Send tape to JRASS Services, 98 Preston Lane, Tadworth, Surrey.

Frustrated amateur with reasonable 'Basic' programming skills requires personal tuition, support and advice from an experienced Basic / Machine Code programmer in developing a specialist stockmarket analysis program (not for commercial use). Tel. Ø1-34Ø-2646 (N.London) evenings

Z8Ø second processor, complete with all software, virtually unused, boxed as new, £275. Bedford (Ø234) 54374.

BBC Model B, UDM DDFS, Microware double sided disc drive, Exmon, Toolkit ROMs, manuals inc. Advanced User Guide, many 65Ø2 reference books, complete set BEEBUG mags, box discs. £55Ø ono. Ø8Ø45 7292.

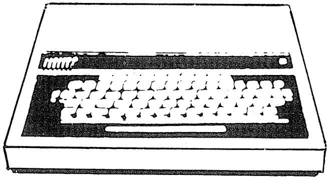
Acorn Mirle business suite, 7 discs, new and unused. Cost £175, accept £1ØØ. Epson MX8Ø F/T II printer, £2ØØ. Phone Ø5Ø86 4274 (Norwich).

'D-Master', the most powerful disc analyser and backup program. You need either an Acorn DFS or DNFS, or Watford DFS 1.3 onwards. It is better than other programs selling for more than £1Ø-£15! Price: £7.ØØ; for nutcracker owners: £3. R-Soft, 22 Marriotts Close, Felmersham, Beds MK43 7HD. Phone (Ø234) 78173Ø.

SYSTEMS

View	£58
ViewSheet	£57
WordWise	£32
WordWise Plus	£49
StarBase	£69
Gemini Data	£115
PrintMaster	£29
Termi	£69
Disk Doctor	£29
WS 2000	£135

BBC BD+	£475
BBC & DDFS	£379
BBC B	£349
BBC BD	£399



Communicator	£23
Databee	£23
Pace Modem (including software)	£135

BBC includes **5** games FREE

Sanyo 555 Q

Only **£999**

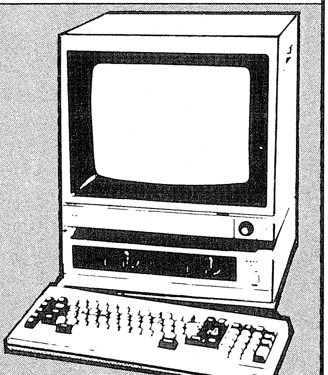
IBM compatible

2 × 360 Drives ★

Price includes:

Wordstar, Mailmerge, Info-Star, Calc-Star, Basic + DOS

Monitor only **£99**



NOW FOR A REAL BARGAIN

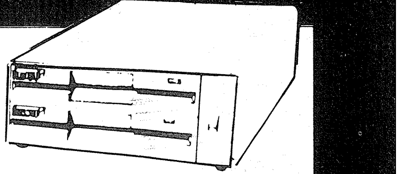
Prices include VAT

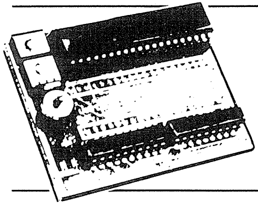
UDM DRIVES

200K - 400K - 800K

plus new 3½″ range

(including manual & formatter)



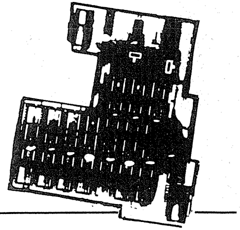


100K single unit	£79
200K dual unit	£150
200K single unit	£125
400K dual unit	£205

400K single unit	£160
800K dual unit	£259

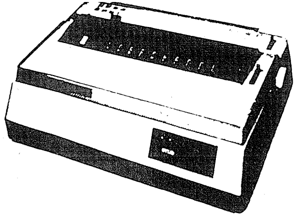
All 80TK drives are switchable 40/80

PS unit .... +£20



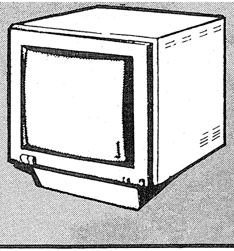
UDM DDFS....... **£95** with Disk Drive ...... **£79** ★

ROM Expansion *including FREE software*.... **£55.00**



Juki 6100	£379
Juki 2200	£249
Epson RX80 FT ..	£269
Epson FX80........	£379
Canon 80 COL	£299
Canon 156 COL ..	£399

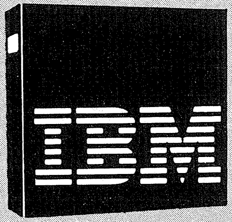
Medium resolution	£199
High resolution	£249
Extra High resolution ...	£399
AP = +£25	
Sanyo/BMC 12″ HC	£110
Philips 12″ Hi. Res.	£95



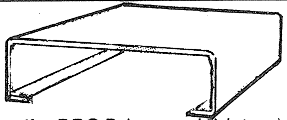
DISKETTES

in packs of 10

SS 40tk...........	£19.90
DS 40tk...........	£20.00
DS 80tk...........	£26.00



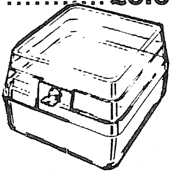
★ HIGHEST QUALITY YOU CAN BUY ★



(for BBC Drives and Printers)

ACCESSORIES

BBC Stands	£15.00
Dust covers	£5.60
Disk storage boxes.......... *(takes 35)*	£15.00
Disk storage boxes.......... *(takes 80)*	£20.00



MICROWARE

NORTH LONDON — 637 Holloway Road, London N19 5SS.
01-281 2431
SOUTH LONDON — 67 Westow Street, Upper Norward,
London SE19. 01-771 6373
MIDLANDS — 14 Charles Street, Hanley, Stoke-on-Trent.
(0782) 269 883

Name

Address

Signed

ORDER FORM

Prices include VAT

+£10 for delivery. £2.50 software and diskettes.

Product	Price	Qty	Total
			£
			£
			£

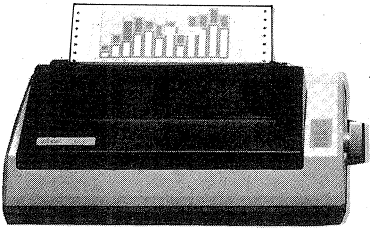
CREDIT CARD NO. ....

TOTAL SUM ENCLOSED____________

All products are brand new and are offered with full manufacturers' warranty.

Special offer to BEEBUG members. £30 off when you produce this advertisement

Yet another unbeatable deal from Datastar Systems!



STAR SG10

Character Sets

96 Standard ASCII Characters
88 Standard International Characters
96 Italic Characters
88 Italic International Characters
96 NLQ Characters
88 NLQ International Characters

Character Matrix

Standard Characters: 9 dot × 11 dot
STAR Block Graphics: 6 dot × 6 dot
IBM Block Graphics: 12 dot × 11 dot
User Defined: 8 dot × 4 to 11 dot
Near Letter Quality: 17 dot × 11 dot

Character Fonts

Normal (10 CPI)
Elite (12 CPI)
Condensed (17 CPI)
NLQ (10 CPI)

Line Spacing ⅙, ⅛ in. or 7/72 in. Standard

Dimensions (W × D × H)
392 × 315 × 145mm (15.4 × 12.4 × 5.7 in.)

Number of Columns
80, 96, 136, (40, 48, 68 Double Width)

Print Head 9 Wires (User Replaceable)

THE *INCREDIBLE* NEW STAR SG-10 NEAR LETTER QUALITY PRINTER.

features:—

- Epson and IBM compatible print codes
- Friction and tractor feeds.
- Externally Accessible DIP-Switches
- High-resolution bit-image graphics
- Downloadable character sets
- NLQ, italics, condensed, enlarged, super & subscript styles.
- High 120 CPS print speed, logic seeking.
- Hex dump facility.
- Macro-instructions.
- Touch-sensitive control panel.
- Standard parallel Centronics interface.
- 2K print buffer — expandable to 10K
- Commodore 64 version available shortly

Star SG-10 £259 + VAT	**£297.85**
Parallel cable for any Micro (max. price)	**£20**
2 Spare ink ribbons	**£5**
2000 sheets of continuous listing paper	**£15**
Next day doorstep delivery service	**£10**
	£347.85

Datastar's all in price £297.85!!!

Post your cheques to

Datastar Systems UK

Unicom House, 182 Royal College Street, London NW1 9NN. Telex 295931 UNICOM G
PERSONAL CALLERS WELCOME — We are situated by the junction of Camden Road, near the railway bridge
MONDAY-FRIDAY 9-6 SUNDAY 10-1 *EXPORT ENQUIRIES WELCOME*

Credit Card Hot Line 01-482 1711

AS MAIN DISTRIBUTORS FOR THE STAR RANGE, WE ARE ABLE TO OFFER EXCELLENT DEALER MARGINS

"YOUR PROGRAMS ALWAYS REACH ME BEFORE ONES ORDERED LOCALLY"

(I. Sinclair, N.S.W., Australia)

WE ALWAYS TRY TO SEND YOUR PROGRAMS ON THE SAME DAY WE GET YOUR ORDER, THAT'S WHY IT'S CALLED SPEEDYSOFT! All programs normally in stock before we advertise them! FOREIGN ORDERS WELCOME, satisfied customers in 63 countries over 3 years. UK prices include VAT: export prices are the same, plus p&p because of the extra work involved. Pay by sterling cheque, bank draft or postal order. All orders must be prepaid: we send your programs as soon as we get your money!. It's fastest to order by phone from any country with your VISA, EUROCARD, MASTERCARD or ACCESS. Call 01-846 9353, 24 hrs, 7 days a week. BIG FREE ILLUSTRATED CATALOGUE WITH EVERY ORDER WITH DETAILED REVIEWS, LOAD TIMES, ETC. Send £1.00 cash (refundable) for catalogue only. FREE GIFT if your order from this ad comes to £30! See last paragraph.

RED MOON

The latest adventure from the greatest! Text only but, as you would expect from Level 9, some of the best-written text around. You must use many weapons and spells as you collect the 9 treasures and find the Red Moon Crystal. No Sticks. (Level 9) **Cassette £6.95**

DDD PROGRAM SUITE

AT LAST, total disk-**based integration and data-file compatability between your DATABASE, SPREADSHEET and GRAPHICS programs.** Comprehensive, indexed manuals with training programs and demos on disk. Works even over 2 drive surfaces. All 3 programs fully compatible. Each can be bought separately, but you **save £19.90** if you buy all 3 together. (Gemini.) **DDD-BASE. DDD-CALC. DDD-PLOT. EACH DISK** (40/80 track) **£29.95 ALL 3 DISKS £69.95**

JUMP JET

"Combines not only the challenge of coping with a flight simulation, but also the thrill of combat ... it's fairly easy to get airborne, but the rest of the mission is very challenging." (C&VG) This is your chance to do what even very few pilots do, fly your own jump jet in combat. Good tunes, great graphics, very clear instruments, intelligent use of split-screens. Sticks OK. (Anirog) **Cassette £9.95 Disk (State 40 or 80 track) £11.95**

REVS

"After the magnificent ELITE we now have another winner, Revs. Graphically, it is excellent. It feels like driving a real racing car ... oozes class, style and, above all, realism ... one of the best simulations ever." (PCW) Sticks OK (Acornsoft) **CASSETTE £14.95 DISK** (40/80 track) **£17.65**

MATCH DAY

"Far superior to any other ... the reality of the whole match, with the reflection as the ball bounces and the quality graphics, make this a worthwhile buy." (ZXComp) "Must be the definitive football game ... this one has the best graphics, playability and atmosphere." (Crash) Good 3D graphics: play over many skill levels and time periods. Sticks OK. (Ocean) **Cassette £9.95**

NEW REPLAY ROM

"THE BEST FORM OF TRANSFER I HAVE MET ... I have nothing but praise to lavish upon it." (PCT) "An unusual device that promises to be a boon for disk-using games fans." (Beebug) These reviews are of the original TAPE/DISK ROM which transfers nearly all software to disk. BUT the actual ROM must be present before the SAVEd software will run. So no pirates! TD ROM is still available at £18.00. **REPLAY ROM** is the new enhanced version which is 100% effective. Even loads a stopped screen for printing. Disk to disk transfers. Midplay stop-and-save. REPLAY ROM is £35.00. (To upgrade existing TD ROM contact Vine.) For either TD ROM or REPLAY ROM, please order by type no: according to your DFS: TYPE 8271 for Acorn DFS, Acorn DNFS, AMCOM S/D, WATFORD S/D. TYPE 1770 for OPUS DDOS 3.45 & SOLIDISK DDOS: TYPE 2791 for Opus 3.15. (Vine) **TD ROM £18.00 REPLAY ROM £35.00**

BEACH HEAD

"A multi-screen arcade game that is very close to being perfect ... graphically it is astonishing ... a vast number of features." (PCW) A whole war on one tape! At last the famous Commodore/Spectrum program runs on your Beeb. 6 battles each require a different survival strategy. 1 or 2 players. 3 skill levels. Sticks OK. (U.S.Gold) **Cassette £9.95**

MOUSE GOING CHEEP!

Save £10.00 on the best BBC Mouse and make your Beeb a bit like an expensive Macintosh. 6502 2nd processor and Wordwise/View compatible. The package includes disk, cassette, two manuals, ROM and Mouse. PLUS AMX ART and ICON DESIGNER. 17 special mouse commands. AMX ART lets you draw pictures/diagrams and print on Epson-compatibles. ICON DESIGNER creates icons for use in your own programs. (AMS) **ROM/DISK PACKAGE.** (State 40 or 80 track: state 5.25 or 3 inch disks.) Recommended price £89.95 **OUR PRICE £79.95** SAVE £10.00

REPTON

"The graphics are excellent, being clear and colourful with smooth movement and scrolling ... the game provides an exciting challenge with a good balance between reward and failure ... very playable, highly addictive and thoroughly enjoyable." (ABComp) 12 screens of progressive difficulty, but on each screen you see only 1/16th of the whole playing area at once. Password feature lets you skip completed screens. Map helps with first 7 screens. No Stick. (Superior) **Cassette £9.95 Disk (State 40 or 80 track) £11.95**

ACCELERATOR ROM

"The best compiler around for the BBC at the moment." (Beebug 7/85) Claimed to be the most comprehensive compiler available for the Beeb. Compiles practically all BBC BASIC: the only one compatible with BBC BASIC syntax and offering full floating-point arithmetic. Package includes two ROMs, manual, utilities disk, quick reference card. (Computer Concepts) Recommended price £64.40 **OUR PRICE £55.00** SAVE £9.40

BBC MUSIC SYSTEM

"The best music software package for the Beeb, using the internal soundchip, that I have ever seen." (Beebug) "It is difficult to imagine a more comprehensive musical program." (ABComp) 76-page illustrated manual. But the 4-page leaflet has you watching one of the pre-recorded tunes play within 2 minutes. Then you start to adapt it... No Stick. (Island Logic) **2 Cassettes £25.95 2 Disks (40/80 track) £24.95**

SHARE ANALYSER

"An excellent program." (MUser) "Very highly recommended." (Which?) Comprehensive reporting, analysis and charting package. All printers OK. 20,000 whole number share prices plus 320 purchase/-sale/dividend records per disk. Many different reports available. 20-page manual. No Stick. (Synergy) **Cassette £17.95 Disk (State 40 or 80 track) £24.95**

WORDWISE + ROM

The greatest BBC wordprocessing ROM. 180-page manual. 56-page manual. Keystrip. Typing tutor. Example text. Example programs enable mailmerging, index generation. COMPATIBLE WITH EXISTING WORDWISE FILES and with 6502 2nd processor. (Computer Concepts) Recommended price £56.35 **OUR PRICE £50.00** SAVE £6.35

FREE TAPES DISKS

If your order from this ad comes to more than £30 *(excluding p&p)* we will send you 2 free 15min blank data tapes or 1 free 5.25 inch disk. Indicate which on your order!

VISA
ACCESS

SPEEDYSOFT

01-846 9353 (24 HRS)

EUROCARD
MASTERCARD

POST TO: SPEEDYSOFT (BB1)
37 CHURCH ROAD, LONDON SW13 9HQ, ENGLAND.
For CATALOGUE ONLY, send £1 cash. Refunded with your first order.
I own a BBC 'B'/BBC 'B' PLUS. I enclose a cheque/PO payable to Speedysoft OR charge my VISA/ACCESS/EUROCARD/MASTERCARD

No. □□□□ □□□□ □□□□ □□□□

Signature: .... Expiry Date ....
Please write clearly. If we can't read it, you won't get it.
Name: ....
Address: ....
....
.... Postcode: ....
PHONE NO: if any, in case of query ....

Program Name	Cass/Disk	Price
Postage & Packing	UK add 75p per order	
	Europe ADD £1.00 per program	
	Outside Europe ADD £1.50 per program	
	Total Order	

STARdataBASE . . .

The database

for the BBC

FREE! MAIL-MERGE WITH VIEW AND WORDWISE

STARdataBASE . . . . £75.00 inc VAT
Post & Packing . . . . £2.30 inc VAT

GCC (Cambridge) Limited
66 High Street, Sawston, Cambridge CB2 4BG
Telephone: Cambridge (0223) 835330/834641
Telex: 81594 SAWCOM

The STARdataBASE, professional database management system consists of a 16K ROM, 75K of extension and applications software on disc and a comprehensive manual. STARdataBASE was designed by a chartered electronics and software engineer to meet a defined need for a high specification, disc-based random access system. The database program and extension software is "Menu Driven", particularly user-friendly, quick to use, but extremely powerful and totally flexible. It is absolutely ideal for use in schools, clubs, colleges, small businesses and in the home.

- Menu driven throughout.
- True random access – will locate a single record in under 2 seconds typically.
- File size is the capacity of a single disc (double sided if applicable) – minus 5K.
- Maximum number of fields 69.
- Maximum number of characters in a field 255 (subject to an overall maximum of 920 in a record).
- Up to 4096 records.
- Records are displayed and input on a "Card Screen" principle. Layout of fields and headings are built entirely by the user by "Painting" on the screen (colour can be used).
- Direct and rapid entry of data into all or selected fields.
- Search criteria include =, <>, >, <, a wildcard "anywhere in the field", and "nowhere in the field".
- Unlimited and rapid sequential searching and selecting.
- All or selected records can be sorted to a depth of 5 fields, each in ascending or descending order, using the super-fast Shell-Metzner routine.
- Instant print or display review of file, giving breakdown of record size, number of records possible and space used, etc.
- Address and label printing (up to 8 across the page).
- Selected whole or part field-content printing with left and right justification, tabbing, and embedded printer control codes possible. Can be mixed with text for invoices and form letters etc.
- Can be configured for any printer (including £ character definition).
- Continuous or paged printout with page heading and pause at each page end, with "New Page" prompt.
- Repeat printing for multiple copies.
- Percentage and arithmetic operations to specified decimal placing.
- Field statistics giving totals, mean and standard deviation, variance and squares.
- Optional password lock.
- Full documented routines (in Basic) which can be spooled onto user-written tailor-made file management and interrogation programs.

STARdataBASE meets and even exceeds professional standards found on large installations. It is unique in speed, flexibility, and ease of use.

Trade and local authority enquiries welcome.
Prices correct at time of going to press.

BEEBED

THE ULTIMATE BASIC PROGRAM EDITOR

As good as Basic is it has one omission — a true program editor. BEEBED solves this by integrating into Basic a true full screen editor which operates directly on the program in memory with no conversion to and from the BBC Basic format.

Features include:—

* True screen scrolling up and down (fast and smooth with new lines appearing at the top and bottom of the screen).
* Insert or Overwrite character input at any point on the screen.
* Character delete to the left and right.
* Copy function to and from any point on the screen (depending on mode inserting or overwriting).
* Global or Selective search/replace, selective mode displays a section of program with the cursor on the exact spot of the string found (all characters, Basic keywords, part keywords and wild cards can be used in the strings).
* Save, Insert, Move, Copy and Delete sections. All of these sections are automatically renumbered when reinserted to keep the program line numbers sequential.
* Very fast partial or total renumber.
* User friendly, all controls are very easy to use, large block cursor for editing, options are menu driven with helpful prompts, all inputs have sensible default values and "are all thoroughly idiot proofed".
* Strings for filenames, search and replace are remembered and can be used again when required.
* Easy insertion of a new line, just press return and a new line appears underneath the cursor with the new line number already calculated.
* Fast cursor movements, locate by line number, undo recent edits, erase to end of line, bad program recover, delete line and more.
* Supplied in 8K EPROM with extensive manual and function key strip.

Requires O.S. 1.2 and Basic I or II.
For further information send an SAE. CHEQUES or PO for £24.50 inclusive made payable to J & O SOFTWARE.

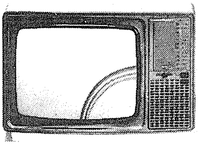
J & O SOFTWARE, 19 LANCASTER RD, NORTHOLT, MIDDX, UB5 4TB.

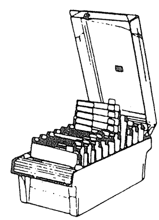
Electronequip

(Authorised BBC Micro Dealer, and Econet service centre)

NORDMENDE

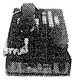
14″ TV/Monitor
£199.00 inc VAT
Remote version
£229.95 inc VAT
Offer includes
RGB/BBC cable 12″ Green Screen **£69.00 inc VAT**





DISC BOXES

30 Disc Box	6.90
40 Disc Box	13.80
45 Disc Box	18.95
90 Disc Box	25.30



MODEMS
Telemod II **£69.95 inc VAT**
Telemod III **£174.99 inc VAT**
Nightingale **£159.85 inc VAT**

DUST COVERS

RX80	5.24
FX80	5.29
KP810	4.60
BBC	4.14
Microvitec	6.09
PW1080	4.60

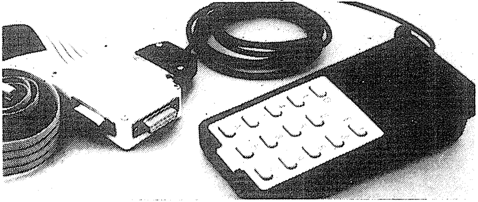
LISTING PAPER
11 × 9.5″ with micro perf's .. £7.50 per 1000

SOFTWARE SALE

most **ACORNSOFT £2.99** inc VAT
most **OTHERS £1.99** inc VAT

VOLTMACE JOYSTICKS

✱10% off list prices

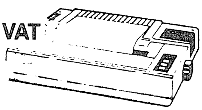


		Discounted price inc VAT
BJVOLT 14	Voltmace Delta 14B Joysticks	13.46
BJVOLTAD	Voltmace 14B/1 Adaptor Box	13.36
BJVOLT3T	Voltmace Delta 3B Twin Joysticks	17.99
BJVOLT3S	Voltmace Delta 3B Single Joystick	11.95

MAIL ORDER ONLY THIS ITEM.

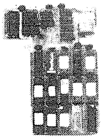
EPSON

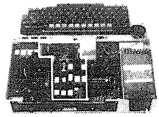
RX-80FT £258.75 inc VAT
FX-80 £377.20 inc VAT



Printer Pack **includes** paper & BBC cable **£8.05 inc VAT**
Screen dump rom available for **£11.50 inc VAT**

SIDEWISE **SIDEWISE FITTED**

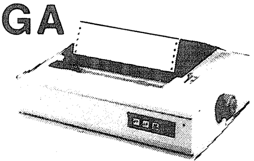




"SIDEWAYS" rom board for BBC Micro.
No soldering required **£39.33 inc VAT**

TAXAN/KAGA KP810

£289.00 inc VAT



Printer price **includes** paper and BBC interface cable.
Screen dump rom available for **£11.50 inc VAT**

SPECIAL OFFERS

800K disc drive switchable with power supply, Plinth and Acorns DNFS
only £409.00 inc VAT

400K single disc drive not P.S.U. and Acorn DNFS **only £277.73 inc VAT**

All prices include VAT

Access VISA

BRANCHES
MAIL ORDER: 36-38 WEST STREET
FAREHAM, HANTS. (0329) 230670
SHOP/TECHNICAL: 59 WEST STREET
FAREHAM, HANTS. (0329) 230671
KINGS LYNN: 17 TENNYSON AVE. (0553) 3782

Postage **46p**. Securicor **£4.90** (per parcel) inc. VAT.
All prices include VAT
Prices subject to variation without prior notification
All offers subject to availability
TRADE, EDUCATIONAL & HMG ORDERS
(Phone for discounts)
Large Stocks – 24 Hour Despatch

MAIL ORDER ENQUIRIES TO BELOW OFFICE

ACORN COMPUTER

Electronequip
59 West Street, Fareham, Hants (0329) 230671

BBC

BULLETIN BOARDS

CORRECTION

The number printed for The Gnome at Home with last month's article on bulletin boards is incorrect. The service is also now operating on a new number which is 01 888 8894. Please note this change.

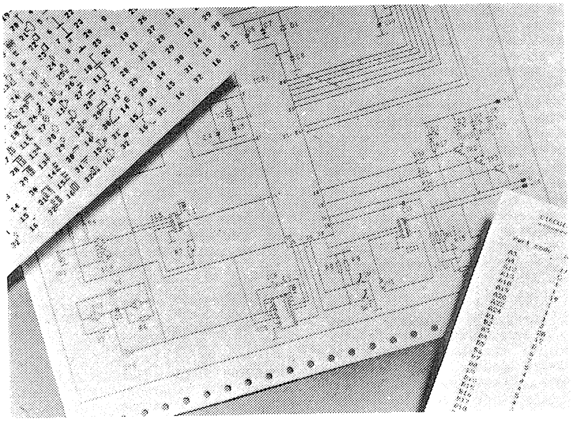
ADVERTISING IN BEEBUG

For advertising details please contact:
Phyllida Vanstone
on
St. Albans
(0727) 62613
or
Lynn Lloyd
on
(0923) 677222
or
St. Albans (0727) 62315

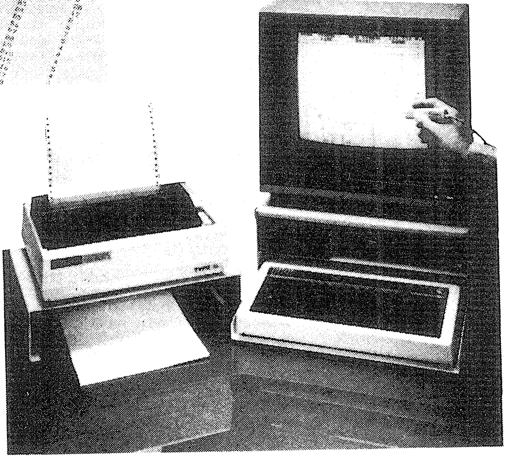
or write to:
P.O. BOX 50
St. Albans
Herts.

CIRKWIK

Schematic drawing on the BBC micro



A lightpen or trackerball driven CAD package orientated to the production of schematic drawings, such as circuit diagrams, flow charts, power systems, pipework diagrams, fluid logic diagrams and many similar professional and engineering applications.



* Lightpen or trackerball driven.
* Virtual screen 8x the BBC's mode 4
* Uses standard dot matrix printer in dual-density graphics mode to produce excellent quality diagrams.
* Automatic parts list generation
* Up to 640 symbols may be in use in any one diagram
* Total symbol library unlimited in size
* Create your own symbols with the lightpen on a highly magnified scale
* Many electronic symbols already included in package.

Datapen

CIRKWIK Program Lightpen only version	£19.95
CIRKWIK Program Trackerball/lightpen version	£24.95
CIRKWIK Program Grafpad/lightpen version	£24.95
DATAPEN Lightpen	£25.00
MARCONI Trackerball	£59.50
ATARI Trak-Ball	£19.95
ATARI BBC Interface	£13.00
MOUSE Interface	£13.00

S.A.E. for details of lightpen, CIRKWIK and other programs.

DATAPEN MICROTECHNOLOGY LTD. Dept. BG3, Kingsclere Road, Overton, Hants. RG25 3JB
TEL: (0256). 770488

Rock steady vision at a rock bottom price

£199.95 (INC. VAT)

CALL 0737 65080 FOR GENEROUS EDUCATION DISCOUNTS

The only thing that seems to stand still on some colour monitors is the price. But at Opus we've now made our RGB High Resolution colour monitor even more affordable – it's down to just £199.95, and the picture quality is as rock steady as ever.
What's more we deliver within 48-hours and give you an unrivalled warranty plan to guarantee your peace of mind for the next two years. And if that's not enough our price includes carriage, VAT and connection leads.
Our JVC monitor (580×470 pixels) has received ample praise from the computer press. Acorn User said:
"... gave steadier pictures than the Microvitec Cub, and the colours were rather better."

Personal Computer News added:
" There is no doubt that the JVC range of ECM colour monitors is excellent value for money... there is no loss in quality of picture after long periods ..."
The unit has a 14" screen and is suitable for use with the BBC Micro, Electron, Sinclair QL, Lynx, Oric, Apple, IBM, Amstrad and RML 480Z and most other leading micros.
Simply post the coupon below to: Opus Supplies Limited, or telephone us quoting your credit card number.

*** While stocks last**

To: Opus Supplies Ltd,
55 Ormside Way, Holmethorpe Industrial Estate, Redhill, Surrey.

Please rush me the following:
(PRICES INCLUDE VAT AND CARRIAGE)

____ High Resolution colour monitor(s) at £199.95 each (inc. VAT)

I enclose a cheque for £____________ or please debit my credit card account with the amount of £____________
My Access □ Barclaycard □ (please tick) no. is:

My computer is____________

Name____________

Address____________

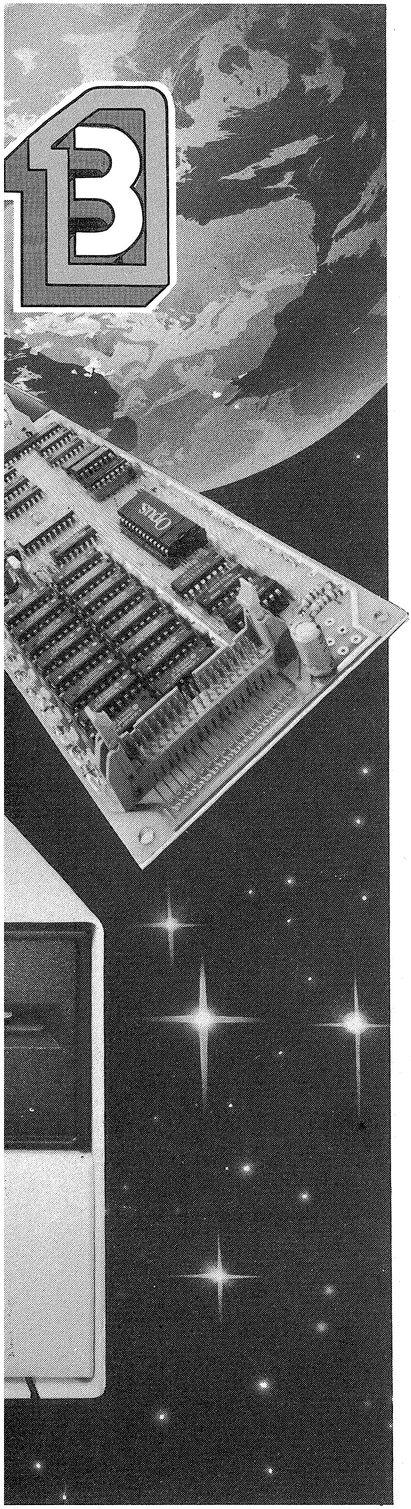
____

Telephone____________

Opus.
Opus Supplies Ltd.

55 Ormside Way,
Holmethorpe Industrial
Estate, Redhill, Surrey.
TELEPHONE: 0737-65080

CHALLENGER
Opus
Opus.
Opus.
CHALLENGER 3



Simply light years ahead of the opposition

Opus announce the biggest disc drive breakthrough for years – CHALLENGER. It could make every other disc system for the BBC micro outdated overnight.

An amazing 3-in-1 design has produced Britain's first complete plug-in disc system for the BBC computer. No more complicated interface connections, soldering or track cutting, Challenger is an all-in-one disc system ready to plug straight into your computer. It's the simplest disc link-up ever and a winning combination that no other system can match.

5¼" DOUBLE-SIDED 40/80 TRACK DISC DRIVE

DOUBLE DENSITY DISC INTERFACE

SUPER FAST SOLID STATE RAM DISC

– Up to 20 times faster than disc drive and providing a massive 256 K of RAM

ALL THIS IN A SINGLE UNIT FOR AN AMAZING £249.95

And just look at the storage capacity of this remarkable stand-alone system.

Our 5¼" disc drive and built in double-density disc interface – an extended version of the highly praised Opus DDOS, voted fastest system by 'Acorn User' and favourite choice of 'A & B Computing' – combine to give you a powerful 1 Megabyte system. But Challenger has a third dimension, a super fast Solid State RAM DISC for instant access to an additional 256 K of RAM. Effectively a lightning fast second disc drive which can be upgraded to ½ Megabyte capacity for only £49.95. It makes Challenger superior to a dual disc drive, giving faster access and less wear and tear on your drive.

Challenger uses the same disc controller chip as the new BBC B+ computer making it compatible with all leading software titles. Call us for our full compatibility listing.

We've provided a selection of utilities on ROM and Challenger's disc drive is 40/80 track switchable. The extended DDOS takes no User RAM, PAGE, the position at which BASIC programs load, is left unchanged, normally staying at &OEOO, and Challenger dynamically moves files on the disc to overcome the 'can't extend' problem.

Challenger's price of £249.95 includes our unbeatable two year warranty, a full operating manual, VAT and free delivery. Call us now on 0737-65080 for details of our nationwide dealer network or simply post the coupon below.

OPUS SUPPLIES THE DRIVE

The Opus disc drive range has few rivals for quality, reliability, and sheer value for money. We've taken the best in Japanese disc drive technology and backed it with Britain's longest guarantee – an amazing two years. For full details of our extensive range of half-height 5¼" disc drives together with information on your nearest stockist, 'phone 0737-65080.

OPUS – BRITAIN'S LEADERS IN DISC DRIVE INNOVATION

To: Opus Supplies Limited, 55 Ormside Way, Holmethorpe Industrial Estate, Redhill, Surrey.
Please rush me the following: (PRICES INCLUDE VAT AND CARRIAGE)

____________________ Challenger Unit(s) £249.95 each

____________________ RAM DISC upgrade(s) to 512K at £49.95 each

I enclose a cheque for £____________ or please debit my credit card account with the amount of £____________

My Access □ Barclaycard □ no. is: □□□□□□□□□□□□□□□□

NAME ____________________

ADDRESS ____________________

____

TELEPHONE ____________________

Opus.
Opus Supplies Ltd.

MEMORIES ARE PRICELESS . . . OURS COST A LOT LESS!

Cumana BBC Microcomputer compatible disk drives: retail price list.

Cumana disk drives supplied with formatting diskette, drive connecting cable and comprehensive user manual. Independent power supply, mains lead and moulded plug included.

5¼ INCH DRIVES

CS100 M 40 Track single sided (100K)	**£114.95**
CS100 40 Track single sided (100K)	**£129.95**
CS400 80 Track double sided (400K)	**£135.95**
CD1200 M 2 x 40 Track single sided (200K)	**£205.95**
CD200 2 x 40 Track single sided (200K)	**£239.95**
CD800/S 2 x 80 Track double sided switchable (800K)	**£255.95**

3½ INCH DRIVES

CS351 40 Track single sided (100K)	**£147.78**
CS354 80 Track double sided (400K)	**£119.95**
CD352 2 x 40 Track single sided (200K)	**£260.83**
CD358 2 x 80 Double sided (800K)	**£219.95**

ALL PRICES INCLUSIVE OF VAT (DELIVERY CHARGES EXTRA)
ALL 80 TRACK 5¼" DRIVES ARE SWITCHABLE TO 40 TRACK MODE
ALL PRICES ARE SUBJECT TO CHANGE WITHOUT NOTICE

Available from the following retail outlets:
W. H. Smith, John Lewis Partnership, Greens Leisure, Laskys, Spectrum UK, Dixons.

Area distributors:
Eltec Home & Business Computers (Bradford) 0274 722512. HCCS Associates (Gateshead) 0632 821924. Hugh-Symons Distribution Services Ltd (Bournemouth) 0202 26535. Kingdom Design (Belfast) 0232 643720. Microworld (Edinburgh) 031 228 1111. National Micro Centre (Stockport) 061 429 8080. J. S. Simnett Computers (Kingston-upon-Thames) 01 541 1495

+ National Dealer Network

5¼ INCH DRIVES (less power supply)

CSX100M 40 Track single sided (100K)	**£89.95**
CSX100 40 Track single sided (100K)	**£99.95**
CSX400 80 Track double sided (400K)	**£114.95**
CDX200M 2 x 40 Track single sided (200K)	**£169.95**
CDX800/S 2 x 80 Track double sided switchable (800K)	**£209.95**
Q.F.S. Double density disk filing system	**£79.95**
Now available double density disk interface for the Electron	**£114.95**

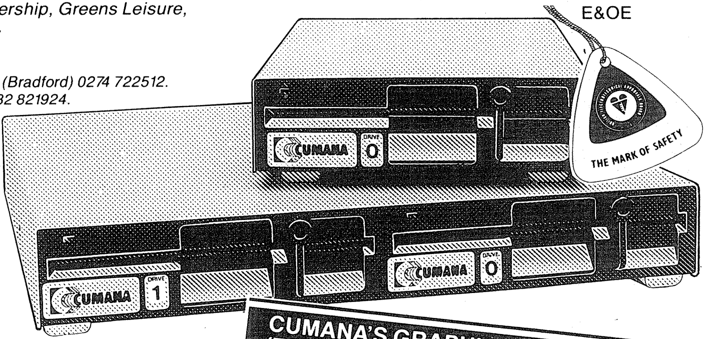
3½ INCH DRIVES (less power supply)

CSX351 40 Track single sided (100K)	**£79.95**
CSX354 80 Track double sided (400K)	**£124.95**
CDX352 2 x 40 Track single sided (200K)	**£139.95**
CDX358 2 x 80 Double sided (800K)	**£214.95**

A selection of peripheral covers, diskettes and data cassettes are also available. For details contact Cumana.

- ☆ Fully assembled and tested before packaging
- ☆ 12 months warranty
- ☆ Attractive hardwearing cabinets in BBC beige
- ☆ CSX/CDX range takes power from BBC micro
- ☆ CS 5¼" drives carry BS 415 safety approval

E&OE



CUMANA'S GRAPHIC PLOTTING AID (TOUCHPAD) AT ONLY £69·95 INC VAT
NEW SOFTWARE LIBRARY AVAILABLE

CUMANA®
The best name in memory

Cumana Limited, Pines Trading Estate, Broad Street, Guildford, Surrey, GU3 3BH. Telephone: Guildford (0483) 503121. Telex: 859380

Loan Repayments

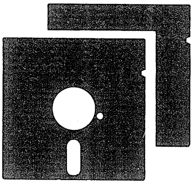
Most of us have to borrow money from time to time. In the short program listed here, Sheridan Williams offers a flexible and easy alternative to the usually tortuous calculations.

Many of us, from time to time, find the need to apply for a loan of some kind. In many cases we may not be sure how much to borrow, or what the repayments will be, or indeed over what period of time the loan should be repaid. This program takes all the hard work out of the calculations involved yet provides great flexibility.

When you run the program, answer the questions with the figures you have in mind, or if you're not sure just press Return. The program will calculate the missing information, and also display a table of repayments if requested.

Two types of loan can be considered: 'variable' where the interest is based on the amount still outstanding, and 'fixed' where the interest is based on the initial sum borrowed, as with many personal loans. In both cases, the program will answer all those 'what if...' questions, and very quickly too. However, the figures are only intended for guidance, though they are reasonably accurate in most instances.

```
   10 REM Program REPAY
   20 REM Version 1.ØB
   30 REM Author Sheridan Williams
   40 REM BEEBUG October 1985
   50 REM Program subject to copyright
   60 :
  1ØØ MODE 7:@%=&2Ø2ØA:VDU14
  11Ø FOR I=1TO2:PRINTTAB(6,I)CHR$131;CH
R$141"Interest repayments":NEXT
  12Ø VDU28,Ø,24,39,4,12
  13Ø A=FNinput("Amount borrowed",TRUE,T
RUE)
  14Ø R=FNinput("Repayments",A,TRUE)
  15Ø IF A=Ø AND R=Ø THEN 14Ø
  16Ø REPEAT
  17Ø I=FNinput("Annual interest rate (%
)",FALSE,FALSE)
  18Ø UNTIL I>=Ø
  19Ø y=FNinput("Number of years loan to
 run",NOT(A=Ø OR R=Ø),FALSE)
  2ØØ IF (A=Ø OR R=Ø) AND y=Ø THEN 19Ø
  21Ø REPEAT
  22Ø n=FNinput("Number of repayments pe
r year",FALSE,FALSE)
  23Ø UNTIL n>Ø
  24Ø PRINT"Variable or Fixed: ";
  25Ø REPEAT:type%=GET:UNTIL INSTR("VvFf
",CHR$(type%))
  26Ø PRINT CHR$(type%AND223):fix%=((typ
e%AND16)=Ø)
  27Ø nr=y*n:i=I/(1ØØ*n):i1=i+1
  28Ø IF i=Ø i2=y*n ELSE i2=(i1^nr-1)/(i
*i1^nr)
  29Ø IF fix% i2=nr/(1+I*y/1ØØ)
  3ØØ PRINT''
  31Ø IF A=Ø A=FNamount:PRINT"Amount bor
rowed = £ ";A
  32Ø IF R=Ø R=FNrepayment:PRINT"Repayme
nt = £ ";R'"Total repaid = £ ";nr*R
  33Ø IF y=Ø y=FNnumberofyears:nr=y*n:PR
INT"Number of years = ";y
  34Ø PRINT'"Table of repayments (Y or N
)? ";:table=(GET AND 223)=89
  35Ø IF NOT table THEN 12Ø
  36Ø VDU12,26:PRINTTAB(4,24)CHR$13Ø"Pre
ss Shift to continue";:VDU28,Ø,22,39,4
  37Ø PRINT"Amount borrowed ";A:IF fix%
A=nr*R
  38Ø PRINT SPC4"Payment"SPC2"To pay"SPC
4"Repayment"
  39Ø FOR x=1 TO nr
  4ØØ IF fix% N=A-R ELSE N=A*i1-R
  41Ø PRINT x,N,A-N
  42Ø IF x MOD n=Ø PRINT"_______ End of
year ";x DIV n;" _______"'
  43Ø A=N:NEXT x:PRINT':REPEAT UNTIL INK
EY(-1):GOTO12Ø
  44Ø END
 1ØØØ DEF FNamount=R*i2
 1Ø1Ø DEF FNrepayment=A/i2
 1Ø2Ø DEF FNnumberofyears
 1Ø3Ø IF R-A*i<Ø THEN PRINT"NEVER REPAID
":=1ØØØØ
 1Ø4Ø IF fix% THEN =A/(R-i*A)/n
 1Ø5Ø =LOG(R/(R-A*i))/(n*LOG(i1))
 1Ø6Ø DEF FNinput(text$,flag,£%)
 1Ø7Ø PRINT text$;:IF flag PRINT" (or Re
turn)";
 1Ø8Ø IF £% PRINT" £"; ELSE PRINT" :";
 1Ø9Ø INPUT " " reply
 11ØØ =reply
```



Double Density

Disc Filing Systems Compared

Much has changed since we first looked at double density disc filing systems back in January 1984 (BEEBUG Vol. 2 No. 8). We asked David Janda to take a fresh look at the whole issue.

Readers who use the standard Acorn DFS (or one of its single density rivals) will be well aware of the limitations that these DFSs have. There are limited file numbers, limited total storage capacity and a lack of speed. A DDFS can, to an extent, overcome most of these problems.

The most annoying problem of the existing Acorn DFS is its limit of only 31 files on one side of a disc. DDFSs can overcome this problem by providing some form of extended catalogue, offering 62 files, or more. 40 track drives have a storage capacity of only 100K per disc surface, and 80 track only 200K. This can also be limiting, especially when data files are used regularly. A DDFS can provide 60% to 80% more storage per side, but 'double' density is a bit misleading!

The speed improvement offered by DDFSs is due to the data being packed more densely on each track. Whereas an ordinary DFS stores data at 10 sectors per track, a DDFS uses 16 or 18 sectors per track. As it is not possible to control the speed at which the drive spins, it is necessary to read/write information that much quicker.

However, readers looking only for a speed improvement need not buy a DDFS. The new Acorn DNFS 1.20 provides a dramatic speed improvement over its predecessor. Similarly Watford's 62-file catalogue may be adequate for your needs.

The level of difficulty of installation of a DDFS depends on what issue board your Beeb has. The DDFS kits all comprise a disc controller board (DCB), an EPROM containing the DDFS itself, and various other TTL chips and header plugs. Fitting a DDFS is similar to fitting a standard DFS. If you own an issue three, or earlier, Beeb it may be necessary to do some soldering.

For the purpose of the review each DDFS was tested on a BBC model B (issue four) with OS 1.2, Basic II, and a sideways ROM board. A 20K Ram card was also fitted to see if there were any problems as far as space is involved.

WATFORD DDFS 1.53

Existing Watford DFS users will have no trouble adapting to Watford's DDFS, as the two products are almost identical in operation. The lack of any new commands for the double density version does not degrade the DDFS in any way - it's still an excellent product.

The disc controller used is the 1770 (as in the B+) which is mounted on the disc controller board (DCB) with three other chips and a few discrete components. Also supplied in the kit are four other chips and a 16K EPROM containing the DDFS.

Documentation supplied with the kit included a 112 page spiral bound DFS/DDFS manual, four page addendum sheet and an A4 sheet of fitting instructions. The DFS/DDFS manual is written for users of Watford's DFS and DDFS, and clearly points out any differences where applicable.

Installing the kit is easy enough if you know what you're doing. The fitting instructions lack detail, and there is no advice on what to do if things go wrong. That said, the DCB is very well made. The extended 40 pins which insert into socket 78 are rigidly supported, and a 40 pin sacrificial socket at the end prevents damage to the socket.

The DDFS provides 80% more storage capacity when used in double density mode. Depending on the number of files selected (31 or 62), 0.5 or 1K will be used up by the directory. The extended file option as well as the density are prompted for when formatting a disc.

As well as offering the standard set of DFS commands, the Watford DDFS provides a number of enhancements and additional commands. Wildcards can be used in *LOAD and *RENAME. With *LOAD the first file in the catalogue will be loaded. A very handy feature is retention of information on the drive, library, directory, density and others over a soft Break.

Extra star commands include MLOAD and MRUN. These enable machine code files to be loaded over the disc work space. *HELP FILES displays information on any opened files, and *HELP SPACE displays free disc space together with the sizes of any 'gaps' on the disc.

In use, Watford's DDFS proved to be very reliable and no compatibility problems were encountered. The reason for this (claims Watford) is an extensive 8271 emulator within the DDFS EPROM which simulates the single density controller. In practice this would appear to work well, as all the protected software ran without trouble.

UDM DDFS 3.10

United Disc Memories has had its DDFS on the market for quite some time. The product has recently been improved and offers a number of features not found in the early versions.

UDM has opted to use the 2793 disc controller chip and unlike most other DDFSs only sixteen sectors per track are used. This means that the double density storage capacity for a 40 track disc is 160K, and 320K for 80 track.

The DCB is a low profile board which is supplied with a 40 pin sacrificial socket. Other parts in the kit include six chips, a header plug and the DDFS EPROM. Also supplied is an eight way DIP switch to set timings for disc access and so on.

UDM also supplied its 100% board for review. This item is mounted in socket 78 and includes two sockets: one for the standard 8271 controller, and the other for the DCB. A flying lead with two-way switch is attached to the 100% board. The board enables switching between the standard Acorn DFS and UDM's DDFS thus giving 100% compatibility.

Installing the DCB and supporting chips is easy enough, but the DIP switches require soldering. One header plug has no directional notch, and needs some guesswork as to which way it should be inserted!

The DDFS can be used in one of three modes: single density, double density or double density with multiple catalogues. The format command requires the drive, density, track and title to be specified, and if double density is chosen the user is asked if extended catalogues are required.

The effect of selecting the extended catalogue mode is to create four catalogues on each side of the disc. Each catalogue is assigned 63.5K and can store 30 files, thus giving a total of 120 possible files per disc. A special locked file called &.&&&&&&& is automatically saved to each catalogue when formatting, and this must not be erased at any cost.

A catalogue is selected using the VOLUME command which is followed by the drive and volume letter (A to D). A nice feature is the ability to automatically save to another catalogue if the current catalogue is full. Loading is facilitated by searching through each catalogue until the correct file is found.

LVL DOS 1.01

The LVL DDFS has also been around for some time. Version 1.01 now includes a formatter and verifier within the DDFS.

	Watford	UDM	LVL	Cumana	Opus
Price	£97.75	£95	£124.95	£69.95	£99.95
Sectors	10/18	10/16	10/18	10/18	10/18
Maximum files	31/62	31/120	31/62	31	31/248
Max file size	disc	255K	255K	255K	252K
Width (mm)	55	60	61	51	45
Length (mm)	79	66	83	81	80
Height (mm)	34	22	27	19	21
Offset (mm)	33	22	42	7	5
Fit with ATPL ROM board	Y	N	Y	Y	Y
Software Compatibility	9	9	3	8	6

Note: Maximum files is the number per side of a disc, not per directory.

The offset is the distance the DCB protrudes to the right of the righthand edge of socket 78.

Nine protected packages were tried on each DDFS. These were Banjax, Castle Quest, Elite (early version), Jet Power Jack, Labyrinth, Magic Mushrooms, Rubble Trouble, Sorcery, and The Mine. The figure given is the number of packages that successfully ran.

When used in its double density mode, the LVL DDFS provides 179K and 359K of usable storage for 4Ø and 8Ø track drives respectively.

The kit contains nine chips, the DCB and the 16K DDFS EPROM. Documentation was in the form of a 34 page manual with a four sided addendum sheet.

Fitting the kit is a little tricky. The DCB is rather large, and inserting it involves bending (very gently) one of the power leads. The pins on the DCB are very weak, and are prone to bending if uneven pressure is placed on the board when inserting it. However, finding the right place to put everything is made easier by a very clear diagram in the packaging, which not only shows where each chip should be inserted, but has the chip itself stuck to that position - you can't go wrong!

The DDFS itself does not incorporate any extra commands (except format and verify), but does get round the 32 file problem in a unique way. Instead of providing 179 or 359K of continuous storage on one side of the disc, each surface (or side) is divided into two drives. This is best explained by an example. A single double-sided disc drive has drive numbers assigned like this:

Disc side	single density	double density
Top	Ø	Ø 4
Bottom	2	2 6

For dual 8Ø track drives in double density, eight drives are provided (numbered Ø to 7). This is very strange but it does mean that there are no extra commands or extensions for the user to learn. However, in use the LVL DDFS proved to be the slowest; and, worst of all, a lot of software would not run.

CUMANA QFS 2.Ø

Cumana - well known for their disc drives - have recently released the new version of their DDFS, or QFS as they call it. Unlike the other DDFSs reviewed, Cumana's DDFS does not offer any extra commands, nor does it allow extra files to be stored, yet it does have its advantages.

The QFS provides 8Ø% extra storage capacity less Ø.5K for the directory when used in double density mode. The only extra command included in the QFS is a formatter which automatically verifies.

The kit is supplied with six chips as well as the QFS EPROM. A seventh chip is attached to the DCB via a flying cable. The DCB is a low profile board which does have its disadvantages, as it's very hard to see if the pins are inserted into the socket correctly.

The formatter used in the QFS is the most comprehensive and user friendly I have ever used. The user is prompted for the drive to be formatted, the number of tracks (8Ø, 4Ø, and 35) and the title and number of the disc. There are options to format both sides of the disc and to format further discs without going through the instructions again. After formatting, the QFS is able to automatically detect the density and number of tracks.

In use, Cumana's QFS proved to be very reliable. It was also one of the fastest, and practically all the test software ran without any trouble.

The simplicity of the whole package makes it very nice to use, yet it could be improved by allowing for a greater number of files.

OPUS DDFS 3.45

The DDFS from Opus can best be described as 'feature packed'. Not only does it allow for up to 248 files per disc surface, but it also includes other useful utilities as well.

The disc controller chip used is the 177Ø, and 355K of user storage is provided. The kit consists of two chips, two header plugs, the DCB, and a 16K DDFS EPROM.

The DCB fits neatly into socket 78 and a reinforcing socket ensures rigid connection. Unlike the others, the Opus DCB is small and compact. Also worthy of note is that the majority of the board faces to the left, so the DCB shouldn't get in the way of other items (ROM boards, etc.) fitted in your Beeb.

To achieve the high number of files possible on one side, the first track is reserved by the DDFS for catalogue and file information. No catalogue can exceed 31 files, so the Opus DDFS divides the disc into five lettered volumes when the disc is formatted in double density mode.

Up to eight volumes may be assigned, and this is done with the VOLGEN command. VOLGEN will display the current volume settings and allow the user to assign or de-assign any of the eight. However, it is necessary to *ENABLE before amending volume as this is a destructive command.

Two commands which I believe are essential in any DDFS are included: these are *4Ø8Ø and *DENSITY. The *4Ø8Ø command allows 4Ø track discs to be used in an 8Ø track drive. The command can be set to on, off or auto. Unfortunately the default is off. The *DENSITY command sets the density that the DDFS software will default to. It is usually automatically selected (which is handy), but many protected games require the density to be set to single, as this option provides.

To actually 'attach' to a particular volume, the *DRIVE command is used. *DRIVE ØC, for example, would set the catalogue to volume C on drive Ø. The method of saving/loading to a particular catalogue is also simple. *LOAD :1G.$.name would load a file called 'name' from volume G on drive 1.

There are, however, a couple of problems with Opus' DDFS. It is not possible to format a disc in double density mode without the default volumes, and, more importantly, a number of packages failed to operate under this DDFS.

THE SYSTEMS COMPARED

The table of software tested will give you some idea as to software compatibility of the various systems.

As far as file capacity is concerned Opus wins hands down with an excellent implementation of the volume handling. Unlike the UDM DDFS, the extra file information is 'hidden' within the first track on the disc, and there is no way that a user can normally corrupt this data. However, UDM has managed to maintain a higher degree of compatibility with the standard Acorn DFS, even without the 1ØØ% board.

Cumana offers a much simpler, faster and cheaper system with no frills, yet a high degree of compatibility and speed. LVL's method of implementing double density is very strange, and its price is high. I don't think it will appeal to many users.

However, my own choice is Watford's DDFS. Even though it does not offer any DDFS like commands and it is only of average speed, it is reasonably priced and is packed with features that have been developed over a long period of time. The extended catalogue option of 62 files would fulfil most requirements, and together with the ability of saving files of any length (up to disc capacity) is a big plus in its favour.

THE BENCHMARKS

To put the various DDFSs through their paces, they were subjected to 13 separate benchmarks. The 13 benchmarks are a measure of the time taken to, respectively, *SAVE and *LOAD 16K, PRINT# and INPUT# 1ØØ integers, PRINT# and INPUT# 1ØØ reals, PRINT# and INPUT# 1ØØ strings, PRINT# and INPUT# 1ØØ records, randomly BPUT# and BGET# 1ØØ bytes, and reverse the order of the contents of a file.

The mean of all 13 benchmarks is given in the table for each of the DDFSs, along with that for Acorn's DFS and DNFS and Watford's DFS. A full table of all the benchmark results is available from the Editorial address on receipt of an SAE. The benchmark program is included in this month's magazine cassette/disc.

MEAN BENCHMARK RESULTS

Acorn 0.90	Acorn DNFS	Watford 1.43	Watford DDFS 1.53		UDM DDFS 3.10	
SD	SD	SD	SD	DD	SD	DD
32.19	16.40	18.66	18.27	16.75	12.99	14.27

LVL DOS 1.01		Cumana QFS 2.0		Opus DDOS 3.05	
SD	DD	SD	DD	SD	DD
19.46	18.88	13.62	14.55	15.18	14.27

BEEBUG WORKSHOP

Storing large quantities of text rapidly eats up the Beeb's limited memory. Surac outlines a technique that reduces the space needed by nearly half.

This month the Workshop looks at one way round the Beeb Adventurer's problem of insufficient memory. If you want to store a lot of text, the 25K or so that is normally available soon gets used up. There is, however, a way of packing text so that you can get up to 50% more in any given space.

First, though, a quick resume of how characters are stored in the Beeb and most other small computers. Each character is allocated one of 256 so-called ASCII codes in the range 0-255, and each code fits neatly into an 8-bit byte. There is also a variable overhead, of at least 5 bytes, for each string in the Beeb.

This is rather wasteful. If you're only displaying text, you don't need 256 characters and can get by with around 60: A-Z, 0-9, plus some punctuation marks. The exact mixture depends on just what text you have. Now, 5 bits can store 32 different character codes and there are 3 groups of 5 bits in two bytes.

However, if we limit ourselves to only 30 characters we can pack three into every two bytes. Unfortunately, 30 is a bit too limiting, so use 60, split them into two groups, and allocate a special code to switch between the groups. Once this is done, all that's left is to find a way of packing three 5-bit codes into two bytes. That is exactly what this first program will do.

```
1000 DEF PROCpackinit
1010 DIM Key$(1),cc%(3)
1020 DIM txtbuf 10000
1030 Key$(0)=
     " ABCDEFGHIJKLMNOPQRSTUVWXYZ.,"
     +CHR$(13)
1040 Key$(1)=
     " 0123456789*;/-=<>()'?$£_;:.,"
     +CHR$(13)
1050 ENDPROC
2000 DEF FNcode(str$,locn%)
2010 LOCAL bitptr%,chcode%,chptr%,
     code%,kptr%,lenstr%,ptr%
2020 chptr%=1
2030 lenstr%=LEN(str$)
2040 REPEAT
2050   chcode%=FNcd(MID$(str$,chptr%,1))
2060  code%=code% OR
       chcode%*2^(10-bitptr%*5)
2070   bitptr%=(bitptr%+1) MOD 3
2080   chptr%=chptr%+1
2090   IF bitptr%=0 THEN PROCpack
2100   UNTIL chptr%=lenstr%+2
2110 IF bitptr% THEN PROCpack
2120 =locn%+ptr%
3000 DEF FNcd(ch$)
3010 LOCAL cc%
3020 IF chptr%=lenstr%+1 THEN =0
3030 cc%=INSTR(Key$(kptr%),ch$)
3040 IF cc% THEN =cc%
3050 kptr%=(kptr%+1) MOD 2
3060 IF INSTR(Key$(kptr%),ch$) THEN
     chptr%=chptr%-1:cc%=31 ELSE
     kptr%=(kptr%+1) MOD 2:cc%=1
3070 =cc%
4000 DEF PROCpack
4010 locn%?ptr%=code% AND &FF
4020 locn%?(ptr%+1)=code% DIV &100
4030 ptr%=ptr%+2
4040 code%=0
4050 ENDPROC
```

PROCpackinit sets up the system, reserving a buffer area to hold the compressed data - in this example 10000 bytes, but that's up to you. It also sets up coding strings in Key$(Ø) and Key$(1). The two strings here are good general-purpose ones, but you can easily change them to suit your own needs.

Very common punctuation, e.g. space, comma and Return, appears in each. This can help the compression, as there is less need to switch between coding sets. The common characters are in the same place in each string. Also, the two strings each hold a maximum of 3Ø characters. Although we can code 32 different values into 5 bits, zero is reserved to mark the end of a string while 31 is the shift code.

FNcode() packs str$ into the buffer starting at location locn%. Line 2Ø6Ø packs the characters, in threes, into code% with FNcd() extracting the code of each in turn. If the character is not in the currently selected coding string, FNcd checks the other; if it finds it, it sends back the shift code (31), otherwise it sets a space. When it reaches the end of the string, FNcd returns a zero.

As every three characters are coded, PROCpack puts their coded values into the buffer. When it reaches the end of str$, any outstanding codes are also stored. On exit, FNcode returns the next free address in the buffer. This suits the best way of setting up a set of compressed strings, as in the following pseudo-Basic:

```
1 PROCpackinit
2 P%=address of start of buffer
3 REPEAT
4   READ a string to code
5   P%=FNpack(string,P%)
6   UNTIL last string coded
7 *SAVE buffer to tape/disc
8 END
```

Don't try to run that! You could read the strings to be packed from DATA statements, the keyboard, or from a tape or disc file; the last is probably the most likely. At the end of the process, your backing store will hold a copy of the compressed text all ready to be unpacked by whatever program is going to use it. If you're writing lots of programs, you'll probably have a standard compression program, to use every time you need it.

The unpacking routines will normally be part of a completely different program from the one which compressed the data.

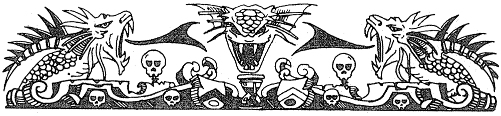
Before you run the procedure you must first have run PROCpackinit and loaded the compressed data into the buffer. Then, by calling PROCdecode() with the start address of the packed string, the data will be expanded and displayed on the screen. I leave you to choose the best way of identifying the start of the particular code you need.

```
5ØØØ DEF PROCdecode(locn%)
5Ø1Ø LOCAL cn%,code%,i%,kptr%,ptr%
5Ø2Ø REPEAT
5Ø3Ø   PROCunpack
5Ø4Ø   FOR i%=1 TO 3
5Ø5Ø     cn%=cc%(i%)
5Ø6Ø     IF cn%>Ø AND cn%<31 THEN
         PRINT MID$(Key$(kptr%),cn%,1);
5Ø7Ø     IF cn%=3Ø THEN VDU1Ø
5Ø8Ø     IF cn%=31 THEN
         kptr%=(kptr%+1) MOD 2
5Ø9Ø     NEXT
51ØØ   UNTIL cc%(1)=Ø OR cc%(2=Ø) OR
       cc%(3)=Ø
511Ø ENDPROC
6ØØØ DEF PROCunpack
6Ø1Ø code%=locn%!ptr% AND &FFFF
6Ø2Ø cc%(1)=code% DIV &4ØØ
6Ø3Ø cc%(2)=(code% AND &3EØ) DIV &2Ø
6Ø4Ø cc%(3)=code% AND &1F
6Ø5Ø ptr%=ptr%+2
6Ø6Ø ENDPROC
```

PROCdecode makes repeated calls to PROCunpack, which reads the next two bytes from the buffer and splits their contents into the corresponding 5-bit codes in cc%(). The codes are then used to extract data from Key$(), shifting between the two character sets every time the code 31 is encountered. As soon as the procedure reaches a zero code it exits as that is the end-of-string marker used.

The decoding program must have exactly the same strings in Key$() as the original packing program. If the strings are different, the most amazing gibberish may appear. There is little point in using these routines unless you need to store a lot of text. They can never quite reach the 5Ø% improvement mark, but get very near if you use a long string which needs as few shift codes as possible.

A complete demonstration of these routines is contained on this month's magazine cassette/disc.



by Mitch

ADVENTURE GAMES ADVENTURE GAMES

This month, the Dungeon Master answers a few enquiries from lost and trapped adventurers.

In the game 'Twin Valley Kingdom' the instructions hint that 'an old, wise man may look upon the secret of life that lies nearby'. The object of the game is to finish with as much wealth as possible then solve the last problem - what to do when you have the maximum amount of points? I have found the Secret of Life and now have the maximum score but I can't find an old, wise man to get him to look at it. Please can you tell me if there is such a man and where I can find him.

Dominic Surman-Wells

To have solved all the problems of this fiendish game and to have found the Secret of Life must have taken a long time and means that by now YOU must be pretty old and wise. It seems likely, therefore, that no better old, wise man should LOOK AT THE SECRET than yourself.

ADVENTURE GAMES ADVENTURE GAMES

I can spare a few moments before I return to feed the dragon. Can you help me with Sphinx Adventure by Acornsoft.
(a) How do you remove the jack from the pit?
(b) How do you cross the Glacier bridge?
(c) How do you get out of the coloured maze and iron passages?

May your sword never go rusty,

David Barnes

Foolish Mortal, the answer to (a) and (b) is close at hand - in fact on your hand, within the Mithril Ring! Only he who wears the ring may cross the bridge or solve the problem of the pit. Remember, waving and rubbing objects is always a good idea.

'But wait!', I hear you moan, 'Where is this accursed ring?' Did your wise old mother never tell you that only fairies wear rings. Go seek the ring in the Fairy King's Grotto, and remember to give him a nice WAVE.

ADVENTURE GAMES ADVENTURE GAMES

I am blind and play text-only adventures via a speech synthesizer (Votrax 'Type and Talk'). I heartily agree with your sentiments regarding mazes! Fearful things, and I am stuck in the Black Maze in 'Castle of Riddles' by Acornsoft. Please give me a clue on what to do. I am sad to hear that more and more adventures are being written with graphics to give location descriptions. It means that I will not be able to play them. I hope that you find it of interest that blind players enjoy adventures as much as anyone.

David Calderwood

Happily the 'Castle' mazes all have clever solutions to their twisty paths. In your case the answer lies with the Metal Rod. Like its predecessor in 'Colossal', this object must be waved. You'll soon have a solution to your problem which like all good fairy stories lies somewhere just by rainbow's end. I should mention that this game cannot be solved without cheating! It appears to be impossible to use the rod for solving the problem of the maze and the Bear's Sitting Room in the same game. Unless a clever troll knows differently...

During my travels I have met a number of blind programmers who have found that the micro has enabled them to compete at work and play in a way I never believed possible.

ADVENTURE GAMES ADVENTURE GAMES

In Bug-Byte's 'Twin Kingdom Valley' how do you get the master key off the Dragon in the North Tower of the Castle?

Philip Macilroy

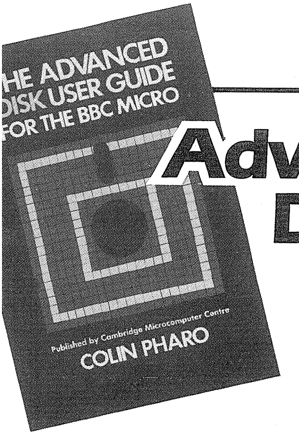
Same way as I get my slippers back from my Dragon - Smash him with a wooden staff.

ADVENTURE GAMES ADVENTURE GAMES

In Program Power's 'ADVENTURE', how do you kill the rat? We've tried everything!

R.G. Petersen

I don't really give a hoot about this problem, but I suggest that you do. Get yourself near the rat and give a loud HOOT. This foolishness will call a hungry owl from the nearby forest to knock the stuffing out of the rodent.


THE ADVANCED DISK USER GUIDE FOR THE BBC MICRO
Published by Cambridge Microcomputer Centre
COLIN PHARO

Advanced Disc User Guide

The Advanced Disc User Guide is the latest in a series of highly acclaimed reference books from the Cambridge Micro Computer Centre. James Fletcher has been taking a look at the latest offering.

The Advanced Disc User Guide, Cambridge Microcomputer Centre, at £14.95.

It is rare to find a technical tome that is written in such an enjoyably readable style as this 446-page Advanced Disc User Guide. I found the first few pages so interesting that I just had to keep on reading. The introduction claims that the book is meant to be read from start to finish, rather than just dipped into as a reference work. Although you would need to be very dedicated to read the whole thing at once, it does make sense to initially skim right through it so as to get some idea of the wealth of 'goodies' that it contains.

The word 'Advanced' in the title is most definitely justified, as some of the concepts used would only be of use to the experienced machine code programmer. However, the book contains such clear and straightforward explanations of the workings of the BBC micro's disc system that it would be well worth a place on the bookshelf of any disc user who wants to know more about the system than the rather meagre amount of information he is offered in Acorn's Disc System User Guide.

The author has succeeded in explaining the workings of the fairly complex 8271 floppy disc controller chip in a remarkably clear manner; although much of the information can be found on the 8271 data sheets it is far from easy to work out what is happening with just the manufacturer's information, and the book uses simple but useful operational flow-charts to help the reader's understanding.

The lengthy but very detailed DFS memory map will appeal to those who just have to know what goes where, and the introduction to the filing system routines and vectors is cleverly done, with each description of a low-level DFS call being accompanied by a fully-documented machine-code example program. Although commands such as OSBGET (read a single byte from a specific file) and OSBPUT (write ditto) are primarily for internal use by the DFS and the machine operating system, you can learn a great deal by using these functions for yourself.

An excellent example of how to build up a random access filing system is given, and the detailed description of a stamp collection catalogue is ideally suited to the beginner, with each step being clearly explained. It is useful to have details of all the various Acorn DFS error messages, although I usually find these less than helpful when something really goes wrong, and the book gives good advice on error prevention and data protection.

Some of the disc utilities provided in the book are both interesting and useful, and I particularly liked the combined 40/80 track formatting program (see also BEEBUG Vol.4 No.4), and the examples of disc overlay techniques which apparently give your computer more memory. The pros and cons of the Acorn DFS are fully discussed, and some of the snags of choosing a non-Acorn DFS are pointed out. Although these can give significant advantages, such as unrestricted numbers of files on each disc, the advice 'regard alternative filing systems as a risk' ties in with my own experiences of some of these products.

All in all then, this is a most useful and interesting book. Perhaps not for the absolute beginner, but for anyone who wants to learn about any aspect of the BEEB's disc system this book is a must. If you can raise the £14.95 - buy it !

This book is available on special offer to BEEBUG members - see price list for full details.

Further Disc Menu Extensions

David Andrews' extended disc catalogue program, described in the June issue of BEEBUG (Vol. 4 No. 2), has provoked considerable feedback. We have put the best of these ideas together to extend this useful utility even further.

The extended disc catalogue program in BEEBUG Vol.4 No.2 provides a menu of your Basic programs on a disc along with a short description of each obtained from a REM statement at the start of each program. Among the many letters of praise for this program that we have received are a number giving further extensions to it. We have taken the best four ideas and present them here as separate additions to the previous program.

New members who have missed the original program can buy a back issue (see page 5Ø) or you can obtain a photocopy of the original article and program by sending an SAE and 3Øp to the Editorial address. Please mark your envelope 'DISCAT photocopy'.

This article provides three extensions to the original program:

1. To allow it to recognise Wordwise and View text files and to load these into the wordprocessors.
2. To recognise *SWAP (Disc Doctor) format discs.
3. To read 62 files on a Watford DFS.

As each part is separate, you need only choose the extensions that you require. Note that the line numbering of the extensions assumes that the numbering of the original program has been left unaltered. The Wordwise and View extensions are combined as one, and this is the extension that we describe first.

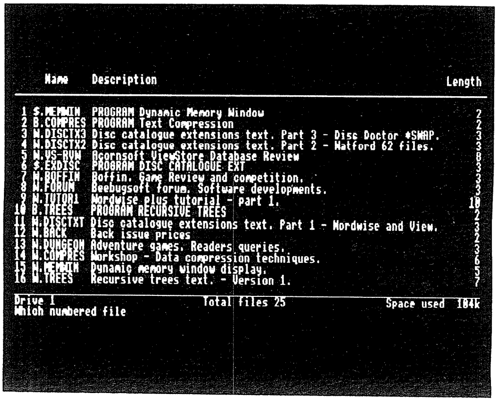
WORDWISE and VIEW

To load text into Wordwise and View, we have to set up a new procedure that will enter the correct wordprocessor and load the text file in the required manner. For View, this requires a *WORD to be executed, followed by 'L abcd' to Load the text file 'abcd'. For Wordwise the program needs to execute a *W. command then enter ':N.' into the wordprocessor, so if Wordwise Plus is in use we will clear the text from memory (although it is unlikely that there will be any there) and in Wordwise this will answer 'N' to the question 'Old Text (Y/N)'. We then enter '2' (Load text) and the filename. This will then load your selected text file into Wordwise.

The following lines are the amendments and additions required.

```
 24Ø PTR#N%=4:F%=BGET#N%:IFF%=244 OR
     F%=81 OR F%=124 THEN 25Ø ELSE 37Ø
 25Ø IFF%=244 P%(T%)=1 ELSE IFF%=124
     P%(T%)=2 ELSE IF F%=81 P%(T%)=3
 26Ø @%=2:NAME$(T%)=NAME$(Z%)
 451 IFP%(f%-1)=2 PROCwwise(1)
 452 IFP%(f%-1)=3 PROCwwise(2)
1470 DEFPROCwwise(z)
1471 IF z=1 THEN 148Ø
1472 A$="*WORD"+CHR$13+"L"+NAME$(f%-1)
     +CHR$13:GOTO 149Ø
148Ø A$="*W."+CHR$13+":N."+CHR$13+"2"
     +NAME$(f%-1)+CHR$13
149Ø FORG%=1TOLENA$:Y%=ASC(MID$(A$,
     G%,1)):X%=Ø:A%=138:CALL&FFF4
1491 NEXT
15ØØ END
```

The text header that is to appear in the disc catalogue must conform to specific guidelines. For Wordwise and Wordwise Plus, the header must be the first item in the file and must start with



```
    Name    Description                                                  Length

 1 $.MEMWIN  PROGRAM Dynamic Memory Window                                    2
 2 B.COMPRES PROGRAM Text Compression                                         2
 3 W.DISCTX3 Disc catalogue extensions text. Part 3 - Disc Doctor *SWAP.      3
 4 W.DISCTX2 Disc catalogue extensions text. Part 2 - Watford 62 files.       3
 5 W.VS-RVW  Acornsoft ViewStore Database Review                              8
 6 $.EXDISC  PROGRAM DISC CATALOGUE EXT                                       3
 7 W.BOFFIN  Boffin. Game Review and competition.                             3
 8 W.FORUM   Beebugsoft forum. Software developments.                         3
 9 W.TUTOR1  Wordwise plus tutorial - part 1.                                10
10 B.TREES   PROGRAM RECURSIVE TREES                                          2
11 W.DISCTXT Disc catalogue extensions text. Part 1 - Wordwise and View.      3
12 W.BACK    Back issue prices                                                2
13 W.DUNGEON Adventure games. Readers queries.                                3
14 W.COMPRES Workshop - Data compression techniques.                          6
15 W.MEMWIN  Dynamic memory window display.                                   5
16 W.TREES   Recursive trees text. - Version 1.                               7
Drive 1                        Total files 25                       Space used  104k
Which numbered file
```

a green embedded command start character (function key 1), followed by two spaces, then *| and finally your title. This must all be on the same line and needs a Return at the end.

For View users, you need to start your text with a 'DM' (Define Macro) stored command. Enter this by using Shift + f8 to edit a command, then type DM followed by Return. Then type 'QQ' to define your macro as 'QQ' and follow that immediately by your heading. Press Return after this and enter the command 'EM' for End Macro. This now completes the extensions for the two Wordprocessors.

DISC DOCTOR *SWAP

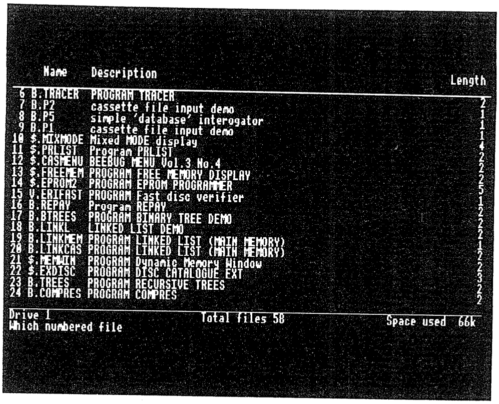
The extensions to allow the disc menu to work with *SWAP catalogues, created with Computer Concept's Disc Doctor, are given below. A single disc surface is treated as two separate discs. After the program prompts you for the disc to be catalogued, it will also ask if this is a *SWAP format disc. If it is then successive catalogues of the disc will alternately produce the two *SWAP catalogues on the disc.

```
 225 IFNAME$(Z%)="!.!!!!!!!" THEN
     V%=V%+512:GOTO370
 385 IF ?BUF%=42 U%=BUF%?1 ELSE U%=32
 400 PRINT"Drive ";D%;" ";CHR$U%;:
     IFU%<>32 PRINT" (*SWAP format)";
1020 PRINTTAB(27,8)"Catalogue of
     Programs"TAB(27,9)STRING$(21,"_")
     TAB(28,11)"Current Drive";SPC(3);
     D%;" ";CHR$U%:Y$=""
1030 IFU%<>32 PRINTTAB(28,13)"Swap
     Files (Y/N)";SPC(2);:Y$=GET$:
     IFY$="Y" PRINT"Yes"
1035 IFY$="Y" PROCoscli("SWAP"):ENDPROC
     :ELSE PRINTTAB(28,13)"New drive"
     SPC(7);:drive%=GET-48
```

To run this program on *SWAP discs, you must title (using *TITLE) one 'half' catalogue with a name starting with '*A' and the other 'half' with one starting with '*B'. The extended program must be saved on both 'halves' of the *SWAP format disc, drive 0. With this amendment, the program will run with both *SWAP and standard format discs.

WATFORD DFS 62 FILES

The appropriate amendments and additions to allow the menu to cope with a



```
     Name     Description                                    Length
  6 B.TRACER  PROGRAM TRACER                                      2
  7 B.P2      cassette file input demo                            1
  8 B.P5      simple 'database' interogator                       1
  9 B.P1      cassette file input demo                            1
 10 $.MIXMODE Mixed MODE display                                  4
 11 $.PRLIST  Program PRLIST                                      2
 12 $.CASMENU BEEBUG MENU Vol.3 No.4                              2
 13 $.FREEMEM PROGRAM FREE MEMORY DISPLAY                         2
 14 $.EPROM2  PROGRAM EPROM PROGRAMMER                            5
 15 V.ERIFAST PROGRAM Fast disc verifier                          1
 16 B.REPAY   Program REPAY                                       2
 17 B.BTREES  PROGRAM BINARY TREE DEMO                            2
 18 B.LINKL   LINKED LIST DEMO                                    2
 19 B.LINKMEM PROGRAM LINKED LIST (MAIN MEMORY)                   1
 20 B.LINKCAS PROGRAM LINKED LIST (MAIN MEMORY)                   2
 21 $.MEMWIN  PROGRAM Dynamic Memory Window                       2
 22 $.EXDISC  PROGRAM DISC CATALOGUE EXT                          3
 23 B.TREES   PROGRAM RECURSIVE TREES                             2
 24 B.COMPRES PROGRAM COMPRES                                     2
Drive 1                      Total files 58         Space used  66k
Which numbered file
```

Watford DFS 62 file catalogue are as follows.

```
 220 FORZ%=0TOR%-1
 360 IFT%=18 OR T%=36 OR T%=54 VDU7:
     PRINT'".. Any key to continue ..":
     IFGET:VDU11:PRINTSPC(27):VDU11,11
 410 PRINTTAB(32)"Total files ";(R%+S%)
     TAB(63)"Space used ";(V%+500)
     DIV1000;"k"
1080 DIMBUF%&400,readcat&50
1160 ?P%=D%:P%!1=BUF%:P%?5=3:P%?6=&53:
     P%?7=0:P%?8=0:P%?9=&24:P%?10=0
1210 S%=0:R%=(BUF%?&105)/8:IF BUF%!&200
     =&AAAAAAAA S%=(BUF%?&305)/8
1220 IF R%<1 PRINT''"No Suitable files.
     "'"Press any key.":G=GET:RUN
1230 DIMNAME$(R%+S%-1),P%(R%+S%-1)
1240 FORZ%=0TOR%-1
1310 NEXT:IF S%=0 THEN 1320
1311 BUF%=BUF%+&200:FORZ%=0TO S%-1
1312 FORY%=0TO6:Q%=R%+Z%
1313 NAME$(Q%)=NAME$(Q%)+CHR$(BUF%?
     (Y%+8*(Z%+1)))
1314 NEXT:G%=BUF%?(Y%+8*(Z%+1))
1315 IF G%>127 G%=G%AND&7F
1316 NAME$(Q%)=CHR$G%+"."+NAME$(Q%)
1317 NEXT
```

These extensions will allow the menu to read both 62 file and normal 31 file catalogues.

With all of these extensions in place, you now have one of the most comprehensive menu programs available for the BBC micro.

Many thanks are due to A.R.Webster, David Andrews, Alan Marshfield, P.J.Swan and David Graham for suggestions and help in compiling these additions.

Looking at Data structures

(Part 2)

Paul Ganney continues his look at data structures by describing how to implement linked lists, and goes on to discuss the principles of tree structures.

Last month we investigated the principles of linked lists and how they operate. Structures like linked lists are not available in languages such as Basic, and so they must be implemented in terms of arrays. The first program does this, and provides insertion and deletion facilities as well as the ability to display the linked list.

```
   10 REM LINKED LIST DEMO
   20 REM VERSION BØ.1
   30 REM AUTHOR  P. GANNEY
   40 REM BEEBUG  OCTOBER 1985
   50 REM PROGRAM SUBJECT TO COPYRIGHT
   60 :
  100 MODE 7
  110 VDU23,1,Ø;Ø;Ø;Ø;
  120 DIM A(1,9)
  130 REM A(Ø, ) IS DATA
  140 REM A(1, ) IS POINTER
  150 REM NULL POINTER IS -1
  160 start%=-1
  170 free%=Ø
  180 A(1,9)=-1
  190 FORI%=ØTO8:A(1,I%)=I%+1:NEXT
  200 :
  210 REPEAT
  220 PROCmenu
  230 UNTIL FALSE
  240 :
 1000 DEF PROCmenu
 1010 PROCheader(1,2,"LINKED LIST")
 1020 PRINT'''CHR$133;"Input choice:"''C
HR$133;"1. Insert an item"'CHR$133;"2. D
elete an item"'CHR$133;"3. List data"'CH
R$133;"4. End"
 1030 REPEAT:A%=GET-48:UNTILA%>ØAND A%<5
 1040 IFA%=1 THEN PROCinsert ELSE IF A%=
2 THEN PROCdelete ELSE IF A%=3 THEN PROC
list ELSE CALL!-4
 1050 ENDPROC
 1060 :
 1070 DEF PROCheader(front%,back%,TEXT$)
 1080 LOCAL spaces
 1090 VDU26,12
 1100 IF LEN TEXT$>33 THEN spaces=Ø ELSE
 spaces=(33-LEN(TEXT$))DIV 2
 1110 FOR I%=Ø TO 1:PRINTTAB(spaces,I%);
CHR$(128+back%);CHR$157;CHR$(128+front%)
;CHR$141;TEXT$;"   ";CHR$156:NEXT
 1120 VDU28,Ø,24,39,3
 1130 ENDPROC
 1140 :
 1150 DEF PROCdelay:TIME=Ø:REPEAT:UNTIL
TIME>1ØØ:ENDPROC
 1160 :
 1170 DEF PROCinsert
 1180 PROCheader(3,4,"INSERT")
 1190 IF free%=-1 THEN VDU7,129:PRINT"LI
ST FULL":PROCdelay:ENDPROC
 1200 INPUT"Data item",A(Ø,free%)
 1210 F%=A(1,free%)
 1220 IF start%=-1 THEN start%=free%:A(1
,start%)=-1:GOTO131Ø ELSE IF A(Ø,free%)
<= A(Ø,start%) THEN A(1,free%)=start%:st
art%=free%:GOTO131Ø
 1230 J%=start%
 1240 K%=FALSE
 1250 REPEAT
 1260 I%=A(1,J%)
 1270 IFI%=-1 THEN K%=TRUE ELSE IF A(Ø,f
ree%)<=A(Ø,I%) THEN K%=TRUE ELSE J%=I%
 1280 UNTILK%
 1290 A(1,free%)=A(1,J%)
 1300 A(1,J%)=free%
 1310 free%=F%
 1320 ENDPROC
 1330 :
 1340 DEF PROCdelete
 1350 PROCheader(5,6,"DELETE")
 1360 IF start%=-1 THEN VDU7,129:PRINT"E
MPTY LIST - NOTHING TO DELETE":PROCdelay
:GOTO145Ø
 1370 INPUT"Data item",D%
 1380 I%=start%
 1390 J%=-1
 1400 K%=FALSE
 1410 REPEAT
 1420 IF I%=-1 K%=TRUE ELSE IF A(Ø,I%)=D
% THEN K%=TRUE ELSE J%=I%:I%=A(1,I%)
 1430 UNTILK%
 1440 IF I%=-1 THEN VDU7,129:PRINT"ITEM
DOES NOT EXIST":PROCdelay ELSE IF J%=-1
THEN S%=A(1,start%):A(1,start%)=free%:fr
ee%=start%:start%=S% ELSE A(1,J%)=A(1,I%
):A(1,I%)=free%:free%=I%
 1450 ENDPROC
 1460 :
 1470 DEF PROClist
 1480 PROCheader(2,3,"LIST")
 1490 IF start%=-1 THEN PRINTCHR$130;"EM
PTY LIST":GOTO155Ø
 1500 I%=start%
 1510 REPEAT
 1520 PRINTA(Ø,I%)
 1530 I%=A(1,I%)
 1540 UNTILI%=-1
 1550 PRINT''"PRESS SPACE":REPEAT:UNTILG
ET=32:ENDPROC
```

PROGRAM NOTES

16Ø-19Ø set up the free space list (at this stage the entire array).

117Ø-132Ø. Insertion. Three 'special cases' are possible. These are tested for first and acted upon should one of them have occurred. Line 119Ø checks for free space. Line 12ØØ inputs the new item into the first free space. Line 122Ø tests if the list is currently empty (start%=-1), and if the new item should be placed at the head of the list. Lines 125Ø-128Ø trace through the list until either the correct place in the list is found or the end of the list is reached. Lines 129Ø-131Ø adjust the pointers as was described in the first article.

134Ø-145Ø. Deletion. The list is first tested for anything to delete (136Ø) and is then traced through (138Ø-143Ø) until either the item is found, or the end of the list is reached. If the item has been found, then pointers are adjusted so that the item before the one to be deleted now points to the one after it, and the deleted item is added to the front of the free space list, its pointer being adjusted to point to the previous head of this list (line 144Ø).

147Ø-155Ø. Listing. Simply traces through the list. I% is the index (place in the array) of the item being listed. After printing the data item (A(Ø,I%)), I% is set to the pointer from this item (A(1,I%)) - the index of the next item. This continues until I% = -1, the terminator.

This version of a linked list, implemented using a two dimensional array, will only work for numeric data. For string data we could use separate arrays for the data and the pointers.

USING MAIN MEMORY

One disadvantage of large arrays is that pressing the Break key automatically clears the array. A better method is to store the data in main memory (adjusting HIMEM so that the O.S. doesn't overwrite it) using indirection operators (see User Guide, p.4Ø9, BEEBUG Vol.3 Nos.7 and 8). The resident integer variables (also unaffected by Break) can also assist further as the pointer variables.

For the purpose of an example, let us assume that we wish to set up a catalogue, each record consisting of an eight character identifier (name), and an integer number. This gives us fifteen bytes per record (8 plus 1 (end-of-string character) for the identifier, 4 for the integer, and 2 for the pointer). Thus, to hold 1ØØ records, we require 15ØØ (&5DC) bytes and hence HIMEM must be set at 3Ø244 (&7624) in mode 7. This section of memory will then be partitioned into the 15-byte blocks by setting all the pointers so as to form the free space list. S% and F% will be used for the start and free space pointers, respectively.

Note that, although !&7BFE (the pointer from the last item) is initially set to -1 (the terminator), all subsequent tests for terminators are for 255, due to the nature of the indirection operators. Note also that the Break key must be programmed to reset HIMEM whenever pressed, in order to preserve the data.

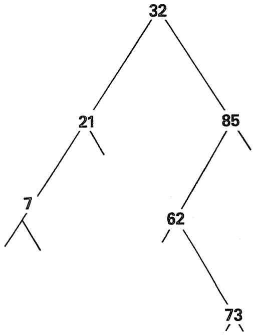
The second program shows how a linked list can be set up in memory and listed. This isn't a complete program, but a full routine, equivalent to the first program, is included on this month's magazine cassette/disc.

```
  1ØØ MODE7
  11Ø VDU23,1,Ø;Ø;Ø;Ø;
  12Ø HIMEM=&7624:S%=255:F%=HIMEM:*K.1ØO
.MH.=&7624MG.6ØM
  13Ø FORI%=&7631 TO &7BEF STEP15:!I%=I%
+2:NEXT:!I%=-1
  14Ø :
 154Ø DEF PROClist
 155Ø PROCheader(2,3,"LIST")
 156Ø LOCAL P%
 157Ø P%=S%
 158Ø VDU14
 159Ø PRINTCHR$13Ø;"Press";CHR$136;"Shif
t";CHR$137;"For next part of list"'
 16ØØ IF P%=255 THEN PRINTCHR$133;"LIST
EMPTY" ELSE REPEAT:PRINT$P%,P%!9:P%=P%?1
4*256+P%?13:UNTIL P%=255
 161Ø VDU15
 162Ø PRINT''"PRESS SPACE":REPEAT:UNTIL
GET=32
 163Ø ENDPROC
```

BINARY TREES

Another linked data structure is the binary tree. The great advantage of the binary tree is that the data structure intrinsically reflects the ordering of the data stored in it. The binary tree has two pointers per element (or node), known respectively as the left and right pointers. The basic principle is that there exists a central node, called the root, which is (of course) at the top of the binary tree. Confused? No need to be, as the ideas are simple enough, it's just the terminology muddling matters as usual.

From the root, the binary tree spreads in two directions, left and right. Those nodes to the left are ordered before the root, those to the right after it. Moving down the binary tree to another node presents the same picture of two pointers. It is this feature that makes binary trees so useful. At any node (position in the binary tree) the structure below this point is also a binary tree (a 'sub-tree'). A node with no sub-trees is called a terminal node. The absence of a sub-tree is recorded by a terminator in the appropriate pointer, in much the same way as the end of a linked list is recorded.



At this point, an example would be useful. Consider the insertion of the number 57 into the purely numeric binary tree in the figure. The process is as follows:

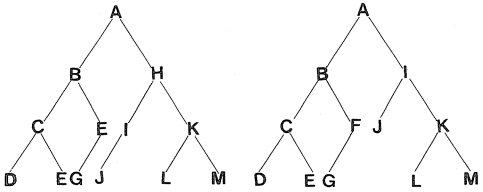
1. 57 is greater than 32, so branch right.
2. 57 is less than 85 so branch left.
3. 57 is less than 62 so branch left.
4. A terminator is reached, so place the data item 57 here, to the left of 62.

Again, you may like to try creating a binary tree from the following items (the first item inserted automatically becomes the root):

32,85,62,21,73,7.

You should end up with the original binary tree.

Having established the principles of inserting an item into a binary tree, what of deletion? This is actually a thornier problem than at first may seem to be the case. The pointer to the deleted node must be made to point to the left sub-tree of the deleted node. The right sub-tree of the deleted node becomes the rightmost element of the left sub-tree of the deleted node. If this sounds confusing (and it is!) follow the process through on the diagram.



DELETE H

POINTER TO H NOW POINTS TO I
I HAS NO RIGHT POINTER, SO THIS
POINTS TO SUB-TREE, K

Listing the data in a binary tree (often called traversing the binary tree) is a simple recursive algorithm:

```
Traverse left sub-tree, if there is
one.
Print node.
Traverse right sub-tree, if there is
one.
```

As BBC Basic supports recursion, there is no problem implementing this routine.

Searching for an item in the binary tree is simply an application of the insertion principles, testing as we go for the required item, outputting an 'Item not found' message if a terminator is reached.

All these features are implemented in the following demonstration program:

```
1Ø REM PROGRAM BINARY TREE DEMO
2Ø REM VERSION BØ.1
3Ø REM AUTHOR  P. GANNEY
4Ø REM BEEBUG  OCTOBER 1985
5Ø REM PROGRAM SUBJECT TO COPYRIGHT
```

```
   60 :
  100 MODE3
  110 ON ERROR REPORT:END
  120 PROCsetup
  130 REPEAT
  140 option%=FNmenu
  150 IF option%=1 THEN PROCinsert
  160 IF option%=2 THEN PROCdelete
  170 IF option%=3 THEN PROCsearch
  180 IF option%=4 THEN PROClist
  190 UNTIL option%=5
  200 MODE 3:END
 1000 DEF PROCsetup
 1010 LOCAL i%
 1020 DIM data$(100),left%(100),right%(1
00)
 1030 root%=-1:free%=0
 1040 FOR i%=0 TO 100
 1050 left%(i%)=-1:right%(i%)=-1
 1060 NEXT i%
 1070 PRINTTAB(31,2)"BINARY TREE DEMO"
 1080 VDU28,0,24,79,4
 1090 ENDPROC
 1100 :
 1110 DEF FNmenu
 1120 LOCAL option%:CLS
 1130 PRINTTAB(10,2)"1. Insert new item"
 1140 PRINTTAB(10,4)"2. Delete item"
 1150 PRINTTAB(10,6)"3. Search for item"
 1160 PRINTTAB(10,8)"4. List all items"
 1170 PRINTTAB(10,10)"5. Exit"
 1180 PRINTTAB(10,14)"Enter your choice"
;
 1190 REPEAT:option%=GET:UNTIL INSTR("12
345",CHR$(option%))
 1200 =option%-48
 1210 :
 1220 DEF PROCinsert
 1230 LOCAL D$:CLS
 1240 IF free%>100 THEN PRINT'"Tree full
.":GOTO 1330
 1250 INPUT''"Enter new data string (max
 6 characters): " D$
 1260 IF root%=-1 THEN root%=free%:GOTO
1300
 1270 found%=FNsearch(D$)
 1280 IF found%>-1 THEN PRINT'D$;" alrea
dy exists in position ";found%:GOTO 1320
 1290 IF turn$="L" THEN left%(lasturn%)=
free% ELSE right%(lasturn%)=free%
 1300 data$(free%)=D$:free%=free%+1
 1310 PRINT'D$;" now entered"
 1320 PRINT'"Press any key to continue":
key%=GET
 1330 ENDPROC
 1340 :
 1350 DEF PROCdelete
 1360 LOCAL found%,P%:P%=root%:CLS
 1370 IF P%=-1 THEN PRINT'"Tree empty":G
OTO 1490
```

```
 1380 INPUT''"Enter data string: " D$
 1390 found%=FNsearch(D$)
 1400 IF found%=-1 THEN PRINT'"Not found
":GOTO 1490
 1410 IF found%=root% THEN root%=left%(r
oot%):GOTO 1430
 1420 IF turn$="L" THEN left%(lasturn%)=
left%(found%) ELSE right%(lasturn%)=left
%(found%)
 1430 P%=left%(found%)
 1440 IF root%=-1 THEN root%=right%(foun
d%):GOTO 1480
 1450 IF P%=-1 THEN left%(lasturn%)=righ
t%(found%):GOTO 1480
 1460 REPEAT:last%=P%:P%=right%(P%):UNTI
L P%=-1
 1470 right%(last%)=right%(found%)
 1480 PRINT'D$;" now deleted"
 1490 PRINT'"Press any key to continue":
key%=GET
 1500 ENDPROC
 1510 :
 1520 DEF PROCsearch
 1530 LOCAL found%,P%:P%=root%:CLS
 1540 IF P%=-1 THEN PRINT'"Tree empty":G
OTO1580
 1550 INPUT''"Enter search string: " D$
 1560 found%=FNsearch(D$):PRINT' D$;
 1570 IF found%>-1 THEN PRINT" found in
position ";found% ELSE PRINT " not found
"
 1580 PRINT'"Press any key to continue":
key%=GET
 1590 ENDPROC
 1600 :
 1610 DEF FNsearch(D$)
 1620 LOCAL found%,P%:found%=FALSE:P%=ro
ot%
 1630 REPEAT
 1640 IF D$=data$(P%) THEN found%=TRUE E
LSE lasturn%=P%:IF D$<data$(P%) THEN P%=
left%(P%):turn$="L" ELSE P%=right%(P%):t
urn$="R"
 1650 UNTIL P%=-1 OR found%
 1660 =P%
 1670 :
 1680 DEF PROClist
 1690 CLS:IF root%=-1 THEN PRINT"Tree em
pty" ELSE PROClist1(root%)
 1700 PRINT'"Press any key to continue":
key%=GET
 1710 ENDPROC
 1720 :
 1730 DEF PROClist1(start%)
 1740 IF left%(start%)<>-1 THEN PROClist
1(left%(start%))
 1750 PRINT data$(start%)
 1760 IF right%(start%)<>-1 THEN PROClis
t1(right%(start%))
 1770 ENDPROC
```

Interfacing the BEEB

Eight books reviewed

With our EPROM Programmer Project we thought it timely to take a look at some of the many books now available on interfacing the Beeb. Geoff Bains has been doing all the reading.

When the BBC Micro was first launched it was much vaunted as having great potential for laboratory work and computer control. The same is true today. Some of that potential has been fulfilled by commercial devices but most is left up to the user with an interest in electronics and skill with a soldering iron.

To help out would-be DIY enthusiasts to construct their dreams, several books have come onto the market. All of these comprise three broad subjects: theory, constructional details, and supporting software. However, the relative proportions of these varies enormously, as do the topics for construction, the depth of discussion, and the skill levels assumed of the reader.

Sensing and Control Projects for the BBC micro by Thomas Nunns. £5.95 from Micro Books.

This book is the simplest of all those reviewed here. The book is divided between projects for the analogue port and those for the user port. The most complex (and appealing) of the ten projects covered is a slot car controller using simple relays and driver chips. The other projects are the like of LED traffic lights and measuring light levels.

Although the projects are, to say the least, uninspiring, the strength of this book is its constructional details. No soldering is required. Everything either screws or clips together. Large lettering, large diagrams and many sections on 'what to do if it doesn't work' mean that this book is ideal for older children or those who have never seen a piece of wire, let alone a chip, before.

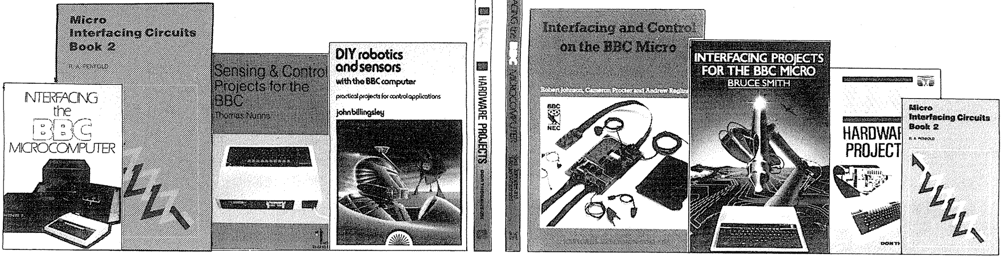
Micro Interfacing Circuits Book 1 and 2 by R.A.Penfold. £2.25 from Babani.

These two slim volumes are the opposite extreme from the previous book. There is next to no constructional details here, nor supporting software either. However, the books cover a wide range of topics - everything from powering LEDs to audio digital filters - and they are meant for other micros as well as the Beeb.

Only the circuits are provided along with a very brief explanation of their workings and pinouts of the various chips and transistors used. Babani books are a well known sight in component shops and their style is pretty uniform, so if you have never come across a Babani electronics book before, the chances are that these are not going to be introductory enough for you anyway.

DIY Robotics and Sensors with the BBC micro by John Billingsley. £6.95 from Sunshine Books.

This book is one of the more specialist tomes. The first half is given over to the run of the mill light pens and relay drivers, even a DIY joystick! The


Micro Interfacing Circuits Book 2
R. A. PENFOLD
INTERFACING the BBC MICROCOMPUTER
Sensing & Control Projects for the BBC
Thomas Nunns
DIY robotics and sensors
with the BBC computer
practical projects for control applications
john billingsley
HARDWARE PROJECTS
INTERFACING THE BBC MICROCOMPUTER
Interfacing and Control on the BBC Micro
Robert Johnson, Cameron Procter and Andrew Ragland
INTERFACING PROJECTS FOR THE BBC MICRO
BRUCE SMITH
HARDWAR PROJECT
Micro Interfacing Circuits Book 2
R. A. PENFOLD

second half is where the rather grandiose title takes over. It is mainly devoted to the control of stepper motors.

I get the impression that this book was written in a rush. There is a lot of theoretical background given and even 'experiments' to prove the fundamental operation of stepper motors. The result is an averaged sized book with an awful lot that you don't actually want. The diagrams, too, are just rough sketches. If you are a dedicated electronics fan with a desire to get into robotics, then this book could provide a starting point. However, don't expect too much in the way of in-depth constructional details. This book really just maps out the terrain.

Interfacing and Control on the BBC micro by R. Johnson, C. Procter, and A. Reglinski. £9.95 from NEC.

This book is essentially a text book for followers of the relevant NEC course. However, it is sold in 'normal' bookshops and so could well tempt the would-be interfacer. Although the book is fairly comprehensive it would not be recommended for most readers.

For a start all the projects are not DIY boards but based around a ready-made circuit board marketed by the NEC. In addition, the book IS a text book and the style does not encourage enthusiasm for the subject. It is also filled with questions and answers which are not really applicable outside the course. That said, the book does offer a reasonable grounding in the use of the Beeb's analogue and VIA ports. However, at nearly £1Ø the price alone is going to put many off.

Interfacing the BBC microcomputer by B. Bannister and M. Whitehead. £7.50 from Macmillan.

Macmillan's Beeb interfacing book can best be described as one for those well familiar with building their own electronics gadgets but have not yet tried this in the computer field.

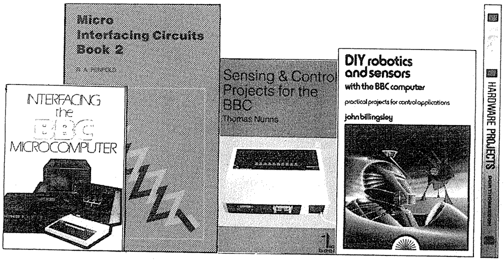
The book starts off with a detailed description of the facilities available. There is a chapter on the user port and one on the 1MHz bus, and one on the handling of analogue data. It is only by page 78 that 'some applications' are discussed. These are not so much applications as an extension of the theory with mention of some specific components. No real constructional details are given; it is assumed the reader knows all about that kind of stuff already.

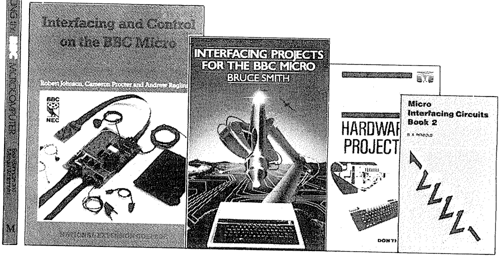
By the time page 1Ø3 comes along the applications are all finished and we are back with the theory in the form of the appendices. These mainly give pages and pages of data sheets from various chip manufacturers. For a near-expert in electronics, unfamiliar with the operation and architecture of a microcomputer, this book would no doubt go down well. However, if you intend to actually build something after reading this, and you have not wielded a soldering iron that much before, let alone designed your own circuit boards, you are likely to be disappointed.

Interfacing the BBC microcomputer by Colin Opie. £8.95 from McGraw-Hill.

Colin Opie's book follows Messrs Bannister's and Whitehead's in more than just title. It too has large sections at the front (about two thirds of the 2ØØ page book) devoted to the explanation of the detailed operation of the user port, 1MHz bus, and the Beeb in general. However, this time the whole is written in a manner that, for me, was a lot more acceptable and encouraging.

Once the hardware projects (the 'experimental hardware') is reached it


Micro Interfacing Circuits Book 2
INTERFACING the BBC MICROCOMPUTER
Sensing & Control Projects for the BBC
Thomas Nunns
DIY robotics and sensors with the BBC computer
practical projects for control applications
john billingsley
HARDWARE PROJECTS


INTERFACING the BBC MICROCOMPUTER
Interfacing and Control on the BBC Micro
Robert Johnson, Cameron Procter and Andrew Reglinski
NEC
INTERFACING PROJECTS FOR THE BBC MICRO
BRUCE SMITH
HARDWAR PROJECT
Micro Interfacing Circuits Book 2

transpires that it is all based around circuit boards that can be purchased, ready made or in kit form, separately (from Watford Electronics). Granted the printed circuit board layouts are given at the end of the book, but it is unlikely that most potential buyers of this book will have the skill or the inclination to start messing around, making their own.

The hardware is advisedly called 'experimental'. There is little in the way of applications here. The boards are potentially very useful but nothing is really said on what and how they could be usefully applied.

Again this is a book for those already knowledgeable in the field. As a reference work for the electronics hobbyist the friendly style and excellent explanations of the theory of the Beeb's ports it is recommended, but the lack of real application and home appeal spoils the book's chances for most of us.

BBC Hardware Projects by Don Thommasson. £8.95 from Melbourne House.

Keeping the best for last, this is one of the two books in this bunch that can really start you off on an interesting and enlightening foray into the world of electronics.

The book starts off with a theory section. This only lasts for a quarter of the book's 200 or so pages but manages to reveal enough of the workings of the Beeb to keep you going for the rest of the book.

Then there is the constructional section. The range of projects is not very inspired but all are laid out on stripboard and so are eminently attemptable by the average Beeb owner. You don't have to be an electrical engineer to manage these.

The projects cover interfacing to the analogue and user ports. There is a light-detector, voltmeter, relay-driver, mains power controller, numeric keypad, and others. Although these may not seem to be great advances in the field of computer control, you do at least leave this book with a feeling that these things are actually possible; even you could achieve this.

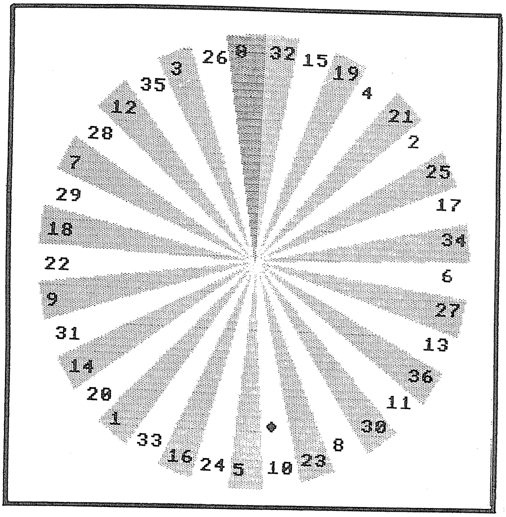
The book does, however, fall down a little on the meagre supporting software. This is a pity as the combination of the excellent Beeb operating system and its interfacing abilities make the system ideal for this kind of thing. Only give half the story and you only have half the fun.

Interfacing Projects for the BBC micro by Bruce Smith. £6.95 from Addison-Wesley.

Like the previous book this one is a practical work. Some theory is given at the start (including a classic padding chapter on hex and binary!) but you soon get into the meat of the projects. These cover a good range - from light pen to EPROM Programmer (yes, other's have done it too!). Each project is accompanied by an adequate description of how the circuit works, a clear stripboard layout diagram, reasonable constructional details, and plentiful supporting software.

It is difficult to know why this book pleases as it does. The presentation is not very good. It is short (only 130 pages), the diagrams are more in the line of sketches and the program listings faint in places. However, please it does. The style is light enough to communicate the author's enthusiasm for the subject and yet it still remains very informative. There is little here for the absolute beginner, but then the book is not aimed at them. If you have a little experience with a soldering iron (or even a lot) you will not go far wrong with this one.

Roulette



Gambling into the night on the tables of Monte Carlo is no more than a dream (or nightmare) for most of us. Richard Pearce brings it all to life with his accurate and colourful implementation.

Can you break the bank at Monte Carlo? Can you turn £5ØØ into £5,ØØØ,ØØØ or will you lose your shirt on the world's most famous gambling table?

Roulette is a one-player game, that runs in mode 1, giving a good display of the betting table and roulette wheel. At the start of the game you are given £5ØØ to gamble, and you can place one or more bets with each bet not exceeding £1ØØ,ØØØ or the amount of money you have remaining. Your aim is to amass a total of £5M and 'break the bank', although as with real life roulette, this is not easy!

When run, the program presents the betting table and a list of options. These options allow you to place bets, and both view and spin the wheel. Please note that when you have completed your bet or bets, you should then select the option to spin the wheel. A brief description of the bets and their odds now follows.

Name	Description	Odds
	(the following bets can include Ø)	
En Plein	Single Number	35-1
A Cheval	Two Numbers (adjacent) on the table, with vertical or horizontal. Zero can be coupled with 1, 2 or 3.	17-1
Trans-versale Plein	Horizontal row of 3 numbers with Ø being combined with any two of 1,2 or 3	11-1
Carre	Block of four numbers	8-1
	(The following cannot include Ø)	
Sixaine	Two adjacent horizontal rows	5-1
Colonne	Vertical column of twelve	2-1
Douxaine	Either the first, second or third dozen numbers, i.e. 1-12, 13-24 or 25-36	2-1
Deux Douxaine	Two dozen. Either numbers 1-24 or 13-36. You must bet an even amount.	1-2

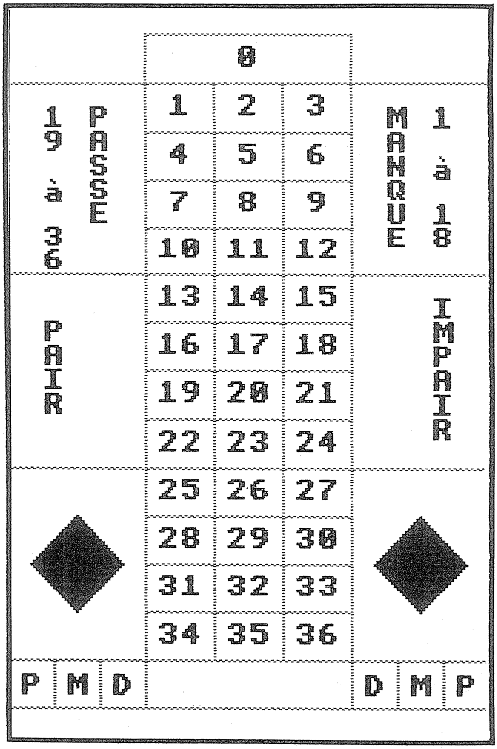
Because of the length of this program and the fact that it runs in mode 1, disc users or anyone who has page set to greater than &EØØ will have to move the program down before running. This can be achieved by using the move down routine published in BEEBUG Vol.3 No.5 or by adding the following lines to your program:

```
1 IF PA.<&EØ1 THEN 1Ø
2 *K.Ø *T|MF.A%=ØTO(TOP-PA.)S.4
:A%!&EØØ=A%|PA.:N.|MPA.=&EØØ|M
O.|MDEL.Ø,4|MRUN|M
3 *FX138,Ø,128
4 END
```

Please take care when typing the program into your computer that no extra spaces are inserted because the program could run out of memory as a result.

```
   1Ø REM PROGRAM ROULETTE
   2Ø REM VERSION BØ.1
   3Ø REM AUTHOR  R.PEARCE
   4Ø REM BEEBUG  OCTOBER 85
   5Ø REM PROGRAM SUBJECT TO COPYRIGHT
   6Ø :
   7Ø ONERRORGOTO19Ø
   8Ø MODE1:PROCsetup
   9Ø REPEAT:CLS
  1ØØ VDU19,2,2,Ø,Ø,Ø
  11Ø VDU23,1,Ø;Ø;Ø;Ø;
  12Ø PROCgame:VDU4,17,3
  13Ø CLS:IFK%=ØGOTO15ØELSESOUND1,2,4,5Ø
:PRINTTAB(5,1Ø)"YOU HAVE BROKEN THE BANK
!"TAB(9,16)"Pretty clever-huh!"TAB(9,18)
"How about lending me"
```

```
  140 PRINT TAB(9,20)"ten quid till Tues
day?":GOTO170
  150 PRINTTAB(7,11)"YOU HAVE NO MONEY L
EFT!":FORI%=30TO0STEP-1:SOUND1,4,I%,2:NE
XT:PRINTTAB(4,14)"You're a pretty awful
gambler"TAB(4,16)"aren't you? I wouldn't
 advise"
  160 PRINTTAB(4,18)"taking your life sa
vings to":COLOUR2:PRINTTAB(4,20)"Monte C
arlo!"
  170 COLOUR1:PRINTTAB(6,24)"Do you want
 another game?":REPEAT:F%=GET:UNTILF%=78
ORF%=89
  180 UNTILF%=78:COLOUR3:END
  190 MODE6:REPORT:PRINT" at line ";ERL
  200 END
  210 :
  220 PRINTTAB(5,3);"EN PLEIN":PRINTTAB(
2,4);"Single Number"
  230 PROCbet:VDU28,23,31,39,10
  240 REPEAT:CLS
  250 INPUTTAB(1,1)"What number?"L%:UNTI
LL%<37
  260 Q%=Q%-B%:IFL%=N%H%=1
  270 PROCwin(35):RETURN
  280 :
  290 PRINTTAB(5,3);"A CHEVAL":PRINTTAB(
4,4);"Two numbers"
  300 PROCbet
  310 PROCnumber
  320 VDU28,23,31,39,16
  330 IFL%=0GOTO360
  340 REPEAT:CLS
  350 INPUTTAB(1,1)"What is the      sec
ond number?"Hi%:UNTILHi%=L%+1ORHi%=L%+3:
GOTO370
  360 REPEAT:CLS:INPUTTAB(1,1)"What is t
he      second number?"Hi%:UNTILHi%>0AND
Hi%<4:GOTO380
  370 IFL%MOD3=0ANDHi%=L%+1GOTO310
  380 Q%=Q%-B%:IFL%=N%ORHi%=N%H%=1
  390 PROCwin(17):RETURN
  400 :
  410 PRINTTAB(3,3);"TRANSVERSALE
 PLEIN":PRINTTAB(2,5);"Horizontal row"
  420 PROCbet
  430 PROCnumber
  440 IFL%=0GOTO480
  450 IFL%MOD3<>1ORL%>34GOTO430
  460 Q%=Q%-B%:IFN%>L%-1ANDN%<L%+3H%=1
  470 PROCwin(11):RETURN
  480 VDU28,23,31,39,15:REPEAT:CLS
  490 INPUTTAB(1,2)"What is the 2nd  num
ber?"L1%:UNTILL1%<3
  500 VDU28,23,31,39,19:REPEAT:CLS
  510 INPUTTAB(1,2)"What is the 3rd  num
ber?"L2%
  520 UNTILL2%<4ANDL2%>L1%
  530 Q%=Q%-B%:IFL%=N%ORL1%=N%ORL2%=N%H%
=1
```



```
  540 PROCwin(11):RETURN
  550 :
  560 PRINTTAB(6,3);"CARRE":PRINTTAB(2,4
);"Block of four"
  570 PROCbet
  580 PROCnumber:IFL%=0GOTO620
  590 IFL%MOD3=0ORL%>32GOTO580
  600 Q%=Q%-B%:IFL%=N%ORL%+1=N%ORL%+3=N%
ORL%+4=N%H%=1
  610 PROCwin(8):RETURN
  620 Q%=Q%-B%:IFN%<4H%=1
  630 PROCwin(8):RETURN
  640 :
  650 PRINTTAB(5,3);"SIXAINE":PRINTTAB(1
,4);"Two horiz. rows"
  660 PROCbet:REPEAT
  670 PROCnumber:UNTILL%MOD3=1ANDL%<32
  680 Q%=Q%-B%:IFN%>L%-1ANDN%<L%+6 H%=1
  690 PROCwin(5):RETURN
  700 :
  710 PRINTTAB(5,3);"COLONNE":PRINTTAB(1
,4);"Vertical column"
  720 PROCbet:REPEAT
  730 PROCnumber:UNTILL%>0ANDL%<4
  740 Q%=Q%-B%:FORI%=0TO11:IFL%+(3*I%)=N
% H%=1
```

```
  75Ø NEXT:PROCwin(2):RETURN
  76Ø :
  77Ø PRINTTAB(5,3);"DOUXAINE":PRINTTAB(
6,4);"Dozen"
  78Ø PROCbet:REPEAT
  79Ø PROCnumber:UNTILL%MOD12=1ANDL%<26
  8ØØ Q%=Q%-B%:IFN%>L%-1ANDN%<L%+12 H%=1
  81Ø PROCwin(2):RETURN
  82Ø :
  83Ø PRINTTAB(1,3);"DEUX DOUXAINES":PRI
NTTAB(3,4);"Two dozen"
  84Ø PROCbet:IFB%MOD2=1B%=B%-1
  85Ø REPEAT
  86Ø PROCnumber:UNTILL%MOD12=1ANDL%<14
  87Ø Q%=Q%-B%:IFN%>L%-1ANDN%<L%+24H%=1
  88Ø PROCwin(Ø.5):RETURN
  89Ø :
  9ØØ PRINTTAB(3,3);"PASSE/MANQUE":PRINT
TAB(4,4);"High/low"
  91Ø PROCbet:REPEAT
  92Ø PRINTTAB(1,6);"Press key for    ch
oice (P/M)"
  93Ø L%=GET:UNTILL%=77ORL%=8Ø
  94Ø IFN%=ØGOTO97Ø
  95Ø H%=Ø:Q%=Q%-B%:IFN%<19ANDL%=77ORN%>
18ANDL%=8ØH%=1
  96Ø PROCwin(1):RETURN
  97Ø PROCØ:IFN1%=Ø:RETURN
  98Ø H%=Ø:Q%=Q%-B%:IFN1%<19ANDL%=77ORN1
%>18ANDL%=8ØH%=1
  99Ø PROCwin(Ø):RETURN
 1ØØØ :
 1Ø1Ø PRINTTAB(2,3);"PAIR/IMPAIR":PRINTT
AB(3,4);"Even/odd"
 1Ø2Ø PROCbet:REPEAT
 1Ø3Ø PRINTTAB(1,6);"Press key for    ch
oice (P/I)"
 1Ø4Ø L%=GET:UNTILL%=73ORL%=8Ø
 1Ø5Ø IFN%=ØGOTO1Ø8Ø
 1Ø6Ø H%=Ø:Q%=Q%-B%:IFL%MOD2=N%MOD2H%=1
 1Ø7Ø PROCwin(1):RETURN
 1Ø8Ø PROCØ:IFN1%=Ø:RETURN
 1Ø9Ø H%=Ø:Q%=Q%-B%:IFL%MOD2=N1%MOD2H%=1
 11ØØ PROCwin(Ø):RETURN
 111Ø :
 112Ø PRINTTAB(4,3);"NOIR/ROUGE":PRINTTA
B(4,4);"Black/red"
 113Ø PROCbet:REPEAT
 114Ø PRINTTAB(1,6);"Press key for    ch
oice (N/R)"
 115Ø L%=GET:UNTILL%=78ORL%=82
 116Ø IFN%=ØGOTO119Ø
 117Ø H%=Ø:Q%=Q%-B%:RESTORE183Ø:FORI%=1T
ON%:READE%:NEXT:IFE%=ØANDL%=78ORE%=1ANDL
%=82H%=1
 118Ø PROCwin(1):RETURN
 119Ø PROCØ:IFN1%=Ø:RETURN
 12ØØ H%=Ø:Q%=Q%-B%:RESTORE183Ø:FORI%=1T
ON1%:READE%:NEXT:IFE%=ØANDL%=78ORE%=1AND
L%=82H%=1
 121Ø PROCwin(Ø):RETURN
 122Ø :
 123Ø PROCwheel
 124Ø MOVE835,1ØØ:PRINT"PRESS SPACE TO":
MOVE835,65:PRINT"RETURN TO MENU":IFGET
 125Ø PROCtable:PROClist
```

```
  85Ø REPEAT
  86Ø PROCnumber:UNTILL%MOD12=1ANDL%<14
  87Ø Q%=Q%-B%:IFN%>L%-1ANDN%<L%+24H%=1
  88Ø PROCwin(Ø.5):RETURN
  89Ø :
  9ØØ PRINTTAB(3,3);"PASSE/MANQUE":PRINT
TAB(4,4);"High/low"
  91Ø PROCbet:REPEAT
  92Ø PRINTTAB(1,6);"Press key for    ch
oice (P/M)"
  93Ø L%=GET:UNTILL%=77ORL%=8Ø
  94Ø IFN%=ØGOTO97Ø
  95Ø H%=Ø:Q%=Q%-B%:IFN%<19ANDL%=77ORN%>
18ANDL%=8ØH%=1
  96Ø PROCwin(1):RETURN
  97Ø PROCØ:IFN1%=Ø:RETURN
  98Ø H%=Ø:Q%=Q%-B%:IFN1%<19ANDL%=77ORN1
%>18ANDL%=8ØH%=1
  99Ø PROCwin(Ø):RETURN
 1ØØØ :
 1Ø1Ø PRINTTAB(2,3);"PAIR/IMPAIR":PRINTT
AB(3,4);"Even/odd"
 1Ø2Ø PROCbet:REPEAT
 1Ø3Ø PRINTTAB(1,6);"Press key for    ch
oice (P/I)"
 1Ø4Ø L%=GET:UNTILL%=73ORL%=8Ø
 1Ø5Ø IFN%=ØGOTO1Ø8Ø
 1Ø6Ø H%=Ø:Q%=Q%-B%:IFL%MOD2=N%MOD2H%=1
 1Ø7Ø PROCwin(1):RETURN
 1Ø8Ø PROCØ:IFN1%=Ø:RETURN
 1Ø9Ø H%=Ø:Q%=Q%-B%:IFL%MOD2=N1%MOD2H%=1
 11ØØ PROCwin(Ø):RETURN
 111Ø :
 112Ø PRINTTAB(4,3);"NOIR/ROUGE":PRINTTA
B(4,4);"Black/red"
 113Ø PROCbet:REPEAT
 114Ø PRINTTAB(1,6);"Press key for    ch
oice (N/R)"
 115Ø L%=GET:UNTILL%=78ORL%=82
 116Ø IFN%=ØGOTO119Ø
 117Ø H%=Ø:Q%=Q%-B%:RESTORE183Ø:FORI%=1T
ON%:READE%:NEXT:IFE%=ØANDL%=78ORE%=1ANDL
%=82H%=1
 118Ø PROCwin(1):RETURN
 119Ø PROCØ:IFN1%=Ø:RETURN
 12ØØ H%=Ø:Q%=Q%-B%:RESTORE183Ø:FORI%=1T
ON1%:READE%:NEXT:IFE%=ØANDL%=78ORE%=1AND
L%=82H%=1
 121Ø PROCwin(Ø):RETURN
 122Ø :
 123Ø PROCwheel
 124Ø MOVE835,1ØØ:PRINT"PRESS SPACE TO":
MOVE835,65:PRINT"RETURN TO MENU":IFGET
 125Ø PROCtable:PROClist
 1260 RETURN
 1270 :
 1280 DEFPROCwheel
 1290 VDU5:VDU26:CLS:MOVE500,500
 1300 FORF%=0TO36:MOVES1%(F%),C1%(F%)
 1310 IFF%=0GCOL0,1
 1320 IFF%MOD2=0GCOL0,1ELSEGCOL0,0
 1330 IFF%=36GCOL0,2:F%=-1
 1340 PLOT85,S1%(F%+1),C1%(F%+1)
 1350 IFF%=-1F%=36
 1360 MOVE500,500:NEXT
 1370 RESTORE1870:FORF%=0TO36
 1380 MOVES2%(F%),C2%(F%):READA$:GCOL0,3
:PRINTA$:NEXT
 1390 ENDPROC
 1400 :
 1410 DEFPROCtable
 1420 CLS:CLG:VDU5:GCOL0,2
 1430 MOVE0,0:MOVE0,1024:PLOT85,736,1024
:MOVE736,0:PLOT85,0,0
 1440 GCOL0,3
 1450 FORI%=0TO3:MOVE32,928-(I%*256):DRA
W704,928-(I%*256):NEXT
 1460 FORI%=0TO1:MOVE32+(I%*672),96:DRAW
32+(I%*672),928:NEXT:FORI%=0TO1:MOVE224+
(I%*288),96:DRAW224+(I%*288),992:NEXT
 1470 FORI%=0TO12:MOVE224,992-(I%*64):DR
AW512,992-(I%*64):NEXT
 1480 MOVE32,96:DRAW704,96
 1490 FORI%=0TO1:MOVE320+(I%*96),160:DRA
W320+(I%*96),928:NEXT:FORI%=0TO1:FORJ%=0
TO1:MOVE96+(I%*64)+(J%*480),96:DRAW96+(I
%*64)+(J%*480),160:NEXT:NEXT
 1500 FORI%=0TO1:MOVE128+(I%*480),224:MO
VE128+(I%*480),352:GCOL0,I%:PLOT85,64+(I
%*480),288:NEXT:FORI%=0TO1:MOVE128+(I%*4
80),224:MOVE128+(I%*480),352:GCOL0,I%:PL
OT85,192+(I%*480),288:NEXT
 1510 MOVE352,976:GCOL0,0:PRINT"0"
 1520 D%=1:RESTORE1830
 1530 FORI%=0TO2:FORJ%=0TO2:D$=STR$(D%):
MOVE256+(J%*96),912-(I%*64):READE%:GCOL0
,E%:PRINTD$:D%=D%+1:NEXT:NEXT
 1540 FORI%=3TO11:FORJ%=0TO2:D$=STR$(D%)
:MOVE240+(J%*96),912-(I%*64):READE%:GCOL
0,E%:PRINTD$:D%=D%+1:NEXT:NEXT
 1550 RESTORE1820:FORI%=0TO2:MOVE48+(I%*
64),140:GCOL0,0:READB$:PRINTB$:MOVE656-(
I%*64),140:PRINTB$:NEXT
 1560 RESTORE1840:FORI%=0TO13:MOVE80,896
-(I%*32):GCOL0,0:READC%:PRINTCHR$(C%):MO
VE624,896-(I%*32):READC%:PRINTCHR$(C%):N
EXT
 1570 RESTORE1850:FORI%=0TO5:MOVE144,896
-(I%*32):GCOL0,0:READD%:PRINTCHR$(D%):MO
VE560,896-(I%*32):READD%:PRINTCHR$(D%):N
EXT
 1580 ENDPROC
 1590 :
 1600 DEFPROCspin
 1610 J%=RND(3)
 1620 FORI%=1TORND(5):FORF%=0TO36:Po%=PO
INT(S%(F%),C%(F%)):GCOL0,3:MOVES%(F%),C%
(F%):VDU255:SOUND0,1,6,1:GCOL0,Po%:MOVES
%(F%),C%(F%):VDU255:NEXT:NEXT
 1630 FORI%=1TOJ%:FORF%=0TO36:Po%=POINT(
S%(F%),C%(F%)):GCOL0,3:MOVES%(F%),C%(F%)
:VDU255:SOUND0,1,6,1:TIME=0:REPEAT:UNTIL
TIME=I%^2:GCOL0,Po%:MOVES%(F%),C%(F%):VD
U255:NEXT:NEXT
 1640 C%=-2:RESTORE1870:REPEAT:READH%:C%
=C%+1:UNTILH%=N%
 1650 FORF%=0TOC%:Po%=POINT(S%(F%),C%(F%
)):GCOL0,3:MOVES%(F%),C%(F%):VDU255:SOUN
D0,1,6,1:TIME=0:REPEAT:UNTILTIME=I%^2:GC
OL0,Po%:MOVES%(F%),C%(F%):VDU255:NEXT
 1660 GCOL0,3:MOVES%(F%),C%(F%):VDU255
 1670 ENDPROC
 1680 :
 1690 DEFPROCball
 1700 PROCspin
 1710 IFN%>0ORN%=0ANDZ%=0GOTO1760
 1720 N%=N1%
 1730 MOVE835,100:PRINT"PRESS SPACE TO":
MOVE935,65:PRINT"RESPIN":REPEAT:UNTILGET
=32:GCOL0,2:MOVES%(F%),C%(F%):VDU255
 1740 VDU28,26,31,39,28:VDU4:CLS:VDU5
 1750 PROCspin
 1760 PROCwinlose:ENDPROC
 1770 :
 1780 DEFPROCwinlose
 1790 IFX%>K%MOVE850,900:PRINT"YOU HAVE
LOST":MOVE950,850:PRINT"£";X%-K%:SOUND1,
4,20,10:SOUND1,4,5,20:ELSE MOVE900,900:P
RINT"YOU HAVE WON":MOVE950,850:PRINT"£";
K%-X%:SOUND1,4,95,2:SOUND1,4,115,2:SOUND
1,4,130,2:SOUND1,4,150,30
 1800 MOVE835,100:PRINT"PRESS SPACE TO":
MOVE935,65:PRINT"CONTINUE":REPEAT:UNTILI
NKEY(1)=32:ENDPROC
 1810 :
 1820 DATA P,M,D
 1830 DATA 1,0,1,0,1,0,1,0,1,0,0,1,0,1,0
,1,0,1,1,0,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0
,1
 1840 DATA 49,49,57,32,32,220,220,32,32,
49,51,56,54,32,32,32,32,73,80,77,65,80,7
3,65,82,73,32,82
 1850 DATA 80,77,65,65,83,78,83,81,69,85
,32,69
 1860 DATA SINGLE NUMBER,TWO NUMBERS,HOR
IZ. ROW,SQUARE,TWO ROWS,COLUMN,DOZEN,TWO
 DOZEN,HIGH/LOW,EVEN/ODD,BLACK/RED,SEE W
HEEL,SPIN BALL
 1870 DATA 32,15,19,4,21,2,25,17,34,6,27
,13,36,11,30,8,23,10,5,24,16,33,1,20,14,
31,9,22,18,29,7,28,12,35,3,26,0
 1880 :
 1890 DEFPROClist
 1900 VDU28,23,31,39,0:VDU4:CLS
 1910 COLOUR2:PRINTTAB(2,Ø);"CHOOSE YOUR
 BET":COLOUR1:PRINTTAB(1,2);"KEY   BET"
 1920 COLOUR3:FORI%=1TO13:PRINTTAB(1,I%+
3)CHR$(I%+64):NEXT
 1930 RESTORE1860:FORI%=1TO13:READG$:PRI
NTTAB(4,I%+3);G$:NEXT
 1940 PRINTTAB(1,2Ø);"Money remaining
 £ ";Q%
 1950 ENDPROC
 1960 :
 1970 DEFPROCmenu:PROClist
 1980 IFQ%=ØENDPROC
 1990 REPEAT:A%=GET-64:UNTILA%>ØANDA%<14
 2ØØØ CLS:COLOUR1
 2Ø1Ø ON A% GOSUB22Ø,29Ø,41Ø,56Ø,65Ø,71Ø
,77Ø,83Ø,9ØØ,1Ø1Ø,112Ø,123Ø,2Ø3Ø
 2Ø2Ø IFSpin%=1ENDPROC ELSEGOTO198Ø
 2Ø3Ø Spin%=1:RETURN
 2Ø4Ø :
 2Ø5Ø DEFPROCbet
 2Ø6Ø VDU28,23,31,39,6:CLS
 2Ø7Ø COLOUR3:REPEAT:CLS
 2Ø8Ø PRINTTAB(1,13);"Money remaining
 £";Q%
 2Ø9Ø INPUTTAB(1,1)"How much do you  wan
t to bet?          £"B%:UNTILB%<1ØØØØ1ANDQ%
-B%>-1
 21ØØ ENDPROC
 211Ø :
 212Ø DEFPROCnumber
 213Ø VDU28,23,31,39,1Ø:CLS
 214Ø REPEAT:CLS
 215Ø INPUTTAB(1,1)"What is the      low
est number    your bet is to   cover",L%
:UNTILL%<36
 216Ø H%=Ø:ENDPROC
 217Ø :
 218Ø DEFPROCgame
 219Ø K%=5ØØ:REPEAT:Q%=K%:X%=K%:Z%=Ø
 22ØØ Spin%=Ø:N%=RND(37)-1
 221Ø PROCtable:PROCmenu
```

```
 222Ø SOUND1,3,1ØØ,25
 223Ø PROCwheel:PROCball
 224Ø UNTILK%>5*1Ø^6ORK%=Ø
 225Ø ENDPROC
 226Ø :
 227Ø DEFPROCØ
 228Ø IFZ%=1ENDPROC
 229Ø N1%=RND(37)-1
 23ØØ IFN1%=ØK%=K%-B%:Q%=Q%-B%:PROClist
 231Ø Z%=1
 232Ø ENDPROC
 233Ø :
 234Ø DEFPROCwin(W)
 235Ø IFH%=1K%=K%+(W*B%)ELSEK%=K%-B%
 236Ø PROClist:ENDPROC
 237Ø :
 238Ø DEFPROCsetup
 239Ø VDU23,255,48,12Ø,252,252,12Ø,48,Ø;
 24ØØ VDU23,22Ø,16,8,Ø,6Ø,6,62,1Ø2,62
 241Ø ENVELOPE1,1,Ø,Ø,Ø,6,3,3,127,-5,-5,
-5,1ØØ,6Ø
 242Ø ENVELOPE2,2,6,Ø,Ø,255,Ø,Ø,126,Ø,Ø,
-126,126,126
 243Ø ENVELOPE3,5,16,12,8,2,1,1,1Ø,-1Ø,Ø
,-1Ø,2ØØ,1ØØ
 244Ø ENVELOPE4,3,Ø,Ø,Ø,Ø,Ø,Ø,127,-1Ø,-5
,-2,12Ø,12Ø
 245Ø PRINTTAB(8,11)"PLEASE WAIT A FEW S
ECONDS"
 246Ø DIMS%(36),C%(36),S1%(36),C1%(36),S
2%(36),C2%(36)
 247Ø F%=Ø:FORA=ØTO36ØSTEP36Ø/37
 248Ø FORA=ØTO36ØSTEP36Ø/37
 249Ø S%(F%)=37Ø*SIN(RAD(A+5))+484:S1%(F
%)=5ØØ*SIN(RAD(A))+5ØØ:S2%(F%)=46Ø*SIN(R
AD(A+5))+473
 25ØØ C%(F%)=37Ø*COS(RAD(A+5))+516:C1%(F
%)=5ØØ*COS(RAD(A))+5ØØ:C2%(F%)=46Ø*COS(R
AD(A+4))+515
 251Ø F%=F%+1:NEXT
 252Ø ENDPROC
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS

LOCAL POINTS OF INTEREST

The keyword LOCAL, followed by one or more variable names, inside a procedure will create local versions of those variables, only accessible inside that procedure. However, if another procedure is called from within the first procedure then the local definitions are good for that second procedure too.

TROUBLE WITH *FX3 - Martin Abernethy

If *FX3,8 is used to select output stream to a printer then VDU1 should not be used to send control codes to the printer as all the 1's are sent as well. If, after a *FX3,8, VDU2 (or Ctrl-B) is also used to enable the printer then a VDU1 not only sends the 1 but the following code twice!

QUICK RANDOM - Roger Burg

RND(X) is comparatively slow in BBC Basic. About twice the speed is the pseudo random function TIME MOD X.

BEEBUG MAGAZINE is produced by BEEBUG Publications Ltd.
Editor: Mike Williams
Assistant Editor: Geoff Bains
Production Editor: Phyllida Vanstone
Technical Assistant: Alan Webster
Secretary: Debbie Sinfield
Managing Editor: Lee Calcraft
Additional thanks are due to Sheridan Williams, Adrian Calcraft, John Yale and Tim Powys-Lybbe.

Editorial Address
BEEBUG
PO BOX 50,
Hollywell Hill,
St. Albans AL1 1EX

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 £40 per page, but please give us warning of anything substantial that you intend to write. A leaflet, 'Notes of Guidance for Contributors' is available on receipt of an A5 (or larger) SAE.

In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using "Wordwise", "View", or other means, but please ensure an adequate written description of your contribution is also included. If you use cassette, please include a backup copy at 300 baud.

HINTS

There are prizes of £5 and £10 for the best hints each month, plus one of £15 for a hint or tip deemed to be exceptionally good.

Please send all editorial material to the editorial address below. If you require a reply it is essential to quote your membership number and enclose an SAE.

SUBSCRIPTIONS

Send all applications for membership, subscription renewals, subscription queries and orders for back issues to the subscriptions address.

MEMBERSHIP SUBSCRIPTION RATES

£ 6.40 6 months (5 issues) UK ONLY
£11.90 UK - 1 year (10 issues)
£18 Europe, £21 Middle East
£23 Americas & Africa, £25 Elsewhere

BACK ISSUES
(Members only)

Vol	Single issues	Volume sets (10 issues)
1	90p	£8
2	£1	£9
3	£1.20	£11
4	£1.20	–

Please add the cost of post and packing as shown:

DESTINATION	First issue	Each subsequent issue
UK	30p	10p
Europe	70p	20p
Elsewhere	£1.50	50p

All overseas items are sent airmail (please send a sterling cheque). We will accept official UK orders but please note that there will be a £1 handling charge for orders under £10 that require an invoice. Note that there is no VAT on magazines.

Back issues are for members only, so it is ESSENTIAL to quote your membership number with your order. Please note that the BEEBUG Reference Card and BEEBUG supplements are not supplied with back issues.

Subscriptions, Back Issues &
Software Address

BEEBUG
PO BOX 109
St. Johns Road
High Wycombe HP10 8NP

Hotline for queries and software orders

St Albans (0727) 40303
Manned Mon-Fri 9am-4.30pm

24hr Answerphone Service for Access and
Barclaycard orders, and subscriptions
Penn (049481) 6666

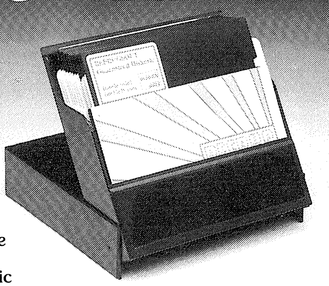
If you require members' discount on software it is essential to quote your membership number and claim the discount when ordering.

DYNAMIC DISCS

TESTED BY BEEBUG

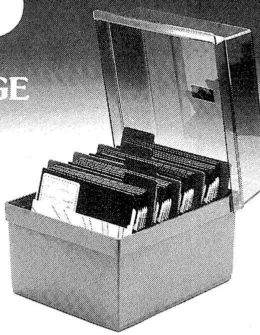
BEEBUG, the largest independent computer user group in the UK, offer 100% tested discs supplied by one of Britain's leading disc manufacturers.

10 FREE LIBRARY CASE



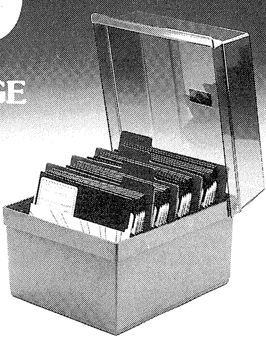
Orders for 10 discs are sent in black plastic library cases.

25 FREE STORAGE BOX



Orders for 25 are delivered in strong plastic Storage box with 4 dividers.

50 FREE STORAGE BOX



Orders for 50 are delivered in strong plastic Storage box with 4 dividers.

COMPARE PRICES

4 Types of Disc To Meet Your Exact Requirement

48 TPI	DOUBLE DENSITY						
10	S/S	D/D	£14.90	10	D/S	D/D	£20.50
25	S/S	D/D	£34.90	25	D/S	D/D	£46.20
50	S/S	D/D	£59.30	50	D/S	D/D	£82.40

96 TPI	DOUBLE DENSITY						
10	S/S	D/D	£20.50	10	D/S	D/D	£21.90
25	S/S	D/D	£46.20	25	D/S	D/D	£49.90
50	S/S	D/D	£82.40	50	D/S	D/D	£93.50

All prices include Storage Box, VAT and delivery to your door (UK)

Suitable for BBC Micro and all other computers using 5¼ inch discs including Atari and Commodore.

Fully Guaranteed – Not only by Beebug but by one of the UK's top disc manufacturers.

We regret that we have had to pass on a slight increase in the price of our discs, but we are now able to offer a wider range to meet your exact requirements.

These discs are the best. Please use the enclosed order form and order from our usual address.

BEEBUG PO BOX 109
St. Johns Road
High Wycombe HP10 8NP.

Official orders are welcome.

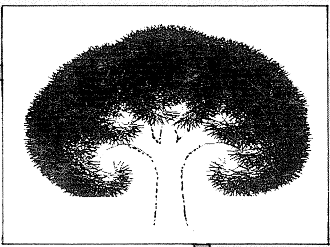
Barclaycard and Access telephone 0494 81 6666
Further information telephone 0727 60263

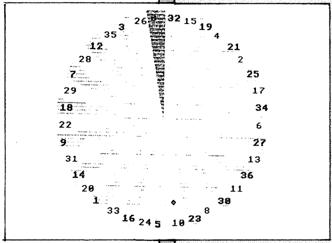
BEEBUG SOFT

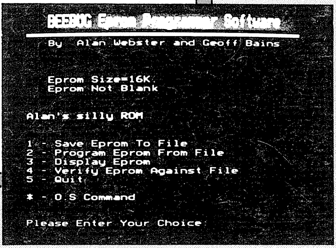
Magazine Cassette/Disc

OCTOBER 1985 CASSETTE DISC CONTENTS

RECURSIVE TREES — displays for the armchair gardener
DYNAMIC MEMORY WINDOW — an eye into your Beeb as it runs your programs
WORDWISE PLUS EXAMPLES — segment programs from the series
LOAN REPAYMENT — calculate how much you owe
EPROM PROGRAMMER DRIVER — blow your own
WORKSHOP PROCEDURES — text compression
DISC MENU EXTENSIONS — the complete disc menu program with all the extensions
DATA STRUCTURES — linked lists and binary trees
ROULETTE — spin the wheel and break the bank
EXTRA FEATURES THIS MONTH
DISC BENCHMARKS — test out your drives with the program used for the DDFS review
3D GRAPHICS SYSTEM DEMO — a fast moving display from Glentop's graphics package







BEEBUG Eprom Programmer Software

```
By  Alan Webster and Geoff Bains


  Eprom Size=16K.
  Eprom Not Blank

Alan's silly ROM

1 - Save Eprom To File
2 - Program Eprom From File
3 - Display Eprom
4 - Verify Eprom Against File
5 - Quit

* - O.S Command

Please Enter Your Choice
```

All this for £3.00 (cass) £4.75 (disc) +50p p&p.
Back issues (disc since Vol. 3 No. 1, cass since Vol. 1 No. 10) available at the same prices.

Subscription rates	DISC UK	CASS UK	DISC O'seas	CASS O'seas
6 months (5 issues)	£25	£17	£30	£20
12 months (10 issues)	£50	£33	£56	£39

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

Cassette subscriptions can be commuted to disc subscription on receipt of £1.70 per issue of the subscription left to run.

All subscription and individual orders to
BEEBUG, PO BOX 109, St. Johns Road, High Wycombe HP10 8NP

Printed in England. DCL 10532