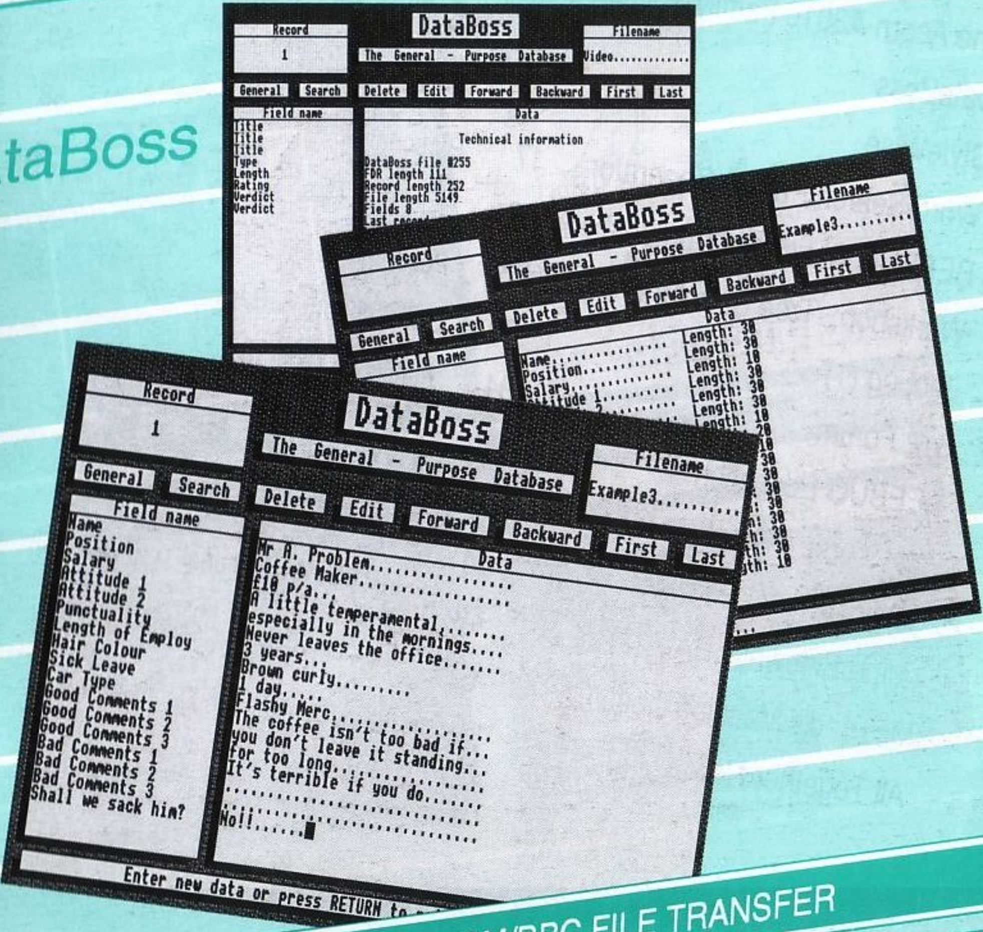
Vol.11 No.7 December 1992

BEEBUG

FOR THE BBC MICRO & MASTER SERIES

DataBoss


Record
1
DataBoss
The General - Purpose Database
Filename
Video............
General Search Delete Edit Forward Backward First Last
Field name
Data
Title
Title
Title
Type
Length
Rating
Verdict
Verdict
Technical information
DataBoss file #255
FDR length 111
Record length 252
File length 5149
Fields 8
Last record

DataBoss
Filename
Example3.........
Record
The General - Purpose Database
Delete Edit Forward Backward First Last
General Search
Field name
Data
Name.................. Length: 30
Position.............. Length: 30
Salary................ Length: 10
Attitude 1............ Length: 30
Attitude 2............ Length: 30
Length: 30
Length: 10
Length: 20

Record
1
DataBoss
The General - Purpose Database
Filename
Example3..........
General Search Delete Edit Forward Backward First Last
Field name
Data
Name
Position
Salary
Attitude 1
Attitude 2
Punctuality
Length of Employ
Hair Colour
Sick Leave
Car Type
Good Comments 1
Good Comments 2
Good Comments 3
Bad Comments 1
Bad Comments 2
Bad Comments 3
Shall we sack him?
Mr A. Problem................
Coffee Maker.................
£10 p/a......................
A little temperamental,......
especially in the mornings....
Never leaves the office.......
3 years...
Brown curly........
1 day.....
Flashy Merc.................
The coffee isn't too bad if...
you don't leave it standing...
for too long...............
It's terrible if you do.......
....
....
No!!......
Enter new data or press RETURN to

- JOBLOG (PART 1)
- IBM/BBC FILE TRANSFER
- ACORN A3010 FAMILY SOLUTION
- STYLEWISE

BEEBUG

Vol.11 No.7 December 1992

FEATURES

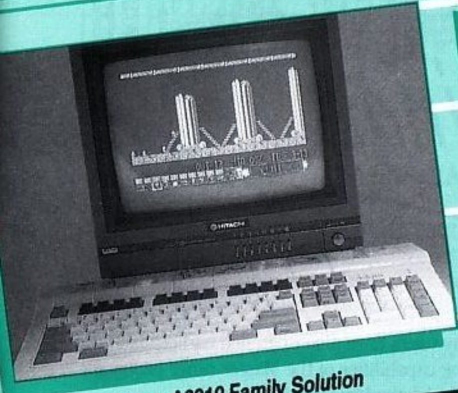
REGULAR ITEMS

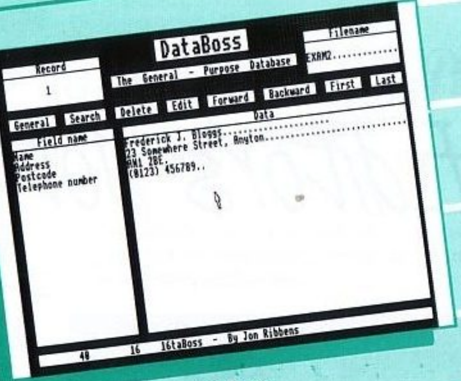
HINTS & TIPS

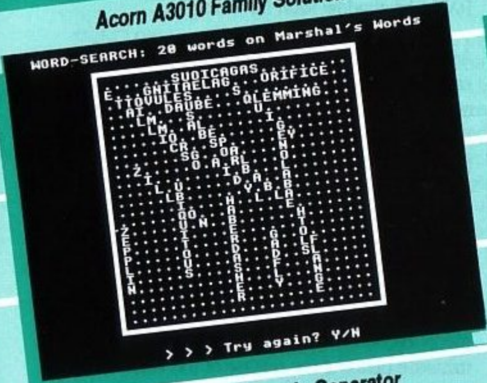
PROGRAM INFORMATION

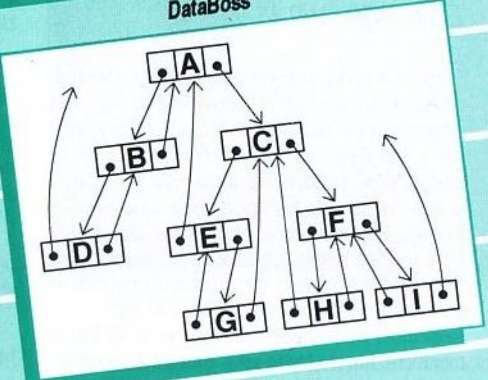
All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly as printed, and checked carefully. When entering a listing, pay special attention to the difference between the digit one and a lower case l (L). Also note that the vertical bar character (Shift \) is reproduced in listings as |.

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

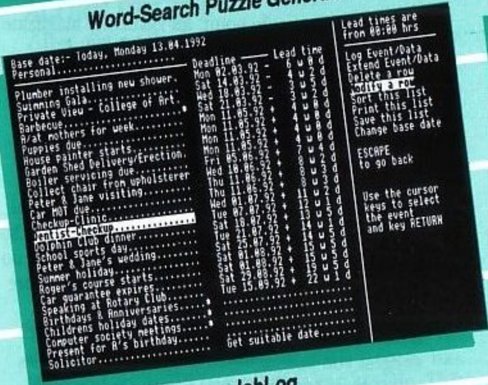

Acorn A3010 Family Solution

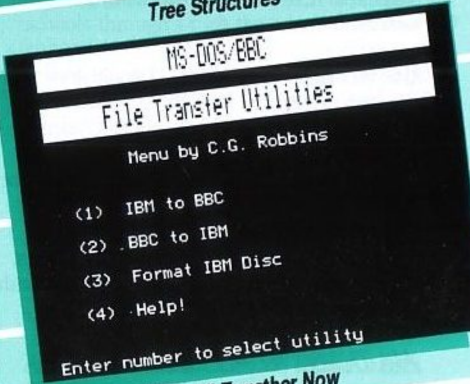

```
DataBoss                                  Filename
Record    The General - Purpose Database  EXAM2............
1
          Delete  Edit  Forward  Backward  First  Last
General  Search                 Data
Field name
Name               Frederick J. Bloggs....................
Address            23 Somewhere Street, Anyton............
Postcode           AN1 2BE.
Telephone number   (0123) 456789..

        40        16     16taBoss - By Jon Ribbens
```
DataBoss


```
WORD-SEARCH: 20 words on Marshal's Words

> > > Try again? Y/N
```
Word-Search Puzzle Generator



Tree Structures


```
Base date:- Today, Monday 13.04.1992                          Lead times are
Personal.....................                                 from 00:00 hrs
                                      Deadline ---- Lead time
Plumber installing new shower.        Mon 02.03.92 -   6 w 0 d   Log Event/Data
Swimming Gala.................        Sat 14.03.92 -   4 w 2 d   Extend Event/Data
Private View - College of Art.        Wed 18.03.92 -   3 w 5 d   Delete a row
Barbecue.....................*        Sat 21.03.92 -   3 w 2 d   Modify a row
A/at mothers for week.........        Mon 11.05.92 +   4 w 0 d   Sort this list
Puppies due...................        Mon 11.05.92 +   4 w 0 d   Print this list
House painter starts..........        Mon 11.05.92 +   4 w 0 d   Save this list
Garden Shed Delivery/Erection.        Mon 11.05.92 +   4 w 0 d   Change base date
Boiler servicing due..........        Fri 05.06.92 +   7 w 4 d
Collect chair from upholsterer        Wed 10.06.92 +   8 w 2 d   ESCAPE
Peter & Jane visiting.........        Thu 11.06.92 +   8 w 3 d   to go back
Car MOT due...................        Thu 11.06.92 +   8 w 3 d
Checkup-Clinic................        Wed 01.07.92 +  11 w 2 d   Use the cursor
Dentist-Checkup...............        Tue 07.07.92 +  12 w 1 d   keys to select
Dolphin Club dinner...........        Sat 18.07.92 +  13 w 5 d   the event
School sports day.............        Tue 21.07.92 +  14 w 1 d   and key RETURN
Peter & Jane's wedding........        Sat 25.07.92 +  14 w 5 d
Summer holiday................        Sat 01.08.92 +  15 w 5 d
Roger's course starts.........        Sat 01.08.92 +  15 w 5 d
Car guarantee expires.........        Sat 29.08.92 +  19 w 5 d
Speaking at Rotary Club......*        Tue 15.09.92 +  22 w 1 d
Birthdays & Anniversaries....*        ........................
Childrens holiday dates......*        ........................
Computer society meetings.....        ........................
Present for A's birthday......        Get suitable date.......
Solicitor.....................
```
JobLog


```
MS-DOS/BBC
File Transfer Utilities

Menu by C.G. Robbins

(1)  IBM to BBC
(2)  BBC to IBM
(3)  Format IBM Disc
(4)  Help!

Enter number to select utility
```
All Together Now

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

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

Program needs at least one bank of sideways RAM.

Program is for Master 128 and Compact only.

Editor's Jottings

EXCHANGE AND MART

A reader was recently extolling to me the benefits of advertising unwanted computers, hardware and software in the pages of BEEBUG magazine. He had found these to be a very effective medium for this purpose. Readers will know that I have also drawn attention to the use of these ads when trying to obtain items (books are one example) which are no longer available new. Another use of these ads is to contact other readers to obtain or exchange information on a particular topic. Remember that all such ads are free to BEEBUG members, but do try to keep your ad as short as possible so that we can fit in as many as we can each month.

Also, on this particular theme, I would draw your attention to the letter from Mr. Kemp in this month's Postbag. If you do have a BBC micro or similar, which is no longer needed, do consider donating it to a school, children's home or maybe a charity who will be able to put it to good use. BBC micros, as readers will be the first to appreciate, can still perform many useful tasks, and can be particularly valuable for children with special needs.

ALL THINGS TO ALL BEEB USERS

We received a letter from another reader in the last few weeks querying another aspect of BEEBUG, and that was which machines we were continuing to support.

I have said before that BEEBUG will continue to cater for users of both the model B and the Master 128 and Master Compact. There are differences between these systems, particularly with regard to later enhancements to the operating system, but many users have also upgraded earlier machines to the capabilities of the Master range (sideways RAM and shadow RAM for example).

As far as I know, the ratio of model B and Master series machines in use by BEEBUG readers is about 50:50. We feel that it is entirely reasonable that some of the articles and programs which we publish are restricted to just some machines or some configurations, but we always try to ensure that every issue has a varied content of interest to all users. We also cater regularly for the Master 512 for those using MS-DOS on this type of system. Regrettably, we no longer have the resources to test programs on the short-lived model B+, which Acorn produced as a stop gap measure pending the release of the Master 128.

To answer a specific point, all BEEBUG magazine discs, and other items such as the *Best of BEEBUG* series published by BEEBUG magazine, will continue to be available in 5.25" DFS format and 3.5" ADFS format as before.

Readers will also be aware that RISC Developments is now responsible for the original Beebugsoft range of products. In this case, stocks of many items are now quite low and there are no plans to print further manuals or duplicate more discs once current stocks run out. If you are interested in any products in this range then I suggest you take advantage of the special offers which have been advertised in the magazine recently (see October or November issues).

The next issue of BEEBUG will be that for January/February, one of our two-month issues, and this is expected to be mailed out in late January 1993. Merry Christmas from everyone at Beebug.
M.W.

HELPING THOSE WITH SPECIAL NEEDS

A query which we receive in the BEEBUG offices from time to time concerns what sources of help are available on the uses of computers (particularly the BBC micro and Master 128) with children and adults with special needs. The *Special Needs User Group* is just such a group, which recently held its AGM.

SNUG (the acronym by which the group is known) operates through a range of regional and specialist groups, details of which are published in SNUG's *The Red Pages*. For example, there is a specific group for the BBC micro, (and others for the Archimedes, the Amiga, and so on), for specialisms (like adults, communications, dyslexia, visual handicaps etc.), and local groups around the country. SNUG also provides discounts from various suppliers to its members.

Anyone requiring further information should contact the secretary, Jeff Hughes, 39 Eccleston Gardens, St. Helens WA10 3BJ, tel. 0744 24608.

ACORN DELAYS

After the announcements of new machines in September (see BEEBUG Vol.11 No.5) supplies of some systems have been very meagre indeed. The low priced A3010 has been available in reasonable quantities, but the A3020 and the A4000 have suffered from a shortage of systems, and the portable A4 and Pocket Book have been almost impossible to obtain. Supplies of all systems have now improved (except for the A4), and all models should now be available from stock (at Beebug Ltd. or from your local dealer).

SCHOOL TO HOME WITH BEEBUG

Beebug Ltd., in conjunction with Acorn, has launched an *A3010 Home Professional* system. This consists of a new Acorn A3010 computer fitted with 2Mb memory (optional 4Mb) and 20Mb IDE hard disc (optionally 60Mb), plus Acorn colour monitor, together with Ovation (RISC Developments' popular DTP package), Pipedream 3 (Colton Software's highly rated spreadsheet, word processor and database), Artisan II (Clares' painting package), plus Chess and Interdictor games and other software to provide a comprehensive system for leisure and more serious applications.

A3010 Home Professional System

The complete system sells for just £899 inc. VAT. Moreover, there is an incentive scheme for schools through which the school receives a £20 cheque for every Home Professional system purchased. Full details of both the A3010 Home Professional System and the associated schools scheme can be obtained from Beebug Ltd., 117 Hatfield Road, St Albans, Herts AL1 4JS, tel. 0727 40303. The scheme runs until 31st January 1993.

BEEBUG OPENING HOURS

Readers should note that the Beebug showroom in St Albans is now open every Thursday evening until 8pm for all your computing needs.

ALL FORMATS COMPUTER FAIRS

The latest All Formats Computer Fairs for December are listed below:

5th Dec	West Midlands: National Motorcycle Museum, near Birmingham NEC (J6 M42).
12th Dec	London: Sandown Park, Esher, Surrey (J9/10 M25).
13th Dec	Wales: University Union, Park Place, Cardiff.

For more information or advance tickets phone 0608 662212.

RISC *New Generation!*

RISC User, the highly popular magazine for Archimedes users, is bigger and better. The new RISC User is now B5 size which offers a sophisticated design, bigger colour illustrations and bigger pages with more information. Altogether better value and no increase in price.

RISC User is still a convenient size to assemble into an easy-to-use reference library, containing all the information you need as an Archimedes user. Every issue of RISC User offers a wealth of articles and programs with professionally written reviews, lively news, help and advice for beginners and experienced users, and items of home entertainment. Altogether RISC User has established a reputation for accurate, objective and informed articles of real practical use to all users of Acorn's range of RISC computers.

ACORN POCKET BOOK
A look into this exciting new palmtop computer.

CABLE NEWS
A review of Lingenuity's package designed to turn your computer into a high tech presentation package

ARM 250: ACORN'S MEGA CHIP
A look into the new mega chip used in Acorn's latest machines.

STATE OF THE ART: CD ROMS
A survey of the latest offerings in the multi-media market.

PC PAGES
A regular series devoted to the PC emulator and to the running of PC software on the Archimedes.

WIMP FUNCTION/PROC. LIBRARY
A collection of functions and procedures providing the necessary building blocks for your Wimp programs.

WRITE-BACK
A readers' section of RISC User for comment, help, information - a magazine version of a bulletin board.

C NOTEBOOK
A series on programming Desktop applications in C.

WP/DTP
Articles on using different DTP and WP packages.

INTO THE ARC
A regular series for beginners.

TECHNICAL QUERIES
A regular column attempting to answer your technical queries.

ARCADE GAMES ROUNDUP
Some of the hottest games available this Christmas.

SUBSCRIPTION DETAILS

As a member of BEEBUG you may extend your subscription to include RISC User for only £10.50 (overseas see table).

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

The Acorn A3010 Family Solution

The recently announced A3010 provides the cheapest ever entry point to the Archimedes range. Mark Moxon explores what this package has to offer for those considering an upgrade.

The first thing that hits you about the new Acorn A3010 is that it looks much nicer than its predecessor, the A3000. The shaping of the computer is much more aerodynamic (for want of a better word), the case is not so deep, and the function keys are now in mellow Acorn green, rather than garish BBC red. The case is also no longer the usual cream, but is "warm grey", and I must say I like it. This, you think, would look nice in your home.

The A3010 playing Lemmings on a normal TV

The A3010 Family Solution (£499 inc. VAT), which I will look at in this review, is aimed squarely at the consumer market, and is being sold through the high street shops in an attempt to capture some of the Christmas market, as well as Acorn dealers like Beebug Ltd. On the other hand, the more expensive A3010 Learning Curve (£799 inc. VAT) is aimed at the same market as the A3000 Learning Curve. As is usually the case, Acorn have come up with an excellent computer at an excellent price, but that's been said before many times. So what is so exciting about this new machine that makes it worth considering?

THE HARD FACTS

The A3010 comes with a host of goodies. Apart from the machine itself there is a mouse, a co-axial lead for connection to a TV, a high density Applications disc, a Support disc, two audio tapes, three manuals, two blank TDK high density discs, and a rather colourful sticker attached to the top of the machine. This can be detached, depending on how loud you want your computer to look.

First of all, let's consider the machine itself. The facts are simple. The A3010 uses an ARM250 RISC processor to deliver a high speed machine, running the RISC OS 3.10 operating system (the very latest version). The machine comes with 1Mb of RAM, upgradable to 2Mb, and has a high density 2Mb floppy disc drive built in to the right-hand side of the case (the same position as on the A3000). The mouse provided is the same as for the recently launched A4, and it has a sleek sloping design which is much more comfortable than the previous Acorn mouse. The mouse is also coloured warm grey to match the case.

On the back of the computer are a TV modulator socket, two joystick ports, a monitor socket (15 pin, as on the A5000), a headphone output, serial and parallel ports and the power switch. The mouse connector is also on the back of the case, which is far more convenient than the socket underneath the A3000, and there's a removable flap at the back where one mini expansion card will fit (the same cards that fit the A3000 will fit the A3010).

WHAT'S NEW?

Physically, therefore, the A3010 represents an improvement on the A3000 in a number of ways. The new TV modulator socket on the back means that the computer can be hooked up to a normal TV screen using the cable provided, just like with the Beeb. The picture quality is certainly usable, but compared to a dedicated monitor the colours crawled and the definition was rather patchy.

Also new are the two joystick ports, which will take standard Atari-style 9-pin joysticks, like the Master Compact.

USING RISC OS 3.10

So, what is it like using the new machine? Well, RISC OS 3.10 has been speeded up over its predecessor, version 3.00, and now the time taken for the machine to boot up from scratch is only just over 10 seconds. The familiar RISC OS Desktop then appears, with a total of 584K of free memory available (with 160K more allocated to the font cache). The applications Alarm, Calc, Chars, Configure, Draw, Edit, Help and Paint are all available in ROM, and all load almost instantly without having to insert a disc into the drive. Thus the computer can be used to produce text and graphics without having to load anything externally.

The high density Applications disc effectively contains the contents of the Applications Discs 1 and 2 supplied with the A5000, with one extra application called TVTest (which helps you set up your TV). See the review of the A5000 (BEEBUG Vol.10 No.6) for more details.

The A3010 comes with three manuals in all, namely the A3010 Welcome Guide, the RISC OS 3 User/Applications Guide, and the RISC OS 3.10 Release Notes. The Welcome Guide describes how to set up the machine, and introduces concepts such as the mouse, windows, discs and printers to those who are new to Wimp computing. As with other Acorn Welcome Guides it is well written, and provides a gentle explanation without getting weighed down with too much jargon. The latter part of the Guide contains details of the A3010 hardware, as the other manuals provided are concerned with the RISC OS operating system, which is generic to all 32-bit Acorn computers. There is handy information on tuning your TV for use with the A3010, and lots of diagrams of the ports on the back.

The User and Applications Guides have been combined into one volume (they are supplied as two separate books with the A5000), bound in normal paperback style rather than the more usual ring bound style. This is not as inconvenient as you might think, although it would be nice to be able to leave the Guide open next to the computer.

The RISC OS 3.10 Release Notes contain details for those upgrading from RISC OS 2 or 3.00, as well as amendments and additions to the other manuals supplied.

Finally there are two audio tapes that come with the machine. These form an introduction to using the RISC OS Desktop, and should be most useful to complete beginners. However, the voice of a man going at a certain pace with no illustrations to hand does seem harder to follow than a very well written manual with lots of illustrations - that's just my personal opinion, but with a manual you can go at your own pace, easily re-read passages and study pictures, all of which are not possible with a tape. On the other hand, the tapes are supposed to complement the manuals, and as a result they do a good job.

ENTER THE FAMILY SOLUTION

The A3010 Family Solution, which is the cheapest way of buying an A3010, costs only £499 inc. VAT - compare that with an 8-bit Master 128 which retails at £399 ex. VAT and doesn't have a disc drive, a mouse or anything like RISC OS. The package consists of the A3010 computer, Minerva Software's EasiWord word processor and a game, which at the moment is Krisalis' Quest for Gold, the Olympic games simulation, but it is understood that a different game will be included nearer Christmas.

EASIWORD

EasiWord is simply a word processor, nothing more, nothing less. It has replaced the dated 1st Word Plus as the word processor supplied with Acorn machines, and I must say I'm very relieved. 1st Word Plus is a sophisticated product with features such as mail merge, but it's not renowned for its usability and flexibility. That's why I'm pleased: EasiWord is not as feature packed as 1st Word Plus, but it contains all the necessary functions for the type of word processing purchasers of the A3010 will want to do, and it is completely RISC OS compliant (i.e. it conforms to Acorn's standards). This last point is more important than you may think, as it meant that I was able to become competent in using EasiWord in about ten minutes (seriously!); such is the power of the RISC OS Desktop, when you are used to using it.

So what features does EasiWord boast? It is fully menu driven, with keyboard shortcuts for almost every function to allow competent users to speed up their editing. The document is displayed using the normal system font, and all the effects possible (bold, italic, underline, superscript and subscript) are shown on the screen just as they will be printed. Other effects that can be applied are upper case, lower case, swap case and title (which makes the first letter of each word a capital, and the rest lower case).

Editing and entering text is simple and intuitive. Text rulers can easily be edited and created, using one of the best methods I have ever seen - you simply double-click with the mouse on the ruler to insert a tab, and double-clicking with the Adjust button on the mouse cycles this tab through the four types (left, right, centred and decimal). Right and left margins can be altered simply by dragging them, and the strip down the left-hand side of the window can be made to show the scope of each ruler in the text in a graphical form.

Other features of EasiWord are: a 50,000 word spelling checker (including an option to check as you type and user dictionaries), the ability to save files in 1st Word Plus or ASCII text format, very easy page layout editing, headers and footers, comprehensive find and replace, a fast Goto function, multiple views on documents, and a host of useful little options. The manual is excellent, and is very easy to follow, even for complete beginners.

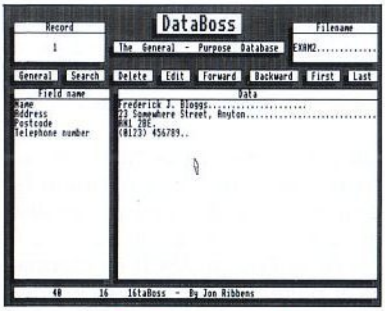
CONCLUSION

As you can see from my enthusiasm, I think the A3010 Family Solution is excellent. Now it's up to Acorn to sell the machine, and sell it they must. It's available in the high street stores and Acorn dealers, but that's not the end of the story; in high street stores it's the blind selling to the blind, and there's always the worry that sales staff will feel more confident selling machines like PCs with which they are more familiar, but which are much less suitable for family computing. The A3010 deserves to win hands down, but the question is, will it? Only time will tell.

DataBoss

Jonathan Ribbens gives you a database with added WIMP.

DataBoss is a standard text database of the sort that is very useful for addresses and fairly simple data that needs little manipulation. It is also an example of just how sophisticated the interface on the Master can be with very little programming. DataBoss will only run on machines with shadow screen modes available.



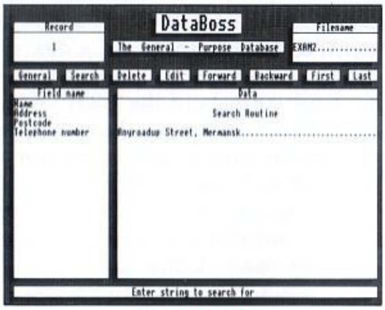
```
Record        DataBoss                              Filename
  1           The General - Purpose Database        EXAM2............
General  Search  Delete  Edit  Forward  Backward  First  Last
Field name          Data
Name                Frederick J. Bloggs......................
Address             23 Somewhere Street, Anyton..............................
Postcode            AA1 2BE.
Telephone number    (0123) 456789..
    40      16     16taBoss  -  By Jon Ribbens
```

Reading a record

DataBoss uses a pointer, in this case driven by the cursor keys, to select items from various menus; there's nothing to stop those of you with a mouse of any sort replacing the input routine with mouse commands.

HIT THE KEYBOARD

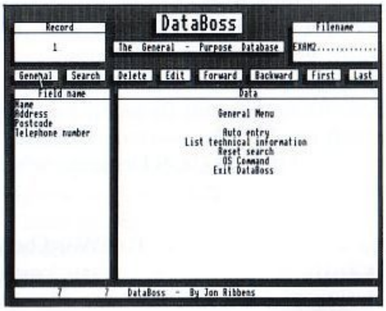
The first listing, Pointb, should be typed in, saved and run. This will produce the machine code for the pointer that appears on the screen in the main program. The code is saved on the disc as Pointer and runs at &900. After it has been run you can delete Pointb from the disc but you should save a backup copy somewhere.



```
Record        DataBoss                              Filename
  1           The General - Purpose Database        EXAM2............
General  Search  Delete  Edit  Forward  Backward  First  Last
Field name          Data
Name                          Search Routine
Address
Postcode
Telephone number    Anyroadup Street, Mermansk..............................
              Enter string to search for
```

Searching

The second program, Datab1, should now be entered and saved under that name. This sets up the screen for the database, loads the machine code and then chains the main program. Datab2 should now be entered. You'll note that the code here is very compact using multi-statement lines; it should be typed exactly as it is to leave sufficient memory for the program to run, so take do care. It must also be saved as Datab2.



```
Record        DataBoss                              Filename
  1           The General - Purpose Database        EXAM2............
General  Search  Delete  Edit  Forward  Backward  First  Last
Field name          Data
Name                          General Menu
Address
Postcode                       Auto entry
Telephone number        List technical information
                              Reset search
                               OS Command
                              Exit DataBoss
     7       7     DataBoss  -  By Jon Ribbens
```

The general menu

HOW TO USE IT

Once you have the program running, a long thin window at the bottom of the screen is used for message prompts; the first message will be asking for a file name. If you have the magazine disc for this issue a file Example is included. The file name appears at the top right of the screen and if it isn't found on the disc you will be asked if you wish that file to be created. On pressing 'Y' you can enter each field name followed by Return, then its maximum length, again followed by Return. When all fields are entered just press Return and the first twenty records will be created on the disc.

Field names are now displayed in the left hand window with their maximum lengths represented by dots to the right. The pointer now appears and is moved about the screen with the cursor keys. Holding down Shift will speed up movement and an item is selected by pressing f0. Forward, Backward, First and Last move you through the records in a file. Edit lets you enter or change the contents of a record and will automatically extend the file on disc if you edit the last record.

Search is a simple 'contains string' search of the entire file. You should note that after a search only those records that were 'found' will be processed or viewed, so further searches will have the effect of an 'AND' search. To re-instate the entire file select the General option then Reset Search. The General option also gives access to OS commands and technical information about the current file, as well as offering Auto Entry which is useful for entering large amounts of data.

There is room for extra code, though it should be kept compact. The most obvious addition would be a hard copy routine that could run from the 'General' menu. I hope you find this program both interesting and useful.

Datab1

```
   10 REM Program Datab1
   20 REM Version B 1.0
   30 REM Author  J.Ribbens
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
  100 IFPAGE>&1400:PAGE=&1400:CHAIN"Data
B1"
  110 MODE128:CLOSE#0:?&8F=(HIMEM=&8000)
  120 VDU23;10,32;0;0;0;
  130 PROCscreen
  140 *Pointer
  150 CHAIN"DataB2"
  160 :
 1000 DEFPROCscreen:?&D0=?&D0 OR2
 1010 VDU23,224,&55AA;&55AA;&55AA;&55AA;
 1020 FORI%=1TO32
 1030 PRINTSTRING$(80,CHR$224);:NEXTI%
 1040 ?&D0=?&D0 AND253:COLOUR129:COLOUR0
 1050 PROCt("DataBoss",1,TRUE)
 1060 PROCt("The  General  -  Purpose  D
atabase",4,FALSE)
 1070 PROCs("General",2,7)
 1080 PROCs("Search",13,7)
 1090 PROCs("Delete",23,7)
 1100 PROCs("Edit",33,7)
 1110 PROCs("Forward",41,7)
 1120 PROCs("Backward",52,7)
 1130 PROCs("First",64,7)
 1140 PROCs("Last",73,7)
 1150 PROCw(2,28,21,9)
 1160 PROCu("Field name",2,9,20)
 1170 PROCw(24,28,78,9)
 1180 PROCu("Data",24,9,55)
 1190 PROCw(2,5,20,2)
 1200 PROCu("Record",2,2,19)
 1210 PROCw(61,5,78,2)
 1220 PROCu("Filename",61,2,18)
 1230 PROCw(2,30,78,30)
```

```
1240 ENDPROC
1250 :
1260 DEFPROCw(x1%,y1%,x2%,y2%)
1270 X1%=x1%*16:X2%=x2%*16+16
1280 Y1%=1024-y1%*32-32
1290 Y2%=1024-y2%*32
1300 GCOL0,0
1310 PROCb(X1%-2,Y1%-4,X2%+2,Y2%+4)
1320 PROCb(X1%-16,Y1%-16,X2%-16,Y2%-16)
1330 GCOL0,1
1340 PROCb(X1%,Y1%,X2%,Y2%)
1350 VDU28,x1%,y1%,x2%,y2%:ENDPROC
1360 :
1370 DEFPROCb(x1%,y1%,x2%,y2%)
1380 MOVEx1%,y1%:MOVEx2%,y1%
1390 PLOT85,x1%,y2%:PLOT85,x2%,y2%
1400 ENDPROC
1410 :
1420 DEFPROCt(A$,Y%,D%):LOCALL%,X%,X2%,
Y2%
1430 L%=LENA$:IFD%:L%=L%*2
1440 X%=40-L%/2
1450 X2%=X%+L%+1
1460 Y2%=Y%:IFD%:Y2%=Y2%+1
1470 PROCw(X%,Y2%,X2%,Y%)
1480 VDU32:IFD%:PROCd(A$):ENDPROC
1490 PRINTA$;:ENDPROC
1500 :
1510 DEFPROCd(A$):LOCALA%,I%,X%,Y%
1520 FORI%=1TOLENA$
1530 A%=10:X%=0:Y%=9
1540 ?&900=ASCMID$(A$,I%):CALL&FFF1
1550 PROCcalc:VDU224,225,10,8,8,226,227
,11:NEXTI%:ENDPROC
1560 :
1570 DEFPROCcalc
1580 PROCcalc2(224,&901)
1590 PROCcalc2(226,&905):ENDPROC
1600 :
1610 DEFPROCcalc2(V%,P%):LOCALI%
1620 VDU23,V%
1630 FORI%=0TO3
1640 VDUFNcalc3(?(I%+P%)),FNcalc3(?(I%+
P%)):NEXTI%
1650 VDU23,V%+1
1660 FORI%=0TO3
1670 VDUFNcalc4(?(I%+P%)),FNcalc4(?(I%+
P%)):NEXTI%
1680 ENDPROC
1690 :
1700 DEFFNcalc3(V%)
1710 =((V%AND128)+(V%AND128)/2+(V%AND64
)/2+(V%AND64)/4+(V%AND32)/4+(V%AND32)/8+
(V%AND16)/8+(V%AND16)/16)
1720 :
1730 DEFFNcalc4(V%)
1740 =((V%AND8)*16+(V%AND8)*8+(V%AND4)*
8+(V%AND4)*4+(V%AND2)*4+(V%AND2)*2+(V%AN
D1)*2+(V%AND1))
1750 :
1760 DEFPROCs(A$,X%,Y%)
1770 PROCw(X%,Y%,X%+LENA$+1,Y%)
1780 VDU32:PRINTA$;CHR$26:ENDPROC
1790 :
1800 DEFPROCu(A$,X%,Y%,W%)
1810 x%=X%*16:y%=1024-Y%*32-32
1820 x2%=((W%/2)-LENA$/2)*16+x%
1830 VDU5:GCOL0,0:MOVEx2%,y%+32:PRINTA$
1840 MOVEx%,y%:DRAWx%+W%*16,y%
1850 VDU4:ENDPROC
```

Datab2

```
   10 REM Program Datab2
   20 REM Version B 1.0
   30 REM Author  J.Ribbens
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
  100 FORI%=0TO4:PROCa(I%):CLS:NEXTI%:PR
OCf:ONERRORPROCa(4):REPORT:PROCj
  110 PROCe:ONERRORCALL&903:PROCb(0):PRO
Ca(3):CLS:PROCa(4):CLS:REPORT:PROCa(3):C
LS:PROCj
  120 PROCg
  130 :
 1000 DEF PROCb(I%):IFI%=0:VDU23;10,32;0
;0;0;:ENDPROC ELSEVDU23;10,64;0;0;0;:END
PROC
```

```
 1010 DEF PROCi:IFINKEY-26:IFX%>0:CALL&9
03:X%=X%-1:CALL&900
 1020 IFINKEY-122:IFX%<79:CALL&903:X%=X%
+1:CALL&900
 1030 IFINKEY-58:IFY%>0:CALL&903:Y%=Y%-1
:CALL&900
 1040 IFINKEY-42:IFY%<30:CALL&903:Y%=Y%+
1:CALL&900
 1050 IFINKEY-2:PROCa(4):VDU30:PRINTX%,Y
%;
 1060 ENDPROC
 1070 DEF PROCh:CALL&900:REPEAT:PROCi:TI
ME=0:REPEATUNTILTIME>3ORINKEY-1:UNTILINK
EY-33:CALL&903:OSCLI"FX15":ENDPROC
 1080 DEF PROCf:DIMf$(19),l%(19):a%=&A00
:R%=1:X%=40:Y%=16:FORI%=a%TOa%+508STEP4:
!I%=-1:NEXTI%:ENDPROC
 1090 DEFFNd(A$,I%)=A$+STRING$(I%-LENA$,
".")
 1100 DEF PROCk(R%):LOCALI%:PTR#H%=D%+(R
%-1)*L%:FORI%=1TOF%:f$(I%)=FNe(l%(I%)):N
EXTI%:ENDPROC
 1110 DEF PROCm(R%):LOCALI%:PTR#H%=D%+(R
%-1)*L%:FORI%=1TOF%:PROCo(f$(I%),l%(I%))
:NEXTI%:ENDPROC
 1120 DEF PROCd:PROCk(R%):PROCa(3):FORI%
=1TOF%:PRINTTAB(0,I%-1);FNd(f$(I%),l%(I%
));:NEXTI%:PROCa(0):PRINT'SPC(8);R%;SPC(
3);:ENDPROC
 1130 DEF PROCa(I%):VDU26:IFI%=0:VDU28,2
,5,20,3ELSEIFI%=1:VDU28,61,5,78,3ELSEIFI
%=2:VDU28,2,28,21,10ELSEIFI%=3:VDU28,24,
28,78,10ELSEIFI%=4:VDU28,2,30,78,30
 1140 ENDPROC
 1150 DEF PROCl(A$,W%):PROCa(4):CLS:PRIN
TTAB(38-LENA$/2,0);A$;:PROCa(W%):ENDPROC
 1160 DEF PROCe:REPEAT:PROCl("Enter file
name of database",1):PRINT:f$=FNc(18):H%
=OPENINf$:IFH%=0:IFFNa:UNTIL0
 1170 IFBGET#H%<>74ORBGET#H%<>82:PROCl("
Sorry - that's not a DataBoss file",0):C
LOSE#H%:PROCj:UNTIL0
 1180 UNTILH%:PROCl("Loading database ..
.",2):CLOSE#H%:H%=OPENUPf$:PTR#H%=2:L%=B
GET#H%:F%=BGET#H%:FORI%=1TOF%:INPUT#H%,A
$,l%(I%):PRINTA$;:IFI%<F%ANDLENA$<20:PRI
NT
 1190 NEXTI%:D%=BGET#H%+256*BGET#H%:PTR#
H%=D%-4:T%=BGET#H%+BGET#H%*256:M%=BGET#H
%+BGET#H%*256:ENDPROC
 1200 DEFFNa:IFFNy("Can't find that file
 - Create (Y/N) ?")=0:=1
 1210 PROCl("Enter fields and lengths -
Enter nothing for field name to finish",
3):F%=0:E%=0:REPEAT:F%=F%+1:VDU31,0,F%-1
:f$(F%)=FNc(20):IFf$(F%)="":IFF%>2:F%=F%
-1:E%=TRUE
 1220 IFNOTE%:IFf$(F%)="":IFF%<3:PROCl("
You need at least two fields!",0):F%=F%-
1:PROCj:PROCl("Enter fields and lengths
- Enter nothing for field name to finish
",3)
 1230 IFNOTE%:IFf$(F%)<>"":PRINT"  Lengt
h: ";:l%(F%)=VALFNc(2):PRINT
 1240 IFNOTE%:IFf$(F%)<>"":IFl%(F%)<1ORl
%(F%)>55:VDU8:F%=F%-1:UNTIL0
 1250 IFNOTE%:IFf$(F%)<>"":IFF%=19:PROCl
("Can't have more than nineteen fields -
 automatically finishing",3):PROCj:E%=TR
UE
 1260 UNTILE%:PROCl("Creating database .
..",3):L%=2:FORI%=1TOF%:L%=L%+l%(I%):NEX
TI%:H%=OPENOUT(f$):BPUT#H%,74:BPUT#H%,82
:BPUT#H%,L%:BPUT#H%,F%:FORI%=1TOF%:PRINT
#H%,f$(I%),l%(I%):NEXTI%:D%=PTR#H%+6:BPU
T#H%,D% MOD 256:BPUT#H%,D% DIV 256
 1270 FORI%=1TOF%:f$(I%)="":NEXTI%:FORI%
=1TO20:PROCm(I%):NEXTI%:PTR#H%=D%-4:BPUT
#H%,20:FORI%=0TO2:BPUT#H%,0:NEXT:CLOSE#H
%:H%=OPENUPf$:CLS:=0
 1280 DEFFNy(A$):LOCALK%:PROCl(A$,0):REP
EAT:K%=FNb AND&DF:UNTILK%=78ORK%=89:PROC
l("",0):=(K%=89)
 1290 DEF PROCj:TIME=0:REPEATUNTILTIME>2
00:ENDPROC
 1300 DEFFNb:OSCLI"FX15":=GET
 1310 DEFFNc(M%):LOCALK%,A$:A$="":PRINTS
```

continued on page 56

StyleWise

Can the computer improve your English? Jonathan Temple shows that we have the technology.

The notion of a 'good' literary style will always be elusive. Swift's "The proper words in the proper places" is the most popular definition; the remark and its fame reflect the difficulty of saying anything less general. Orwell wrote that good prose is like a window pane: the barrier between the reader and understanding should be as transparent as possible.

The program presented here aims not to judge your prose, but to point out where things might be going wrong. It is modelled on professional 'style checkers' available for the IBM PC. Like them, *StyleWise* is designed for non-fiction writing, and is especially suitable for technical articles, PhD theses and so on. The program is best seen as a kind of early warning system, to be used on the first draft of an article. It indicates where there may be room for improvement. Its role and results should never be taken too seriously. Even so, it should provide some interest and entertainment, in the cause of better prose.

StyleWise first presents a catalogue of the current disc, and asks you to enter a filename. Entering 'C' will allow you to catalogue another disc. The program is designed for text files in either ASCII (spooled) or Wordwise Plus format; two are included on this month's disc called *Good* and *Bad*. Although it should work with the files of other word processors, embedded commands may alter the results. Please note that the '@' symbol is used as an end of text marker by StyleWise. If '@' appears in the text, StyleWise will end its analysis at that point. Also note that analysis can be rather slow. To remedy this I have written a machine code version of StyleWise, which is included on this month's magazine disc as *StyleMC*.

Once you have seen the results, you can choose 'A' to analyse another file. Results for up to twenty files are stored, and can be recalled by pressing 'R'. This option also allows you to analyse all the files combined: useful for a thesis saved in separate chapters, or a long document split into several blocks. Pressing 'C' clears all the previous results.

The program works by measuring some of the characteristics of your text, and then showing how far they are from a desirable norm. Of these, probably the most important is the 'Complexity' rating; it can be used as a general guide to the clarity and readability of a piece. If your text emerges as complex this may be due to the use of long words or long sentences, probably both. Another cause may be complicated sentences with many clauses, if these are denoted by parentheses or commas.

The other ratings are self-explanatory. Two indicate the average word length and the proportion of long words (those with more than eight letters). These will be high if your prose contains a lot of jargon or technical terms. For certain kinds of subjects it may prove difficult to improve this rating, at least without worsening the piece in other ways.

This is less true of the next measure, the average sentence length. Short sentences are usually recommended for clarity and simplicity. Obviously it will always be possible for a piece to include many long and complicated sentences, and yet still be well-written. However, it is that much harder to keep such prose clear and readable; the demands made on your literary skill are that much greater.

In general it is probably better to err on the low rather than the high side of each norm. The one exception to this is sentence variety. Using sentences of different lengths is a widely recommended way of adding interest to potentially stodgy writing. To quote a Teach Yourself book, "a judicious alternation of short and long sentences gives variety and a flowing style".

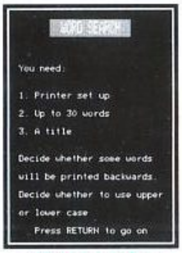
Whether or not your prose emerges from StyleWise with its reputation intact, you may be wondering how the program's settings were decided upon. What constitutes a desirable sentence length? In making such judgements, I used several examples of good and bad prose from "The Complete Plain Words" by Sir Ernest Gowers. Even so, there remains an element of personal prejudice, and this is one reason for not taking the results too seriously. A second is that prose could be 'passed' by StyleWise despite being extremely bad. If you are interested in making it better, "Gowers" is probably a good place to start.

```
   10 REM Program StyleWise
   20 REM Version B 1.0
   30 REM Author  J.R.W.Temple
   40 REM Beebug  December 1992
   50 REM Program subject to copyright
   60 :
  100 MODE 6
  110 PROCinit
  120 PROCreset
  130 PROCanalyse
  140 PROCdisplay(F)
  150 opt=FNoptions
  160 IF opt=1 AND F=20 THEN PRINT"  Too
 many files to add another";:GOTO 150
  170 IF opt=1 THEN 130
  180 IF opt=2 THEN PROCallres:GOTO 150
  190 IF opt=3 THEN 120
  200 CLS:PRINT'
  210 END
  220 :
 1000 DEF FNoptions
 1010 PRINT TAB(2,23)"(A)nother (R)esult
s (C)lear (Q)uit"
 1020 REPEAT
 1030 G$=CHR$(GET AND 223)
 1040 opt=INSTR("ARCQ",G$)
 1050 UNTIL opt>0
 1060 =opt
 1070 :
 1080 DEF PROCreset
 1090 W(0)=0:S(0)=0:LW(0)=0
 1100 P(0)=0:V(0)=0:TL(0)=0
 1110 F=0:F$(0)="all files combined"
 1120 ENDPROC
 1130 :
 1140 DEF PROCallres
 1150 CLS:@%=5
 1160 PRINT TAB(4,10-F DIV2);
 1170 PRINT "Filename"SPC(18)"Words"'U$
 1180 FOR N%=1 TO F
 1190 PRINT N%;"  "F$(N%) TAB(29),W(N%)
 1200 NEXT
 1210 PRINT 0;"  "F$(0) TAB(29),W(0)
 1220 PRINT U$'"    Enter a number (0-";
F;
 1230 INPUT "): "K%
 1240 IF K%<0 OR K%>F THEN PRINT'"    No
t valid - Press a key";:G=GET:GOTO 1150
 1250 @%=&90A
 1260 PROCdisplay(K%)
 1270 ENDPROC
 1280 :
 1290 DEF PROCdisplay(F)
 1300 W=W(F):S=S(F):LW=LW(F)
 1310 P=P(F):V=V(F):TL=TL(F)
 1320 AWL=(TL/W)
 1330 CM=(AWL-3)*AWL*0.8+(W/S)/3+(P/W)*(
P/W)*400+20*(LW/W)-5
 1340 WM=(AWL*AWL)/2+1
 1350 LM=(80*LW/W)+4
 1360 SM=(W/S)/1.5-1
 1370 VR=0.05:IF S>1 THEN VR=V/(S-1)
 1380 VM=VR*60-3
 1390 CLS
 1400 T$="Analysis of "+F$(F)
 1410 PRINT''TAB((40-LEN T$)/2) T$
 1420 PRINT'U$
 1430 PRINT TAB(0,6)"Low   --Best--   Hi
gh"
 1440 PRINT'B$"  Complexity"
 1450 PRINT'B$"  Word length"
 1460 PRINT'B$"  Long words"
 1470 PRINT'B$"  Sentence length"
 1480 PRINT'B$"  Sentence variety"
 1490 PROCmark(CM,8):PROCmark(WM,10)
 1500 PROCmark(LM,12):PROCmark(SM,14)
 1510 PROCmark(VM,16)
 1520 PRINT''';W;" words  ";S;" sentence
s (av.";INT(W/S+0.5);" words)"
 1530 PRINT TAB(0,21) U$
 1540 ENDPROC
 1550 :
 1560 DEF PROCmark(X,Y)
 1570 X=INT(X+0.5):IF X<0 THEN X=0
 1580 IF X>20 THEN X=20
 1590 VDU 31,X,Y,225
 1600 ENDPROC
 1610 :
 1620 DEF PROCanalyse
 1630 LL=0:L=0:W=0:LW=0
 1640 S=0:P=0:V=0:TL=0
 1650 CLS:PRINT'TAB(15)"StyleWise"'U$
 1660 VDU 28,0,20,39,3,12
 1670 *CAT
 1680 VDU 26:PRINT TAB(0,21) U$
 1690 VDU 28,0,24,39,22,12
 1700 INPUT'"Enter filename: "F$
 1710 IF F$="C" THEN 1660
 1720 C=OPENIN(F$)
 1730 IF C=0 THEN CLS:PRINT'F$" not foun
d - Press a key":G=GET:CLS:GOTO 1700
 1740 CLS:PRINT"Analysing "F$"..."
 1750 VDU 26:len=EXT #C
 1760 bnum=1+len DIV R%
 1770 PROCcheck
 1780 CLOSE #C
 1790 F=F+1:F$(F)=F$
 1800 W(F)=W:S(F)=S:LW(F)=LW
 1810 P(F)=P:V(F)=V:TL(F)=TL
 1820 W(0)=W(0)+W:S(0)=S(0)+S
 1830 LW(0)=LW(0)+LW:P(0)=P(0)+P
 1840 V(0)=V(0)+V:TL(0)=TL(0)+TL
 1850 ENDPROC
 1860 :
 1870 DEF PROCcheck
 1880 PRINT TAB(0,24)"Word count: 0";
 1890 block!9=0:PROCnextblock
 1900 REPEAT
 1910 REPEAT:REPEAT
 1920 PROCchar
 1930 UNTIL C%=64 OR INSTR(P$,C$)=0
 1940 UNTIL C%>31 AND C%<127
 1950 IF C%=64 THEN 2040
 1960 W=W+1:L=L+1:W$=""
 1970 PRINT TAB(12,24);W;
 1980 REPEAT
 1990 W$=W$+C$:PROCchar
 2000 UNTIL C%<31 OR C%>126 OR INSTR(P$,
C$)>0
 2010 IF LEN(W$)>8 THEN LW=LW+1
 2020 TL=TL+LEN(W$)
 2030 IF C%=46 OR C%=58 OR C%=59 THEN PR
OCsentence
 2040 UNTIL C%=64
 2050 ENDPROC
 2060 :
 2070 DEF PROCsentence
 2080 IF W$="Mr" OR W$="Mrs" OR W$="Ms"
OR W$="eg" OR W$="ie" OR W$="NB" THEN EN
DPROC
 2090 IF L>30 THEN L=30+(L DIV2)
 2100 R=LL/L:IF R>1 THEN R=L/LL
 2110 IF S>0 THEN V=V+(1-R)*(1-R)
 2120 LL=L:S=S+1:L=0
 2130 ENDPROC
 2140 :
 2150 DEF PROCchar
 2160 PROCgetc
 2170 IF C%=2 THEN REPEAT:PROCgetc:UNTIL
 C%=7 OR C%=13 OR C%=64 OR C%>126:IF C%<
>64 THEN PROCgetc
 2180 IF C%=40 THEN P=P+4
 2190 IF C%=44 OR C%=58 OR C%=59 THEN P=
P+1
 2200 C$=CHR$(C%)
 2210 ENDPROC
 2220 :
 2230 DEF PROCgetc
 2240 M%=M%+1
 2250 IF M%=E% THEN PROCnextblock:M%=M%+
1
 2260 C%=?M%
 2270 ENDPROC
 2280 :
 2290 DEF PROCnextblock
 2300 ?block=C
 2310 block!1=store:block!5=R%
 2320 X%=block MOD 256
 2330 Y%=block DIV 256
 2340 A%=3:CALL &FFD1
 2350 bnum=bnum-1
 2360 IF bnum=0 THEN store?(len MOD R%)=
64
 2370 M%=store-1
 2380 ENDPROC
 2390 :
 2400 DEF PROCinit
 2410 VDU 23,224,0,0,255,0,0,255,0,0
 2420 VDU 23,225,60,60,255,60,60,255,60,
60
 2430 VDU 23,226,0,0,0,255,0,0,0,0
 2440 DIM F$(20),W(20),S(20),LW(20)
 2450 DIM TL(20),P(20),V(20)
 2460 P$=" ,.:;-"+CHR$(13)+CHR$(34)+"/()
?![]@"
 2470 VDU 23;10,32;0;0;0;
 2480 U$=STRING$(39,CHR$226)
 2490 B$=STRING$(21,CHR$ 224)
 2500 free=(HIMEM-LOMEM)-4000
 2510 R%=256*(free DIV 256)
 2520 DIM block 16,store R%+50
 2530 E%=store+R%
 2540 ENDPROC
```

Word-search Puzzle Generator

Barry Thorpe speeds up the creation of these brain-teasers to keep you amused over the festive season.

Wordsq, provided in the listing below, is a fairly straightforward example of the computer doing a useful job quickly. The program generates a word-search grid containing up to thirty words you have selected. The output is to any Epson compatible printer.

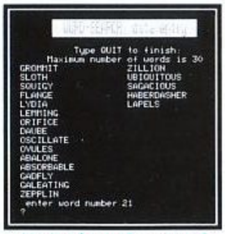


```
WORD SEARCH

You need:

1. Printer set up
2. Up to 30 words
3. A title

Decide whether some words
will be printed backwards.
Decide whether to use upper
or lower case
   Press RETURN to go on
```

The opening screen

INPUT

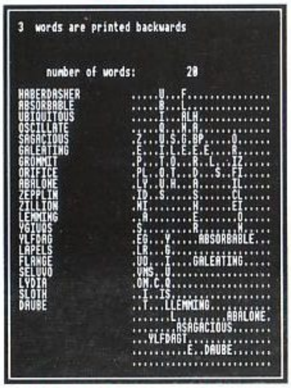
On running the program you will first be asked if you want upper or lower case letters on the grid. You are then asked if you want some of the words to appear reversed; if you answer 'yes' here, up to a third of the words will be printed backwards.



```
        Type QUIT to finish:
   Maximum number of words is 30
GROMMIT                 ZILLION
SLOTH                   UBIQUITOUS
SQUIGY                  SAGACIOUS
FLANGE                  HABERDASHER
LYDIA                   LAPELS
LEMMING
ORIFICE
DAUBE
OSCILLATE
OVULES
ABALONE
ABSORBABLE
GADFLY
GALEATING
ZEPPLIN
 enter word number 21
?
```

Entering the word list

This is followed by the main input screen. Here you enter first a title for the puzzle, then up to thirty words to be in the puzzle. If you want fewer than thirty just type *quit* after the final word.

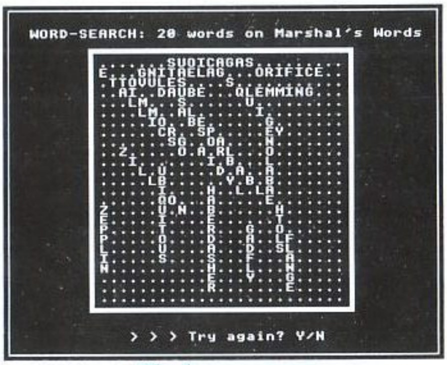


```
3  words are printed backwards

     number of words:        20

HABERDASHER      .....U...F..............
ABSORBABLE       .....B...L..............
UBIQUITOUS       .....I...ALH............
OSCILLATE        .....Q...N.A............
SAGACIOUS        .Z...U.S.G.BP.....O.....
GALEATING        .E...I.L.E.E.E....R.....
GROMMIT          .P...T.O...R..L...IZ....
ORIFICE          .PL..O.T...D...S..FI....
ABALONE          .LY..U.H...A......IL....
ZEPPLIN          .ID..S.....S......CL....
ZILLION          .NI........H......EI....
LEMMING          ..A........E......O.....
YGIUQS           .S.........R......N.....
YLFDAG           .EG...Y.....ABSORBABLE..
LAPELS           .LR...G.................
FLANGE           .UD...I....GALEATING....
SELUVO           .VNS..U.................
LYDIA            .ON.C.Q.................
SLOTH            ..I..IS.................
DAUBE            ..T...LLEMMING..........
                 .......L.........ABALONE.
                 ........ASAGACIOUS......
                 ...YLFDAGT..............
                 ..........E..DAUBE......
                 ........................
```

The skeleton printout

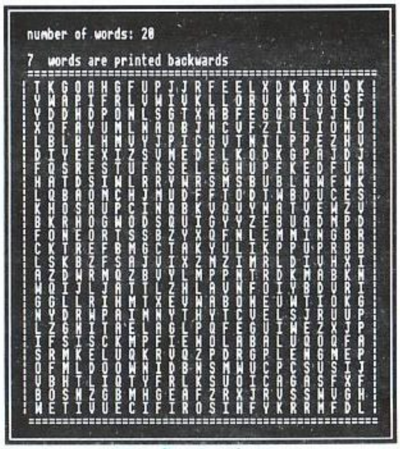
OUTPUT

Once all the words are entered you will see them inserted into the grid, starting with the longest, but the extra letters are not added at this stage. If you don't like the way the words are arranged you can ask for the process to be run again. Once you're happy with the result you can send a skeleton printout to the printer (the answer!) with a reminder of the word list. Finally you send the completed puzzle to the printer with the chance to amaze you friends and confound your enemies - *good luck*.



```
WORD-SEARCH: 20 words on Marshal's Words

......SUOICAGAS.........
E...GNITAELAG...ORIFICE..
.TTOVULES....S..........
..AI..DAUBE...QLEMMING...
...LM...S......U........
....LM..AL......I.......
.....IO..BE......G......
......CR..SP.....EY.....
.......SG..OA.....N.....
..Z.....O.A.RL....O.....
...I.......I.B...L......
....L.U.....D.A..A......
.....LB......Y.B.B......
......I....H..L.LA......
......QO...A.....E......
Z.....U.N..B.......H....
E.....I....E.......T....
P.....T....R...G...O....
P.....O....D...A..LF....
L.....U....A...D..SL....
I.....S....S...F...A....
N..........H....L...N...
...........E....Y...G...
...........R........E...
........................

> > > Try again? Y/N
```

The key on screen



```
number of words: 20

7  words are printed backwards
===================================================
! T K G O A H G F U P J J R F E E L X D K R X U D K !
! Y W A P I F R L V W I V K L L O R V K M J O G S F !
! Y D D H D P O N L S G T X A B F E G Q G L Y J L V !
! X Q F A Y U M L N A D B J N C V F Z I L L I O N O !
! L B L B L H N V Y J P I C G V T N I L P P E Z H V !
! D I Y E E X I Z S V M E D E L K O D K G P A J D J !
! F Q S R E S T U F R S E L E G H U P F K E D F U A !
! H A T D S I N L R R Y W R S M S B U B L N W F W K !
! L Q B A O M C H J M U D F F T O D T W B D U C E S !
! K H O S A U P C I N Q B K V Q V Y N A T V E W Z V !
! B K R H O G W O D S Q U I G Y Y Z E B U A D M P D !
! F O N E O R T S S B Z Y X Q F N L E M M I N G B R !
! C K T R E F B M G C T A K Y U L I K P P U P R B B !
! L S K B Z F S A J V I X Z M Z I M R L D I V H X I !
! A Z D W R M Q Z B V Y L M P P N T R D K M A B K N !
! W Q I J L J A T T L Z H L A V N F O I Y B D V R I !
! G G L L R I H M T X E V W A B O H E U W T I O K G !
! N Y D R W P A I M N Y T H Y T C V E L S J R U U P !
! L Z G M I T A E L A G L P Q F E G U I W E Z X J P !
! I F S I S C K M P F I E N O L A B A L U Q O Q F A !
! S R M K E L U Q K R V Q Z P D R G P L E N G M E P !
! O F M L D O O W N I D B N S M W U C P C S U S I J !
! U B H T L I Q T Y F R L K S U O I C A G A S F X F !
! B O S N Z G B M H G E A F Z R X J R V S S W V G H !
! W E T I V U E C I F I R O S I H F Y K R R M F D L !
===================================================
```

The final printout

```
   10 REM Program Wordsq
   20 REM Version B 1.0
   30 REM Author  B. Thorpe
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
  100 MODE7:PROCinit
  110 PROCdouble("WORD SEARCH",129,134,0
)
  120 PRINTTAB(5,5) CHR$134 "You need:"
  130 PRINTTAB(5,8) CHR$131 "1. Printer
set up"
  140 PRINTTAB(5,10) CHR$131 "2. Up to "
;maxwords%;" words"
  150 PRINTTAB(5,12) CHR$131 "3. A title
"
  160 PRINTTAB(5,15) CHR$134 "Decide whe
ther some words"''TAB(5) CHR$134 "will b
e printed backwards."
  170 PRINTTAB(5,19) CHR$131 "Decide whe
ther to use upper"''TAB(5) CHR$131"or lo
wer case"
  180 PRINTTAB(8,23) CHR$134 "Press RETU
RN to go on"
  190 REPEAT UNTIL GET =13
  200 PROCtestbuffer
  210 CLS:A$=FNinputandcheck("Upper case
 or Lower case (U or L)",1,0,10)
  220 lowercase=FNupper(A$)="L"
  230 A$=FNinputandcheck("Do you want so
me words backwards? Y/N",1,0,10)
  240 backwards=FNupper(A$)="Y"
  250 PROCgetdata
  260 PROCsort
  270 MODE4
  280 PROCinsertwords
  290 PROCprintchoice
  300 PROCfillsquare
  310 final=TRUE
  320 PROCprintsquare
  330 *FX229
  340 MODE7
  350 END
  360 :
 1000 DEF PROCinit
 1010 VDU23,241,255,255,255,255,0,0,0,0
 1020 VDU23,242,0,0,0,0,255,255,255,255
 1030 VDU23,243,240,240,240,240,240,240,
240,240
 1040 VDU23,244,15,15,15,15,15,15,15,15
 1050 VDU23,245,255,255,255,255,240,240,
240,240
 1060 VDU23,246,255,255,255,255,15,15,15
,15
 1070 VDU23,247,15,15,15,15,255,255,255,
255
 1080 VDU23,248,240,240,240,240,255,255,
255,255
 1090 *KEY10OLD|MRUN|M
 1100 final=FALSE:bk$="No"
 1110 size%=25:maxwords%=30
 1120 maxlen%=size%-4
 1130 DIM word$(maxwords%),square$(size%
,size%)
 1140 FOR W%=1 TO maxwords%
 1150 word$(W%)=STRING$(maxlen%,"#"):wor
d$(W%)=""
 1160 NEXT
 1170 rev$=STRING$(maxlen%,"#"):rev$=""
 1180 *FX229,1
 1190 ENDPROC
 1200 :
 1210 DEF PROCdelay
 1220 T=TIME
 1230 REPEAT
 1240 UNTIL TIME-T>200
 1250 ENDPROC
 1260 :
 1270 DEF PROCdouble(t$,bc%,fc%,ln%)
 1280 x%=(39-LENt$)DIV2-4
 1290 VDU31,x%,ln%,141,bc%,157,fc%:PRINT
t$"  "CHR$156
 1300 VDU31,x%,ln%+1,141,bc%,157,fc%:PRI
NTt$"  "CHR$156
 1310 ENDPROC
 1320 :
 1330 DEF FNupper(s$)
 1340 LOCAL A%,B$,I%
 1350 B$=""
 1360 FOR I%=1 TO LENs$
 1370 A%=ASC(MID$(s$,I%,1))
 1380 IF A%>96 AND A%<123 THEN A%=A%-32
 1390 B$=B$+CHR$(A%)
 1400 NEXT
 1410 =B$
 1420 :
 1430 DEF FNinputandcheck(prompt$,max%,x
%,y%)
 1440 REPEAT
 1450 CLS
 1460 PRINTTAB(x%,y%)CHR$131 prompt$
 1470 INPUTans$
 1480 PRINT'CHR$131 "is this ok? Y/N"
 1490 PROCyesno
 1500 ok=yes AND LENans$<=max%
 1510 IF LENans$>max% VDU7:PRINT"TOO LON
G":PROCdelay
 1520 UNTIL ok
 1530 =ans$
 1540 :
 1550 DEF PROCyesno
 1560 REPEAT
 1570 a$=FNupper(GET$)
 1580 UNTIL INSTR("YN",a$)
 1590 yes=a$="Y":no=NOT yes
 1600 ENDPROC
 1610 :
 1620 DEF FNlower(s$)
 1630 LOCAL A%,B$,I%
 1640 B$=""
 1650 FOR I%=1 TO LENs$
 1660 A%=ASC(MID$(s$,I%,1))
 1670 IF A%>64 AND A%<91 THEN A%=A%+32
 1680 B$=B$+CHR$(A%)
 1690 NEXT
 1700 =B$
 1710 :
 1720 DEF PROCcentre(t$,y%)
 1730 len%=LEN(t$):x%=(39-len%)DIV2
 1740 PRINTTAB(x%,y%)t$;
 1750 ENDPROC
 1760 :
 1770 DEF PROCtestbuffer
 1780 REPEAT
 1790 *FX21,3
 1800 probe1=ADVAL(-4):VDU2,1,0,1,0,3:pr
obe2=ADVAL(-4)
 1810 printerconnected=probe1=probe2
```

```
 1820 IF NOT printerconnected THEN CLS:P
RINT TAB(8,10)CHR$129 CHR$136"please con
nect printer"''TAB(8)"Press RETURN to go
 on":REPEAT UNTIL GET=13
 1830 UNTIL printerconnected
 1840 ENDPROC
 1850 :
 1860 DEF PROCgetdata
 1870 bcount%=0
 1880 CLS:PROCdouble("WORD-SEARCH: data
entry",131,129,0)
 1890 PROCcentre(CHR$131+"Type QUIT to f
inish:",3):PROCcentre(CHR$134+"Maximum n
umber of words is "+STR$maxwords%,4)
 1900 VDU28,0,24,39,20
 1910 title$=FNinputandcheck("What is th
e general subject matter?",35,0,10)
 1920 num%=0
 1930 REPEAT
 1940 num%=num%+1
 1950 word$(num%)=FNinputandcheck("enter
 word number "+STR$(num%),maxlen%,0,10)
 1960 IF lowercase THEN word$(num%)=FNlo
wer(word$(num%)) ELSE word$(num%)=FNuppe
r(word$(num%))
 1970 IF num%>15 THEN x%=20:y%=num%-15 E
LSE x%=0:y%=num%
 1980 VDU26:PRINTTAB(x%,4+y%)word$(num%)
:VDU28,0,24,39,20
 1990 quit=FNupper(LEFT$(word$(num%),4))
="QUIT":finished=(num%=maxwords%) OR qui
t
 2000 IF (backwards AND NOT quit) THEN P
ROCreverseword(num%)
 2010 UNTIL finished
 2020 IF quit THEN num%=num%-1
 2030 VDU26,12
 2040 IF backwards THEN bk$=STR$(bcount%
)+" " ELSE bk$="No "
 2050 ENDPROC
 2060 :
 2070 DEF PROCinsertwords
 2080 REPEAT
 2090 PROCclearsquare
 2100 CLS:PRINTTAB(10,0)"WORD SQUARE PUZ
ZLE" TAB(5,10)"..inserting word no. "
```

```
 2110 FOR W%=1 TO num%
 2120 PRINTTAB(28,10);W% TAB(5,12)SPC20
TAB(5,12) word$(W%)
 2130 wordlen%=LENword$(W%)
 2140 REPEAT
 2150 REM vert horiz or diag
 2160 X=RND(100):vert=FALSE:horiz=FALSE
 2170 rowinc%=0:colinc%=0
 2180 IF X<33 THEN vert=TRUE
 2190 IF X>66 THEN horiz=TRUE
 2200 IF X>32 AND X<67 THEN vert=TRUE:ho
riz=TRUE
 2210 IF vert THEN rowinc%=1:row%=RND(si
ze%-wordlen%):col%=RND(size%)
 2220 IF horiz THEN colinc%=1:col%=RND(s
ize%-wordlen%):row%=RND(size%)
 2230 IF horiz AND vert THEN row%=RND(si
ze%-wordlen%)
 2240 srow%=row%:scol%=col%
 2250 PROCtryfit:IF wordfits THEN PROCsl
otwordin
 2260 UNTIL wordfits
 2270 NEXT W%
 2280 PROCdisplaysquare
 2290 PRINTTAB(10,31)"> > > Try again? Y
/N";:VDU7
 2300 PROCyesno:satisfied=NOT yes
 2310 UNTIL satisfied
 2320 ENDPROC
 2330 :
 2340 DEF PROCtryfit
 2350 clash=FALSE:letter%=1
 2360 REPEAT
 2370 IF square$(row%,col%)<>"." THEN IF
 square$(row%,col%)<>MID$(word$(W%),lett
er%,1) THEN clash=TRUE
 2380 letter%=letter%+1
 2390 row%=row%+rowinc%:col%=col%+colinc
%
 2400 UNTIL clash OR letter%>wordlen%
 2410 wordfits=NOT clash
 2420 ENDPROC
 2430 :
 2440 DEF PROCslotwordin
 2450 row%=srow%:col%=scol%
 2460 FOR letter%=1 TO wordlen%
 2470 square$(row%,col%)=MID$(word$(W%),
letter%,1)
 2480 row%=row%+rowinc%:col%=col%+colinc
%
 2490 NEXT letter%
 2500 ENDPROC
 2510 :
 2520 DEF PROCdisplaysquare
 2530 CLS
 2540 left%=1+(39-size%)DIV2:right%=left
%+size%-1
 2550 top%=3:bm%=size%+top%-1
 2560 PROCborder
 2570 FOR R%=1 TO size%:FOR C%=1 TO size
%
 2580 VDU31,left%+C%-2,top%+R%-1,ASC(squ
are$(R%,C%))
 2590 NEXT:NEXT
 2600 IF final THEN PRINTTAB(5,30)bk$ "
words are printed backwards"
 2610 ENDPROC
 2620 :
 2630 DEF PROCfillsquare
 2640 IF lowercase THEN offset=96 ELSE o
ffset=64
 2650 FOR row%=1 TO size%
 2660 FOR col%=1 TO size%
 2670 IF square$(row%,col%)="." THEN squ
are$(row%,col%)=CHR$(RND(26)+offset)
 2680 NEXT:NEXT
 2690 ENDPROC
 2700 :
 2710 DEF PROCreverseword(W%)
 2720 IF RND(100)>33 THEN ENDPROC
 2730 FOR letter%=LENword$(W%) TO 1 STEP
-1
 2740 rev$=rev$+MID$(word$(W%),letter%,1
)
 2750 NEXT
 2760 REM check palindrome
 2770 IF rev$<>word$(W%) THEN bcount%=bc
ount%+1
 2780 word$(W%)=rev$
 2790 rev$=""
 2800 ENDPROC
 2810 :
```

```
 2820 DEF PROCprintskeleton
 2830 VDU2,1,27,1,64,1,27,1,87,1,1:PRINT
SPC(14)"WORD SEARCH"':len%=LENtitle$:PRI
NTSPC((39-len%)DIV2)title$''
 2840 VDU1,27,1,87,1,0:PRINTTAB(10)"numb
er of words: "num%'
 2850 FOR R%=1 TO size%
 2860 IF R%<=num% PRINTTAB(5) word$(R%);
 2870 PRINTTAB(25);
 2880 FOR C%=1 TO size%
 2890 PRINTsquare$(R%,C%);
 2900 NEXT:NEXT
 2910 PRINT''TAB(5) bk$ " words are prin
ted backwards"
 2920 VDU1,12,1,27,1,64,3
 2930 ENDPROC
 2940 :
 2950 DEF PROCprintsquare
 2960 REPEAT
 2970 VDU2,1,27,1,64,1,27,1,69,1,27,1,87
,1,1
 2980 REM  select wide type emphasised
 2990 PRINTSPC(14)"WORD SEARCH"'
 3000 len%=LENtitle$:PRINTSPC((39-len%)D
IV2)title$'
 3010 VDU1,27,1,64,1,27,1,15,1,27,1,87,1
,1:REM condensed double-width
 3020 VDU1,27,1,ASC"3",1,48:REM adjust l
ine-spacing
 3030 PRINTSPC(8) STRING$(52,"=")
 3040 FOR R%=1 TO size%
 3050 PRINT SPC(7)"| ";
 3060 FOR C%=1 TO size%
 3070 PRINT square$(R%,C%);" ";
 3080 NEXT
 3090 PRINT"|"
 3100 NEXT
 3110 PRINTSPC(8) STRING$(52,"=")
 3120 VDU1,27,1,64,1,27,1,69
 3130 PRINT''SPC(8)"number of words: ";n
um%''SPC(8)bk$ " words are printed backw
ards"
 3140 FOR I%=1 TO 21:PRINT:NEXT:VDU1,27,
1,64,3
 3150 CLS:PRINTTAB(5,10)"Another copy Y/
N?"
 3160 PROCyesno
 3170 UNTIL no
 3180 :
 3190 DEF PROCclearsquare
 3200 FOR R%=1 TO size%
 3210 FOR C%=1 TO size%
 3220 square$(R%,C%)="."
 3230 NEXT:NEXT
 3240 ENDPROC
 3250 :
 3260 DEF PROCborder
 3270 CLS:PRINT"WORD-SEARCH: ";num%;" wo
rds on "title$
 3280 PRINT TAB(left%-2,top%-1)CHR$245 S
TRING$(size%,CHR$241) CHR$246
 3290 FOR L%=top% TO bm%
 3300 VDU31,left%-2,L%,243,31,left%+size
%-1,L%,244
 3310 NEXT
 3320 PRINT TAB(left%-2,bm%+1)CHR$248 ST
RING$(size%,CHR$242) CHR$247
 3330 ENDPROC
 3340 :
 3350 DEF PROCprintchoice
 3360 PRINTTAB(0,31)SPC(38);TAB(4,31)"Sk
eleton print of this? Y/N";
 3370 PROCyesno
 3380 IF yes THEN PROCprintskeleton
 3390 CLS:PRINTTAB(5,10)"One moment, ple
ase..."
 3400 ENDPROC
 3410 :
 3420 DEF PROCsort
 3430 S%=1
 3440 REPEAT
 3450 sorted=TRUE
 3460 FOR W%=1 TO num%-S%
 3470 L1%=LEN(word$(W%)):L2%=LEN(word$(W
%+1)):IF L1%<L2% THEN X$=word$(W%):word$
(W%)=word$(W%+1):word$(W%+1)=X$:sorted=F
ALSE
 3480 NEXT
 3490 S%=S%+1
 3500 UNTIL S%=num% OR sorted
 3510 ENDPROC
```

BEEBUG Christmas Competition

Win yourself a Christmas hamper with a difference by entering our festive competition.

Not content with the excellent response to our Tenth Anniversary quiz in Vol.11 No.3, we've decided to get into the swing of Christmas by running a slightly trickier competition, with an even better prize.

Below are three *alphametic* problems, which are simply arrangements of words, the letters of which can be substituted by digits to form an arithmetically correct sum. For example, the alphametic:

```
    SEND
+   MORE
   -----
=  MONEY
```

has the solution:

```
     9576
+    1085
    -----
=   10652
```

Your task is to work out which letters stand for which digits in each of the three sums; note that each letter must represent a *different* digit. Treat each sum separately and don't assume that if 'M' stands for the digit '3' in the first sum, it will have the same value in the other two.

The first five correct entries we pick out of the hat will win a BEEBUG Christmas software hamper, containing the following goodies:

Applications I Disc
Applications II Disc
General Utilities Disc
Arcade Games Disc
Board Games Disc
ASTAAD CAD Package
10th Anniversary Disc

Don't forget to tell us which format of disc you would like to win: either 5.25" DFS or 3.5" ADFS.

```
      BBC
+   MICRO
+    USER
   ------
= BEEBUG
```

```
    MINCE
+    PIES
   ------
=   EATEN
```

```
        A
+   MERRY
+    XMAS
   ------
= TURKEY
```

Send you entries in to BEEBUG Christmas Competition, 117 Hatfield Road, St. Albans, Herts AL1 4JS. The closing date is January 11th, and do send your solutions in early to avoid the Christmas post. Good luck and Merry Christmas!

BEEBUG WORKSHOP

Tree Structures (Part 2)

by Mike Williams

With most tree structures, as in the examples of binary trees which we looked at last month, there are more null pointers than there are pointers to other nodes. We can take advantage of this otherwise wasted space to store additional information which helps to improve the process of tree traversal.

With the simple representation of a binary tree which we treated last time, there was no automatic way of returning to a higher level node in a tree without having stored previously a pointer to that node - hence the use of a stack mechanism to achieve just that. However, we can use the space occupied by the null pointers (at the ends of branches) to contain *threads* which point to a higher level node. The resulting *threaded* tree corresponding to the example given in figure 3 last month is shown here in figure 1.

Every node now has two links: some like node C have left and right pointers to subtrees, while nodes like D have two thread links, and some nodes have links of both types. There are also two special cases of threaded links, in the leftmost and rightmost positions. A way of using these will be explained later.

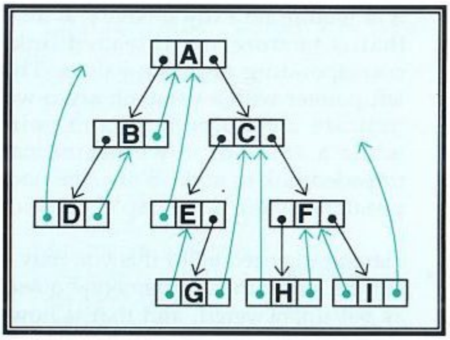


Figure 1. Threaded tree structure

In general, we cannot assume in our implementation of tree structures that the data for a higher node will be placed earlier (or higher) in the corresponding arrays. We must therefore introduce some means of distinguishing between normal left and right pointers, and threaded links.

There seem to be two possibilities for making the necessary distinction, and it may be worth noting these in principle, as the choice as to which is best may be determined by factors other than those which are relevant here. We could include additional information with each node which would indicate whether the left and right-hand pointers were genuine pointers to subtrees, or threads to higher nodes. In theory this could be accomplished in two bits, one for the left pointer and one for the right, each of which could be set to 0 or 1 depending on the use of those pointers. However, Basic does not provide us with single bit variables, and thus we would have to use a whole byte or more of memory for each node in a tree. However, if indirection operators were being used directly on an area of memory, or you were writing in machine code, then this solution would merit consideration.

I propose to use an alternative, which will require no extra memory at all, and that is to store any threaded links as corresponding negative values. Thus a left pointer with a value of, say, 5 would indicate a genuine left-hand pointer, while a value of -5 would indicate a threaded link to node '5' (i.e. the node in position 5 in our array implementation).

Having digested all of this you may well still be left with a fundamental question, as yet unanswered, and that is how the threads are determined (as shown in figure 1). It works as follows:

If the left-hand pointer of a node is a thread, the corresponding link points to the predecessor node (in *inorder*) of that node, otherwise it points to the left subtree of that node.

If the right-hand pointer of a node is a thread, the corresponding link points to the successor node (in *inorder* again) of that node, otherwise it points to the right subtree of that node.

Given these definitions we can now postulate that given a node 'k', then the successor 'k1' can be determined by the following function as follows:

```
DEF FNsuccessor(k)
LOCAL exit%,k1
exit%=FALSE
k1=RLink(k)
IF k1<0 THEN =-k1
REPEAT
IF LLink(k1)>=0 THEN k1=LLink(k1)
ELSE exit%=TRUE
UNTIL exit%
=k1
```

This function, of finding the successor to a randomly chosen node 'k', is at best cumbersome using the simple tree structure of last month, as there is no quick way of back-tracking to a node higher than 'k', unless a history is maintained of how node 'k' has been reached - that's what the stack did. Thus the routine above is more efficient, and this can be proved if direct comparisons are made.

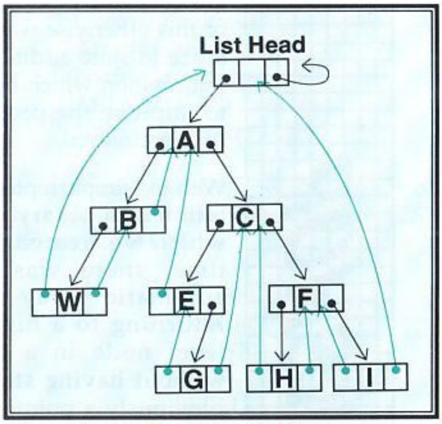


Figure 2. Modified threaded tree structure incorporating the list head

In accommodating the new algorithm there are two further changes that it is useful to make to the conventions of the previous program. Since the values of pointers can now be either positive or negative, depending upon their function, it is no longer possible to use the zero elements within the arrays as there is no way of distinguishing between -0 and +0 in BBC Basic.

In addition, it is useful to use the first node (in position 1) as a list head with the left-hand pointer of the list head pointing to the root of the tree ('A' in the example), and the right-hand pointer pointing to itself. The unattached threads from figure 1 can then be made to point to the list head. This arrangement is analogous to our use of an element of a two-way linked list as a list head in a previous Workshop (see BEEBUG Vol.11 No.5). It simplifies the processing of the list structure, and avoids the need for special cases with empty or near empty trees. If the tree is empty, then the left-hand pointer of the list head points to itself. The revised tree structure is shown in figure 2.

The program listed here implements all of these features. The example tree used last month is represented in the same way as before as a set of DATA statements, and for convenience the settings for the list head are included as a first, artificial, node. This means that there is one more set of values to read in than there are nodes in the tree. The program leaves out all the previous stack handling features, and includes the successor function detailed above, but split into two separate units for convenience. This is because we have to traverse the left-hand pointers to find the 'first' node in inorder traversal, before we can start applying *all* of the original routine to find each successor in turn.

Out of curiosity, I added an identical timing loop to both last month's program, *Tree01*, and this month's, repeating the traversal 100 times in each case. On my system this gave a figure of 22.00 seconds for *Tree01*, but 13.47 seconds for *Tree02*, substantially reducing the time taken. In this case we have also achieved that speed increase with a shorter program, and using less storage (no stack), contrary to the more usual trade off between speed and memory requirements.

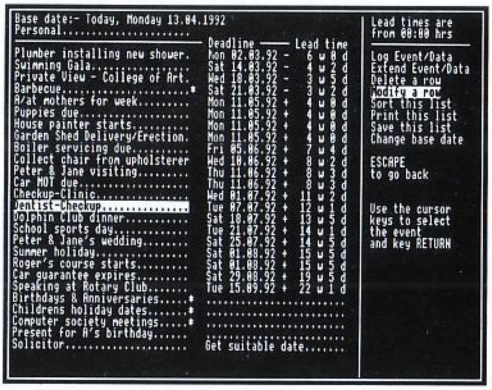
I hope you find our exploration of tree structures interesting. Next month we will look at a further way of implementing tree traversals permitted by BBC Basic, and consider some other aspects of tree structures.

```
   10 REM Program Tree02
   20 REM Version B 1.0
   30 REM Author  Mike Williams
   40 REM BEEBUG  December 1992
   50 REM program subject to copyright
   60 :
  100 ON ERROR REPORT:PRINT" at ";ERL:EN
D
  110 DIM Data$(100),LLink(100),RLink(10
0)
  120 MODE 3:PRINT'"Inorder Tree Travers
al"
  130 PROCcreate_tree
  140 PRINT
  150 PROCinorder(1)
  160 END
  170 :
 1000 DEF PROCcreate_tree
 1010 LOCAL I%,N%:READ N%
 1020 FOR I%=1 TO N%+1
 1030 READ Data$(I%),LLink(I%),RLink(I%)
 1040 NEXT I%
 1050 ENDPROC
 1060 :
 1070 DEF PROCinorder(k)
 1080 LOCAL k1
 1090 k1=k:k1=FNsuccessor2(k1)
 1100 REPEAT
 1110 PRINT Data$(k1)+" ";
 1120 k1=FNsuccessor1(k1)
 1130 UNTIL k1=k
 1140 ENDPROC
 1150 :
 1160 DEF FNsuccessor1(k)
 1170 LOCAL k1
 1180 k1=RLink(k)
 1190 IF k1<0 THEN =-k1 ELSE =FNsuccesso
r2(k1)
 1200 :
 1210 DEF FNsuccessor2(k1)
 1220 LOCAL exit%:exit%=FALSE
 1230 REPEAT
 1240 IF LLink(k1)>=0 THEN k1=LLink(k1)
ELSE exit%=TRUE
 1250 UNTIL exit%
 1260 =k1
 1270 :
 2000 DATA 9
 2010 DATA "",2,1
 2020 DATA A, 3, 4,B, 5,-2,C, 6, 7
 2030 DATA D,-1,-3,E,-2, 8,F, 9,10
 2040 DATA G,-6,-4,H,-4,-7,I,-7,-1
```

JobLog (Part 1)

Jeff Gorman presents a multi-layered personal organiser.

This program can be used in a variety of different ways and at different levels to organize your time. The program *only* operates under ADFS and will also run under RISC OS 2 on an Archimedes.



```
Base date:- Today, Monday 13.04.1992                                     Lead times are
Personal......................                                           from 00:00 hrs
                                  Deadline ——— Lead time
Plumber installing new shower.   Mon 02.03.92 -   6 w 0 d                Log Event/Data
Swimming Gala.................   Sat 14.03.92 -   4 w 2 d                Extend Event/Data
Private View - College of Art.   Wed 18.03.92 -   3 w 5 d                Delete a row
Barbecue.....................*   Sat 21.03.92 -   3 w 2 d                Modify a row
A/at mothers for week.........   Mon 11.05.92 +   4 w 0 d                Sort this list
Puppies due...................   Mon 11.05.92 +   4 w 0 d                Print this list
House painter starts..........   Mon 11.05.92 +   4 w 0 d                Save this list
Garden Shed Delivery/Erection.   Mon 11.05.92 +   4 w 0 d                Change base date
Boiler servicing due..........   Fri 05.06.92 +   7 w 4 d
Collect chair from upholsterer   Wed 10.06.92 +   8 w 2 d                ESCAPE
Peter & Jane visiting.........   Thu 11.06.92 +   8 w 3 d                to go back
Car MOT due...................   Thu 11.06.92 +   8 w 3 d
Checkup-Clinic................   Wed 01.07.92 +  11 w 2 d
Dentist-Checkup...............   Tue 07.07.92 +  12 w 1 d                Use the cursor
Dolphin Club dinner...........   Sat 18.07.92 +  13 w 5 d                keys to select
School sports day.............   Tue 21.07.92 +  14 w 1 d                the event
Peter & Jane's wedding........   Sat 25.07.92 +  14 w 5 d                and key RETURN
Summer holiday................   Sat 01.08.92 +  15 w 5 d
Roger's course starts.........   Sat 01.08.92 +  15 w 5 d
Car guarantee expires.........   Sat 29.08.92 +  19 w 5 d
Speaking at Rotary Club.......   Tue 15.09.92 +  22 w 1 d
Birthdays & Anniversaries.....*  ........................
Childrens holiday dates.......*  ........................
Computer society meetings.....*  ........................
Present for A's birthday......   ........................
Solicitor.....................   Get suitable date.......
```

JobLog in full flight

The illustration above shows the general idea behind *JobLog*. At its most basic, it can be used as a simple reminder or appointment list. Since any item in a list (e.g. those shown with an asterisk) can be sub-divided into a maximum of 29 parts (and again and again, up to five sub-divisions) it can become a useful aid when working out the details of projects or classifying simple data.

The figure shows dates and memos but times of day can also be entered, e.g. in the 'Appointments' extension. Dated items have been sorted by date, with memos sorted into the alphabetical order of the first two letters of the event/data title. Items containing times of day in the format 'hh:mm' can also be sorted. The sort facility enables items to be added as one thinks of them, and then quickly put into an order which gives a rough idea of urgency.

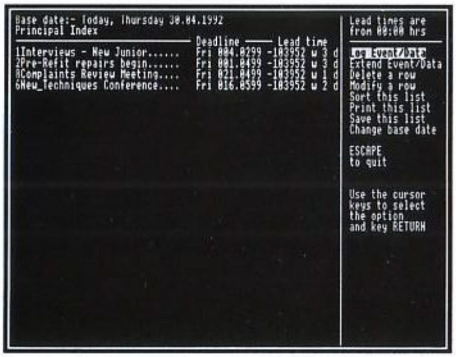
Each date or interval in a sub-list can be calculated either forwards or backwards from the current date, from a different base date, or from the project deadline set up in its 'parent' item. The idea behind the base date change is to allow for planning the following week's work on, say, a Friday.

Lists can be printed in three sizes, enabling filing in a ring-bound Personal Organiser or slipping inside an A5 sized diary or in an A4 sized project file, ringbound or otherwise. Organiser users need not obtain expensive specially perforated paper since at least one large stationery chain markets a plastic punch which can be used, a couple of sheets at at time, on ordinary paper.

USING THE PROGRAM

This month's listing will enable the Principal Index to be created, dated and saved. Next month's issue will contain a listing which adds the remaining features referred to above.

Make sure you have selected ADFS as the filing system for your computer, and format a disc to ADFS format by using the 'Aform' option from the ADFS Utilities on the Master's Welcome disc. Dedicate the disc only to this program. Type in the program listed here. Since this is a partial listing, take care to follow the line numbering exactly as printed. Save the listing with the filename 'JobLogOne' in the $ directory.



```
Base date:- Today, Thursday 30.04.1992                          Lead times are
Principal Index                                                 from 00:00 hrs
──────────────────────────── Deadline ───── Lead time
1Interviews - New Junior......  Fri 004.0299 -103952 w 3 d      Log Event/Data
2Pre-Refit repairs begin......  Fri 001.0499 -103952 w 3 d      Extend Event/Data
8Complaints Review Meeting....  Fri 021.0499 -103952 w 1 d      Delete a row
6New_Techniques Conference....  Fri 016.0599 -103952 w 2 d      Modify a row
                                                                Sort this list
                                                                Print this list
                                                                Save this list
                                                                Change base date

                                                                ESCAPE
                                                                to quit

                                                                Use the cursor
                                                                keys to select
                                                                the option
                                                                and key RETURN
```

Starting time management

Next, create a !Boot file as follows:

```
*Build $.!Boot
CHAIN 'JobLogOne'
```

Press Escape and type:

```
*OPT4,3
```

Pressing Shift-Break with the disc inserted will then start the program, and offer a dotted line for overtyping the first job/event title. Since the extension filenames are based on the first three letters of the job/data title and its (invisible) record number, characters not acceptable to the ADFS (spaces, dots, asterisks and colons) are not accepted within the first three characters of the entry. Use a hyphen or underline character if a word break is needed. For the time being, pressing Return then allows you to enter a deadline in the format dd.mm.yyyy.

The cursor keys enable you to run up and down the menu. Selecting options not yet available will drop you out of the program, but for the time being you can use the 'Save' option as well as 'Log Event/Data'. In the full version, each touch of the Escape key will take the screen back a stage to eventually quit, with the option of keying F4 to restart.

Because the structure of the software is so flexible, it's the kind of thing that will benefit from an exploration by the user rather than detailed instruction. We will be completing the program next month with a further explanation of its functions.

```
   10 REM Program JobLogOne
   20 REM Version B 1.0
   30 REM Author  Jeff Gorman
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
   70 ON ERROR GOTO 130
   80 arc=(INKEY-256>=160 AND INKEY-256<
=164):IF arc THEN *Country Master
   90 IF NOT arc THEN *Shadow
  100 DIM task$(29),sort%(29),fnme$(5),t
itle$(5),opt$(50):*TV0,1
  110 MODE 128:PROCtopWnd:PROCwnd("line"
):PROCtodayDt:PROCinitialise:PROCreset:P
ROCreadData:END
  120 :
  130 ON ERROR OFF:VDU26:REPORT
  140 PRINT" at line ";ERL:*CLOSE
  150 PROCcursor(1):END
  160 :
  170 DEF PROCtopWnd
  180 PRINT TAB(59)CHR$(169)" Lead times
 are";
  190 PRINT TAB(59,1)CHR$(169)" from 00:
00 hrs ";
  200 PRINT TAB(0,2)STRING$(32,CHR$(166)
)" Deadline " STRING$(5,CHR$166)" Lead t
ime "SPC(1) CHR$(169) SPC(1);:PRINT STRI
NG$(17,CHR$(166));:ENDPROC
  210 :
  220 DEF PROCwnd(l$)
  230 IF l$="blank" VDU28,61,31,79,3,12
  240 IF l$="mid" VDU28,0,31,58,3
  250 IF l$="topSide" VDU28,61,15,79,3
  260 IF l$="botSide" VDU28,61,31,79,15,
12
  270 IF l$="line" PROCcursor(0):VDU28,5
9,31,60,3:FOR c%=0 TO 28:PRINT TAB(0,c%)
```

```
CHR$(169);:NEXT:VDU26,31,0,0
  280 ENDPROC
  290 :
  300 DEF PROCcursor(s%)
  310 VDU23,1,s%;0;0;0;:ENDPROC
  320 :
  330 DEF PROCtodayDt
  340 day$ =MID$(TIME$,5,2)
  350 mth$ =STR$(FNconvertDate)
  360 year$=MID$(TIME$,12,4)
  370 PROCgetDatum(day$,mth$,year$)
  380 todayDt$=datumDt$:ENDPROC
  390 :
  400 DEF FNconvertDate
  410 mth$=MID$(TIME$,8,3):RESTORE 440
  420 mth%=0:REPEAT:mth%=mth%+1:READ mon
$:UNTIL mon$=mth$ OR mth%=12:=mth%
  430 :
  440 DATA Jan,Feb,Mar,Apr,May,Jun,Jul,A
ug,Sep,Oct,Nov,Dec
  450 :
  460 DEF PROCgetDatum(day$,mth$,year$)
  470 day$=FNpad(2,day$,"0",-1)
  480 mth$=FNpad(2,mth$,"0",-1)
  490 datumDt$=day$+"."+mth$+"."+year$
  500 strtDtID%=FNdayID(VALday$,VALmth$,
VALyear$):name$=FNdayName(strtDtID% MOD
7)
  510 ENDPROC
  520 :
  530 DEF FNgetDate(typ$):ok=TRUE:out=0
  540 REPEAT:PROCwnd("blank")
  550 PRINT "Enter "typ$" date"''"(dd.mm
.yyyy)"
  560 VDU31,0,3:date$=FNip(10,no$+".",-1
,-1,0)
  570 ok1=MID$(date$,3,1)="." AND MID$(d
ate$,6,1)="."
  580 ok2=VAL(MID$(date$,4,2)) >0 AND VA
L(MID$(date$,4,2)) < 13
  590 ok3=VAL(LEFT$(date$,2)) > 0 AND VA
L(LEFT$(date$,2)) <= FNmonLen(VAL(MID$(d
ate$,4,2)),VAL(RIGHT$(date$,2)))
  600 ok4=VAL(RIGHT$(date$,4))>1989 AND
VAL(RIGHT$(date$,4))<2080
  610 ok=(ok1 AND ok2 AND ok3 AND ok4) O
R esc:IF NOTok PROCmistake
  620 UNTIL ok OR out
  630 IF out:="" ELSE:=date$
  640 :
  650 DEF FNip(len,valid$,brackets,escap
e,memo):PROCcursor(1):esc=0
  660 LOCALa$:a$="":l$="":r$="":e%=0
  670 end=0
  690 IF brackets l$="<":r$=">":e%=1: *f
x21,0
  700 PRINT l$ STRING$(len,".") r$;STRIN
G$(len+e%,CHR$8);:REPEAT G%=GET
  710 accept=G%>27 AND G%<127
  720 IF (G%=32 OR G%=46) AND (len=30 AN
D LENa$<3) VDU7:UNTIL FALSE
  750 IF G%=13 AND a$="" AND NOT memo VD
U7:UNTIL FALSE
  760 IF G%=13 AND a$>"" end=TRUE:*fx229
  770 IF G%=127 AND a$="" UNTIL FALSE
  780 IF G%=127 a$=LEFT$(a$,LEN(a$)-1):P
RINT CHR$G%;".";CHR$8;:UNTIL FALSE
  790 IF (G%>13 AND G%<127 AND G%<>27) A
ND (LEN(a$)=len OR INSTR(valid$,CHR$G%)=
0) VDU7:UNTIL FALSE
  800 IF accept b$=CHR$G%:PRINT b$;:a$=a
$+b$
  810 UNTIL end:IF memo AND a$="":=STRIN
G$(24,".")
  820 =a$
  830 :
  840 DEF PROCmistake:VDU7,28,61,31,79,8
,12:*fx200,1
  850 PRINT''"Error - Press any"'"key to
 start again":G=GET:CLS:*fx200,0
  860 ENDPROC
  870 :
  880 DEF FNsplitDate(dt$,s%,n%)
  890 =VAL(MID$(dt$,s%,n%))
  900 :
  910 DEF FNmonLen(mon%,yr%):=30+ABS((mo
n%>7)+(mon% AND 1))+(mon%=2)*(2+(FNlpYr(
yr%)))
  920 :
  930 DEF FNlpYr(yr%):=(yr%MOD4=0 AND yr
%MOD100<>0) OR yr%MOD400=0
  940 :
```

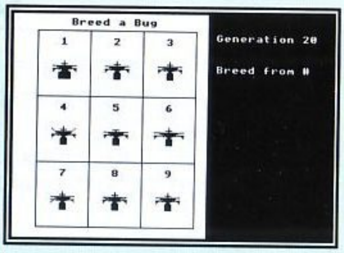
```
 950 DEF FNdayID(days%,mnth%,yr%)
 960 yearDays%=365.25*(yr%-1)
 970 IF mnth%=1:=days%+yearDays%
 980 monLen%=0:FOR m%=1 TO mnth%-1
 990 monLen%=monLen%+FNmonLen(m%,yr%)
 1000 NEXT:=days%+monLen%+yearDays%
 1010 :
 1020 DEF FNexecMonth(exDy%):agg%=0:h%=0
:REPEAT:h%=h%+1:pAgg%=agg%
 1030 agg%=agg%+FNmonLen(h%,execYr%)
 1040 UNTIL agg%+1 > exDy%:em$=STR$(h%)
 1050 IF h%<=9 em$="0"+em$
 1060 =em$
 1070 :
 1080 DEF FNexecDay(exDy%,em$,ey%)
 1090 execDate%=execDayNo%-pAgg%
 1100 IF execDate%<=9 execDate$="0"+STR$
execDate% ELSE execDate$=STR$execDate%
 1110 =execDate$
 1120 :
 1130 DEF FNdayName(rem%):LOCAL no%,:no%
=-1:RESTORE 1160:REPEAT:no%=no%+1
 1140 READ prefx$:UNTIL no%=rem% OR no%=
6:=prefx$+"day"
 1150 :
 1160 DATA Satur,Sun,Mon,Tues,Wednes,Thu
rs,Fri
 1170 :
 1180 DEF PROCinitialise
 1190 cap$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 1200 punct$="!#%&@`|{}""?()<>/\'*+:;,._
-=":lc$="abcdefghij klmnopqrstuvwxyz"
 1210 no$="#01234 56789-":bl%=0:wh%=7
 1220 let$=punct$+cap$+lc$:opt%=0:new=0
 1230 rw%=0:tempDtID%=0:spc%=2
 1240 pLen%=66:tearOff%=6:botM%=5:free%=
pLen%-(tearOff%+botM%):newSheet=TRUE
 1250 choice%=0:ch%=0:index=TRUE:prnt=0
 1260 event$=STRING$(35,"e"):event$="":r
eplace$=STRING$(60,"r"):replace$=""
 1270 depth$="0":s$="":fnme$(0)="Index"
 1280 memoMrk$=" ":extndDln$="":star=0
 1290 full=0:mod=0:change=0:level%=0
 1300 noDlne=0:fn$="Index":dtData$=""
 1310 title$(0)="   Principal Index"+STR
ING$(15," ")+"$@0":extend=0:esc=0:same=0
```

```
 1320 extndMrk$="$":*fx200,0
 1330 VDU2,1,27,1,64,3:*KEY4 RUN|M
 1340 ENDPROC
 1350 :
 1360 DEF PROCreset:delete=0:memo=0:esca
pe=0:printMenu=0:topMenu=0:intvlMenu=0
 1370 predMenu=0:skip=0:esc=0:ENDPROC
 1380 :
 1390 DEF PROCreadData
 1400 LOCAL max%:PROCclear(0)
 1410 index=(level%=0)
 1420 IF extend AND mod fnme$(level%)=ol
dFnme$
 1430 IF index fnme$="Index" ELSE fnme$=
fnme$(level%)
 1440 hndl%=OPENUP("$."+fnme$)
 1450 IF hndl%=0 row%=-1:PROCenterData:E
NDPROC
 1460 PROCwnd("mid"):CLS
 1470 VDU31,0,0:PROCcursor(0)
 1480 max%=0:row%=-1:PTR#hndl%=&40:REPEA
T:row%=row%+1
 1490 INPUT#hndl%,task$(row%)
 1500 rcd$=LEFT$(task$(row%),3)
 1510 IF VALrcd$>0 PROCformRow(task$(row
%),row%)
 1520 IF VALrcd$>max% max%=VALrcd$
 1530 rcdNo%=max%:UNTIL EOF#hndl% OR rcd
$="":CLOSE#hndl%:PROCclear(row%)
 1540 row%=row%-1:PROCmenu:ENDPROC
 1550 :
 1560 DEF PROCenterData:PROCwnd("mid"):r
ow%=row%+1:rcdNo%=rcdNo%+1:new=(row%=0)
 1570 rcd$=FNpad(3,STR$(rcdNo%),"0",-1)
 1580 PROCwnd("blank"):PROCgetTask(row%)
 1590 IF NOT skip PROCformRow(task$(row%
),row%)
 1600 PROCmenu:ENDPROC
 1610 :
 1620 DEF PROCgetTask(rw%):mod=(opt%=3)
 1630 extndMrk$=MID$(task$(rw%),34,1)
 1640 star=(extndMrk$="*"):same=0
 1660 IF NOT star extndMrk$="$"
 1690 PROCwnd("blank")
 1700 PRINT"Enter a heading";
 1710 PROCwnd("mid")
```

```
1720 PRINT TAB(0,rw%);
1730 event$=FNip(30,let$+no$,0,-1,0)
1770 event$=FNpad(30,event$,".",0)
1780 PROCrhColumn(rw%)
1790 ENDPROC
1800 :
1810 DEF PROCformRow(rw$,rw%)
1830 PRINT MID$(rw$,4,30);
1930 exDy%=VAL(MID$(rw$,37,2))
1940 em%=VAL(MID$(rw$,40,2))
1950 ey%=VAL(MID$(rw$,43,4))
1960 execID%=FNdayID(exDy%,em%,ey%)
1970 daysToExec%=execID%-strtDtID%
1980 sort%(rw%)=execID%
1990 IF NOT skip PROCprintRHS(daysToExe
c%,execID%,rw$,rw%)
2000 new=0:ENDPROC
2010 :
2020 DEF PROCprintRHS(time%,dayID%,rw$,
rw%)
2030 dayName$=LEFT$(FNdayName(dayID% MO
D 7),3)
2040 PRINT STRING$(spc%," ");
2050 IF NOTprnt VDU31,31+spc%,rw%;
2060 week%=ABS(time% DIV 7)
2070 week$=FNpad(4,STR$week%," ",-1)
2080 dy%=time% MOD7:day$=STR$(ABS(dy%))
:dy$=" d":a%=0
2090 IF week%>=1 wk$=" w "ELSE PROCwhen
2100 sign$=" ":IF time%>=2 sign$="+"
2110 IF time%<-1 sign$="-"
2120 PRINT dayName$;
2130 PRINT SPC(1)MID$(rw$,37,3);
2140 PRINT MID$(rw$,40,3);
2150 PRINT MID$(rw$,45,2);
2160 PRINT SPC(1)sign$ SPC(a%)week$ wk$
 day$;:IF rw%<28 PRINT dy$ ELSE PRINTdy$
;
2170 ENDPROC
2180 :
2190 DEF PROCwhen:week$="":wk$="":a%=7
2200 IF dy% = 0 a%=5:day$="Today"
2210 IF dy% = 1 a%=2:day$="Tomorrow"
2220 IF dy% =-1 a%=1:day$="Yesterday"
2230 IF dy%>=-1 AND dy%<2 dy$=""
2240 ENDPROC
2250 :
2260 DEF PROCrhColumn(rw%)
2270 skip=0:predMenu=TRUE:PROCwnd("mid"
):replace$="":PROCwnd("blank")
2280 mod=(opt%=3):index=(level%=0)
2290 memoMrk$=MID$(task$(rw%),35,1):IF
mod extndMrk$=MID$(task$(rw%),34,1)
2300 choice%=FNmarker(3,-1)
2340 IF choice%=0 PROCenterDlnDt(rw%)
2380 PROCrvrs(wh%,bl%):PROCwnd("mid"):P
RINT TAB(0,rw%)LEFT$(event$,30);
2390 change=TRUE:IF NOT mod PROCmenu
2400 ENDPROC
2410 :
2420 DEF FNmarker(numItms%,escape):menu
=(topMenu OR predMenu OR intvlMenu OR pr
intMenu):IF menu PROCwnd("blank")
2430 PROCcursor(0):esc=0:IF new escape=
0
2460 v%=0:*fx 4,1
2470 IF menu PROCshowMenu:numItms%=num%
 ELSE PROCwnd("mid")
2480 last%=numItms%:VDU31,0,0
2490 REPEAT:PROCrvrs(bl%,wh%)
2500 IF NOT menu opt$=MID$(task$(v%),4,
30) ELSE opt$=opt$(v%)
2510 PRINT TAB(0,v%)opt$;
2520 key%=GET:PROCrvrs(wh%,bl%)
2530 PRINT TAB(0,v%)opt$;
2540 v%=v%+(key%=139)-(key%=138)
2550 IF v%>last% v%=0
2560 IF v%<0 v%=last%
2570 IF NOT menu opt$=MID$(task$(v%),4,
30) ELSE opt$=opt$(v%)
2580 IF key%=27 esc=TRUE:PROCrvrs(wh%,b
l%):PRINT TAB(0,v%)opt$;:=v%:key%=13
2590 PROCrvrs (bl%,wh%)
2600 PRINT TAB(0,v%)opt$;
2610 UNTIL key%=13:*fx4,0
2620 *fx12,0
2630 *fx200
2640 *fx229,0
2650 PROCrvrs(wh%,bl%):=v%
2660 :
2670 DEF PROCshowMenu
2680 IF topMenu num%=8+(level%<2):PROCg
```

```
etList(0,num%)
 2690 IF predMenu num%=0:PROCgetList(10,
num%)
 2720 PROCrubric("option",full):
 2730 PROCwnd("topSide"):ENDPROC
 2740 :
 2750 DEF PROCrvrs(fgd%,bgd%):COLOUR fgd
%:COLOUR 128+bgd%:ENDPROC
 2760 :
 2770 DEF PROCgetList(incr%,n%):IF index
 AND NOT prnt m$="quit" ELSE m$="go back
"
 2780 PROCwnd("blank"):FOR clear%=0 TO 5
0:opt$(clear%)="":NEXT
 2790 RESTORE (2850+incr%)
 2800 z%=-1:REPEAT:z%=z%+1:READ opt$(z%)
 2810 PRINT TAB(0,z%) opt$(z%)
 2820 UNTIL z%=n%
 2830 ENDPROC
 2840 :
 2850 DATA Log Event/Data,Extend Event/D
ata,Delete a row,Modify a row,Sort this
list,Print this list,Save this list,Chan
ge base date,Back to index
 2860 DATA Deadline date,Interval,Time a
nd/or Memo,No change
 2870 DATA From Base date,From a new dat
e,From the "Due" date
 2880 DATA Organiser size,A5 size,A4 siz
e
 2890 :
 2900 DEF PROCclear(r%):IF r%<0 r%=0
 2910 FOR clear%=r% TO 29
 2920 task$(clear%)="":NEXT:ENDPROC
 2930 :
 2940 DEF FNpad(nu%,e$,symb$,lead)
 2950 IF lead:=STRING$(nu%-LEN(e$),symb$
)+e$
 2960 =e$+STRING$(nu%-LEN(e$),symb$)
 2980 :
 3070 DEF PROCmenu:full=(row%>=28)
 3080 PROCreset:PROCwnd("blank"):PROCwnd
("mid"):topMenu=TRUE
 3090 opt%=FNmarker(numItms%,-1)
 3100 topMenu=0:IF esc AND extend PROCre
trace:ENDPROC ELSE IF esc PROCquit:ENDPR
OC
 3110 PROCreset:IF opt%=0 AND full VDU7:
PROCmenu
 3120 IF opt%=0 PROCenterData
 3180 IF opt%=6 PROCwriteData(fnme$(leve
l%)):PROCmenu
 3210 ENDPROC
 3220 :
 3230 DEF PROCwriteData(fn$):*CLOSE
 3240 IF task$(0)="" AND NOT esc PROCnot
hing("save"):ENDPROC
 3250 IF same AND skip AND opt%<>5 ENDPR
OC
 3260 PROCwnd("blank")
 3270 PRINT"Updating "'"Please wait"
 3280 hndl%=OPENOUT("$."+fn$)
 3290 PRINT#hndl%,title$(level%):PTR#hnd
l%=&40:save%=-1:REPEAT:save%=save%+1
 3300 PRINT#hndl%,task$(save%):UNTIL sav
e%=29:CLOSE#hndl%:change=0:ENDPROC
 3310 :
 3400 DEF PROCrubric(rubric$,full)
 3410 PROCwnd("botSide")
 3420 IF full PRINT "***Sheet full***"
 3440 ENDPROC
 3450 :
 3580 DEF PROCenterDlnDt(rw%):memoMrk$="
 ":hold$=FNgetDate("deadline")
 3600 IF hold$="" ENDPROC
 3610 day%=FNsplitDate(hold$,1,2)
 3620 mon%=FNsplitDate(hold$,4,2)
 3630 yr% =FNsplitDate(hold$,7,4)
 3640 dd$=FNpad(2,STR$(day%),"0",-1)+"."
 3650 dm$=FNpad(2,STR$(mon%),"0",-1)+"."
 3660 dyr$=STR$(yr%)
 3670 IF mod extndDln$=dd$+dm$+dyr$
 3680 IF VALdd$ > FNmonLen(VALdm$,VALdyr
$) VDU7:PROCenterDlnDt(rw%)
 3690 task$(rw%)=FNbuildRow(dd$+dm$+dyr$
):PROCwnd("mid")
 3700 PROCformRow(task$(rw%),rw%)
 3710 ENDPROC
 3720 :
 3730 DEF FNbuildRow(dtData$)
 3740 =rcd$+event$+extndMrk$+memoMrk$+ST
R$level%+dtData$
```

Applications I Disc



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

File Handling for All

on the BBC Micro and Acorn Archimedes

by David Spencer and Mike Williams

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

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

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

The topics covered by the book include:

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

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

ASTAAD

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

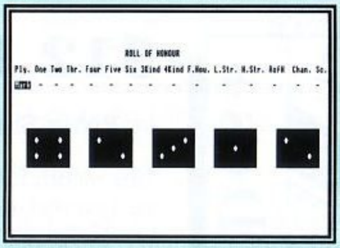
- full mouse and joystick control
- built-in printer dump
- speed improvement
- STEAMS image manipulator
- Keystrips for ASTAAD and STEAMS
- Comprehensive user guide
- Sample picture files

	Stock Code	Price
ASTAAD (80 track DFS)	1407a	£5.95
Applications II (80 track DFS)	1411a	£4.00
Applications I Disc (40/80T DFS)	1404a	£4.00
General Utilities Disc (40/80T DFS)	1405a	£4.00
Arcade Games (40/80 track DFS)	PAG1a	£5.95
Board Games (40/80 track DFS)	PBG1a	£5.95

	Stock Code	Price
ASTAAD (3.5" ADFS)	1408a	£5.95
Applications II (3.5" ADFS)	1412a	£4.00
Applications I Disc (3.5" ADFS)	1409a	£4.00
General Utilities Disc (3.5" ADFS)	1413a	£4.00
Arcade Games (3.5" ADFS)	PAG2a	£5.95
Board Games (3.5" ADFS)	PBG2a	£5.95

All prices include VAT where appropriate. For p&p see Membership page.

Board Games



ROLL OF HONOUR
Ply. One Two Thr. Four Five Six 3Kind 4Kind F.Hou. L.Str. H.Str. RoFH Chan. Sc.

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

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

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

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

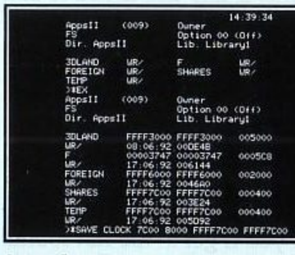
CRIBBAGE - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

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

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

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

Applications II Disc



```
                                   14:39:34
AppsII     (009)        Owner
FS                      Option 00 (Off)
Dir. AppsII             Lib. Library1

3DLAND      WR/         F            WR/
FOREIGN     WR/         SHARES       WR/
TEMP        WR/
>*EX
AppsII     (009)        Owner
FS                      Option 00 (Off)
Dir. AppsII             Lib. Library1

3DLAND      FFFF3000 FFFF3000   005000
WR/         08:06:92 00DE48
F           00003747 00003747   0005C8
WR/         17:06:92 006144
FOREIGN     FFFF6000 FFFF6000   002000
WR/         17:06:92 0046A0
SHARES      FFFF7C00 FFFF7C00   000400
WR/         17:06:92 003E24
TEMP        FFFF7C00 FFFF7C00   000400
WR/         17:06:92 005D92
>*SAVE CLOCK 7C00 8000 FFFF7C00 FFFF7C00
```

CROSSWORD EDITOR - for designing, editing and solving crosswords

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

3D LANDSCAPES - generates three dimensional landscapes

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

RUNNING FOUR TEMPERATURES - calibrates and plots up to four temperatures

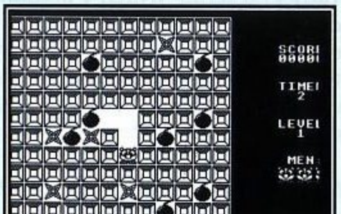
JULIA SETS - fascinating extensions of the Mandelbrot set

FOREIGN LANGUAGE TESTER - foreign character definer and language tester

SHARE INVESTOR - assists decision making when buying and selling shares

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

Arcade Games



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

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

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

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

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

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

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

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

	Stock Code	Price
File Handling for All Book	BK02b	£ 9.95
File Handling for All Disc (40/80T DFS)	BK05a	£ 4.75
Joint Offer book and disc (40/80T DFS)	BK04b	**£ 11.95**
Magscan (40 DFS)	0005a	£ 9.95
Magscan (80T DFS)	0006a	£ 9.95
Magscan (3.5" ADFS)	0007a	£ 9.95

	Stock Code	Price
File Handling for All Disc (3.5" ADFS)	BK07a	£ 4.75
Joint Offer book and disc (3.5" ADFS)	BK06b	**£ 11.95**
Magscan Upgrade (40 DFS)	0011a	£ 4.75
Magscan Upgrade (80T DFS)	0010a	£ 4.75
Magscan Upgrade (3.5" ADFS)	1458a	£ 4.75

All prices include VAT where appropriate. For p&p see Membership page.

Tel. (0727) 40303   Fax. (0727) 860263

512 Forum

by Robin Burton

This month I've an interesting tweak for you which really would have fitted in better with a topic covered in the Forum about a year ago.

Since this item is already two years 'late' (or one, depending on which way you look at it) I thought I'd leave the software compatibility information mentioned last month for another Forum.

ECHO OFF AGAIN

You might remember that in Volume 10, Forum discussions centred on ways of removing unwanted screen output from commands executed within batch files.

Well I've recently again come across (or found, rather) a letter which I actually received a couple of years ago from a member about this very subject. Richard prefers to remain anonymous, so I won't tell you any more about him, we'll just move on to the subject matter.

It seems that he'd been looking (quite literally) at COMMAND.COM, and had come across a block of text strings, each of which began with a "!". The particular one that raised his interest, indeed the one that was the reason for the whole exercise, was "!echo on". This, he thought, looked just like a normal manual command, but presumably the preceding "!" character meant something significant to COMMAND.COM. Could it be something perhaps like the default setting?

Before we go any further, I'd suggest that if you want to look at this yourself, and definitely if you intend to follow the exercise, then you put a copy of COMMAND.COM onto a work disc and use that version until you're satisfied everything is correct.

Throughout the following I'll assume that your current drive is A: and that it contains your boot disc plus the standard issue software. One other point to note is that none of this is any use if you're still using DOS Plus 1.2. That version of COMMAND.COM is quite different to 2.1's and the string we're interested in simply isn't there. 2.1 users read on...

If your real copy of COMMAND.COM is set to 'system' remember that you'll need to append the '/S' switch to the copy command, or you'll get a 'file not found' message. For example, to copy the file from A: to B: use:

```
COPY A:COMMAND.COM /S B:
```

Of course, if your copy of COMMAND.COM is visible on normal DIR displays you won't need to add the '/S'. When the file is on the work disc you'll probably also need to change its attributes to those of a normal file before you'll be able to update it, the command for which, assuming the file is on drive B: again and FSET is available on drive A;, is:

```
FSET B:COMMAND.COM [RW/DIR
```

This command can be issued even if you're not sure of the file's current attributes, so make sure you do it or you might get 'access denied' during the next stage of the procedure. By the way, the 'missing' right square bracket isn't a mistake; as I've mentioned before in the Forum it's not necessary.

MODIFYING COMMAND.COM

To look at the part of the file we're interested in, using EDBIN is probably the easiest method. Of course, if you have another binary editor and prefer to use it, do so by all means, but EDBIN is on issue disc 1, so everyone can follow this example.

To load the file for editing, the command is:

```
        EDBIN B:COMMAND.COM
```

which will produce a display of:

```
        Size of file = 006908 bytes
        Number of bytes read = 006908
        -
```

As you can see from the two figures, the entire file has been read into RAM so you can now examine any part of it. The '-' is an invitation to enter a command and the one we need is 'd', for display. If you want to refresh yourself on the EDBIN commands just enter 'h'. The 'd' command is followed by an offset address, which in this case is 5ddd. The format of the entry is simply as shown in figure 1, and the hex and ASCII display will be produced as soon as you press Return.

and the solution to this is the obvious one, which Richard discovered: just 'pinch' one of the two nulls before the D at the end of the line.

To do this you need to use the 'e' (edit) command, followed by the address you wish the cursor to appear at. If you're not familiar with EDBIN I'd suggest you enter:

```
        - e 5ddd
```

which will place the editing cursor at the beginning of the first line of the above display (just one line will appear this time), then move the cursor to the first of the two null bytes manually by pressing the right cursor key.

All you need to do now is to enter each of the hexadecimal values from the next byte on the left of the cursor position into the current byte, and then press cursor left to move to the previous byte to change that too.

For example, to 'move' the '0A' just enter 'A' then press cursor left. Next, enter 'D' followed by another cursor left, then

```
- d 5ddd
1974:5DDD  21  65  63  68  6F  20  6F  6E  00  25  73  0D  0A  00  00  44 !echo on.%s....D
1974:5DED  69  67  69  74  61  6C  20  52  65  73  65  61  72  63  68  20  igital Research
1974:5DFD  28  63  29  20  31  39  38  35  2C  20  31  39  38  36  20  2D (c) 1985, 1986 -
1974:5E0D  20  32  39  20  4D  61  79  20  38  36  00  EA  00  92  1F  00  29 May 86......
1974:5E1D  00  F0  00  4B  20  00  00  F6  00  4B  20  00  00  F9  00  D0 ...K ....K .....
1974:5E2D  38  AC  0B  FE  00  56  21  00  00  03  01  64  21  00  00  08 8....V!....d!...
1974:5E3D  01  10  23  AC  0B  0C  01  20  23  00  00  10  01  5D  32  AC ..#.... #....]2.
1974:5E4D  0B  15  01  10  23  AC  0B  19  01  5D  32  AC  0B  1E  01  10 ....#....]2.....
-
```

Figure 1: The area of interest

You can see the command '!echo on' at the start of the first line, and this is what we're going to change. You'll also notice that this command is terminated by a null, followed by '%s' then a carriage return and line feed, plus two more nulls before the DR copyright notice. Obviously if we change 'on' to 'off' we need an extra byte,

enter '73', press cursor left and so on. When the cursor gets to the zero byte following '6E', that is, you've just amended '25' to '00' and moved to the next byte left, enter '66' and move left one byte, then repeat the same entry. Your display should now appear as shown in figure 2.

When you're satisfied this is correct press Ctrl-C to end the edit (don't worry that this messes up the display, that's quite normal), and then you can save the file. If you want to check your changes before saving the file just issue the original display command again, d 5ddd, when the display will appear as in figure 1, but now showing the amended command string.

To write the file back to disc enter 'w' (write), followed by Return, then 'q' (quit) and Return to exit to DOS.

if you do at all, is up to you. I know of no problems, but there are no guarantees.

The amended COMMAND.COM appears to work quite normally, but by default 'echo' is off. The result is that whenever a batch file is run, none of the batch commands appear on screen unless you explicitly request it by including an echo on command. You can also, of course, turn echo off again later if you wish. If you read the previous articles on this topic there's one aspect of this 'tweak' which you might think could be a problem, but it isn't.

```
- d 5ddd
1974:5DDD  21  65  63  68  6F  20  6F  66  66  00  25  73  0D  0A  00  44  !echo off.%s...D
```

Figure 2: The amended bytes

All you now need to do to try this amended version of COMMAND.COM is to prepare a new boot disc, replacing the original copy of COMMAND.COM with your modified version. Again you might need to set the destination file to read-write/directory before the copy, and to make the new version of the program disappear from DIR displays, assuming the new disc is in drive A:, is to enter:

```
    FSET A:COMMAND.COM [RO/SYS
```

after the copy.

TESTING

Bear in mind that whenever you amend a piece of software based on operating principles of guesswork and inspiration but no hard facts, as is the case here, you're well advised to exercise extreme caution before you trust the new program completely. The advice is the same here, although Richard tried the amended version without trouble and I've tested it too, perhaps not exhaustively, but again with no problems. However, how soon you decide to use the amended COMMAND.COM as your live version,

To illustrate, having restarted the 512 from your new boot disc, try keying in the following example batch file. The most straightforward way is to enter:

```
    COPY CON TEST.BAT
```

followed by Return, then enter these lines pressing Return after each:

```
    DIR
    CLS
    CHKDSK
    PCSCREEN3
    COLOUR 1 4
```

After the last Return press Ctrl-Z and then Return again. You can now run the file by entering its name 'TEST'. The result is that each of the commands executes, any output to (or effect on) the screen is exactly as normal (you'll end up with blue text if you have a colour monitor) but you don't see any of the commands themselves. OK, so what?

If you check in Volume 10 Issue 6 you'll see that there are limitations to piping screen output to NUL: or to a temporary file to avoid screen echo. The problem is that the effect of operations that should affect the screen, like CLS, PCSCREEN and so on, is also suppressed.

As you can see from our test batch file above, with the modified version of COMMAND.COM there are no such problems, the screen commands operate correctly, and the reason is very simple. When you redirect output, such as piping to NUL: or a file, you change the destination of the output data stream. By contrast, this change to COMMAND.COM is (or should that be 'seems to be'?) exactly the same as the effects of the 'echo off' command, with the benefit that the command isn't needed.

Of course this modification prevents the batch commands appearing on screen, but just like a normal echo off command it doesn't suppress the resulting output from command execution. Should you also wish to avoid that you'll still need to use piping and the previous warnings still apply. For example, if you have a file copy command but don't want to see the output from it, including the terminating 'n files copied', the command should be:

```
    COPY FILE1.XTN FILE2.XTN >NUL:
```

as described before.

ODDS AND ENDS

A few issues ago I mentioned developments in processors for PCs and ended by saying the 512 was still more than quick enough for a good many jobs. I learned some interesting facts on this very point only last week.

I've been developing a complex software algorithm over the last few weeks and initially wrote the code in '86 assembler, since I'm most familiar with that these days and the 512 is (reasonably) fast. However, ultimately the technique was to be employed on an Archimedes, when it was finally fully developed and proved to work.

To cut a long story short, I had a 33K input file of test data which the 512 processed in 4.42 seconds. The program was converted to ARM code and run on an A310. With the same input data this took 6.9 seconds. The ARM code was then optimised for RISC techniques, when the run time dropped to 2.85 seconds. During last weekend I also tried the 512 program, using the same input, in an old Olivetti M24, an XT clone, which took 11.76 seconds, then in a 33Mhz 486, which took 0.76 seconds.

None of this is perhaps very useful information, but I thought it interesting that, at least in memory and processor operations, the 512 is almost three times the speed of an 8MHz XT. At the same time a 486/33 runs the same program almost six times faster (though its I/O is MUCH faster) but what frankly amazed me was that, even running highly optimised code, an ARM2 is only about 35% quicker than my 512!

The moral seems to me to be - hang onto your 512 (and your cash) for a while yet!

I had another letter from a member a while ago which mentioned an interesting way of getting some programs to run that otherwise wouldn't. It involves running them within the D86 debugger, part of the shareware assembler package I mentioned a month or two back.

It seems that some programs that normally crash the 512 will run correctly if they're run in D86. All that's needed is to load the debugger and the program in the normal way, then issue the 'g' (go) instruction. This effectively makes the debugger transparent, but when the program in question issues an interrupt call that would usually crash the 512, the debugger traps the call and executes it itself. It obviously does this by much

continued on page 40

BEEBUG Education

by Mark Sealey

Product:	**The Bowbridge Pack**
Supplier:	**Scottish Council for Educational Technology, Dowanhill, 74 Victoria Crescent Road, Glasgow G12 9JN.**
Price:	**£12.50 ex. VAT**

The Scottish Council for Educational Technology has featured before on these pages in reviews of software for the BBC. Here it is again with a relatively minor yet well put together item. This will be of use to teachers working in history and technology (Information Technology Capability in English schools) as well as in Scottish schools. Here - as the publishers point out - there is "obvious relevance to the Standard Grade course unit on *Historical Investigation.*"

The Bowbridge Pack is a data file available in Apple Mac and Archimedes formats as well as for the BBC B and Master 128. For the 32-bit Acorn machines, the disc contains two text formats (tab and CR separators) as well as a version for the popular educational database Key+.

The 8-bit file (which comes on a 5.25" 40 track DFS disc) is available only for the Key database (ITV Schools) and as such would seem at first glance a relatively insignificant product. It is certainly very cheap. But the principles underlying the pack make an excellent illustration of some aspects of best practice in the educational use of computers.

SUBJECT MATTER

To produce this database the authors have drawn on the University of Dundee Library Archives Department to compile a register of accidents at work which occurred at the Bowbridge Jute Works in Dundee between 1896 and 1897.

This may seem of about as much interest as your dentist's family tree. Yet it is exactly the sort of body of information of which true historical inquiry is made. To use such a file is to engage, at the same time, on about as extreme a form of open ended and closed IT activity as is possible.

WELCOME TO THE BOWBRIDGE WORLD

The reason why this product deserves to succeed lies in the fact that it IS so self-contained: rarely, at its level, can so exhaustive a data set have been made so easily studiable. It is its own little (and factually accurate) microworld. To explore it is not unlike working inside some of the LOGO microworlds that were created for secondary schools in the early days of LOGO availability on the BBC computers.

There can have been little remarkable about what went on in a Jute factory almost a hundred years ago - except its typicality. The publishers claim that much can be learnt about life and work at that time. This review needs to substantiate or disprove such a claim; on this - and whether it has a place both in the wider curriculum and whether its use can develop valid study skills in their own right - the overall worth of the package can be assessed.

THE DATAFILE

So, Bowbridge is a 76K data file in Key format. There are 167 records in 16 fields covering the victim of the accident, its severity and where and how it happened etc. Each record is given a unique number and the data is of several types, including a code for the 'Injury Class' and textual notes where extra information has been recorded. Since the subjects of some of the records are young people, there is added interest to pupils of compulsory education age.

SUPPORTING MATERIAL

One of the classic ways of persuading the non-IT specialist (and there are still many of these) that the dreaded keyboard has a place in their discipline or can extend their expertise is by bringing it onto home territory. Computer time-tabling for the administrator, e-mail for the foreign exchange teacher, mapping packages for the geographer, and historical data for the historian are such examples.

As an approach, this is less likely to work if there is any excuse. Poor documentation and hence user-unfriendliness would come near the top of any list of such excuses.

But the SCET manuals are exemplary: there are three of them: a 28-page teacher's booklet that deals with everything from copying and arranging discs, use of the computer in the school, adult references and curriculum context to precise control and manipulation of the data in each relevant database. To have their hands held as they explore familiar or potentially familiar material in this way often does wonders for the reluctant microphone.

FOR THE PUPILS

The Study Book contains nearly 40 pages of background information on the Bowbridge works, which began in the middle of the last century and was not closed down until about fifteen years ago. It deals with the purpose and accuracy of accident registers and - more importantly - the reasons for their inception and place in the world of labour nowadays. In that the subsequent role of the International Labour Organisation is mentioned, there is scope for secondary teachers in social studies and tutors of adult education classes to stimulate discussion.

Pupils studying textile production or even, possibly, those working in training in the industry - and definitely on certain management courses - would find the sections in this book on how the industry has changed quite enlightening. This information is presented well, in a readable format, at a leisurely yet informative pace and always setting specific developments in the wider historical context.

But the most impressive of the three supporting materials is the 'Tasks' booklet. The layout is clear, materials are freely copiable and are aimed to take the pupil through a series of tasks using both the other SCET documents and the database itself. There is also a facsimile page of the actual register. In all cases it is the historical and social subject matter that dominates, not the computer.

Perhaps just a little too much is assumed about using Key (or Key+). Some simpler tasks such as: "Find how many workers actually lost their lives" or "Find how many accidents happened after 8pm" or even "Move to the fifth record of the file" would have been useful before passing on to the more analytical tasks. Such basic ground is covered in the teacher's book, though, and it would be up to the teacher/tutor to prepare tasks of this kind.

SOUND PRACTICE

SCET clearly expects the material to be used in a certain way. The documentation and manuals support this approach. What is so commendable is that it almost inevitably promotes *use* of the computer in a significant and relevant context to the growing understanding of the student to explain and explore underlying historical principles, trends and changes. To have so much primary material on hand in a readily accessible format makes very good sense and encourages active inquiry and active, purposeful use of the computer - in preference to the umpteenth scrutiny of class shoe sizes or birthdays.

A thorough immersion in out-of-the-way Bowbridge is quite likely to whet the appetite for further, directed, use of the computer in context or to sharpen existing skills.

From a historical point of view, the field of accidents at work is an important one; specifically so because it was a requirement that resulted directly from an act of parliament (in 1895) and can be seen (unlike, perhaps, the many variables handled in more complex worlds of Fletcher's Castle or Mary Rose) to have had specific causes and clear results, all of which are perceivable.

Lastly, from a purely educational point of view, the tasks in the book are specifically structured so as to foster pupils' intelligent use of evidence to test hypotheses and ability to draw sound conclusions.

CONCLUSION

This product would appear to be of somewhat limited appeal. Despite dealing with an admittedly specialist area, it does so in an exemplary way and - thanks to the outstanding supplementary (yet essential) documentation - could be thought of as the very best practice in IT in an often neglected area of the curriculum. Recommended. B

512 Forum (continued from page 37)

debugger traps the call and executes it itself. It obviously does this by much more legal and well behaved methods, because the application doesn't crash the system.

I did say that A86/D86 is a remarkable package and this further demonstrates the fact that it's an extremely well written piece of software. However, top marks to one member for finding a method of curing some problem programs that I must admit I doubt I'd have thought of, even though I use A86 regularly. Try this on your problem programs, and it might very well work when all other approaches have failed.

As a final snippet this month, Dr. Sutton, a member from Swansea, has recently expanded the memory of his 512 and finds that FRACINT (version 15) now seems to run quite well. Previously it simply reported insufficient memory.

Dr. Sutton says he's investigated only a part of this complex program so far but, he says enough of it works (within the limits of the CGA display, of course) to be interesting.

FRACINT is shareware, but I have no further information on it. Even so, if you're interested in fractals and have enough memory you might like to give it a try. B

1st Course - Graphics (4)

Alan Wrigley concludes his introduction to graphics programming.

COLOUR REVISITED

In previous articles in this series, I have described the use of GCOL to set the current foreground and background colours to be used with graphics plotting. So far in our examples we have used the simplest plotting action (0), which is to plot each pixel in the current foreground colour, overwriting whatever was there already. As I indicated, however, there is a range of other plotting actions (numbers 1-3) which make use of the boolean operators (OR, AND or EOR). The boolean operation is performed between the existing colour of the pixel and the colour specified in the GCOL command, and the resulting colour is plotted. This opens up a whole range of possibilities for animation.

Of the three boolean plotting actions, the most widely used is EOR plotting (plotting action 3). I will therefore concentrate on this in this article; however, you may like to explore OR and AND plotting yourself to see their effect.

EXCLUSIVE-OR PLOTTING

One of the properties of the EOR action which makes it particularly useful for graphics is as follows: given any two numbers together with their EOR product, then EORing any two of these three numbers together produces the third. For example, 2 EOR 1=3, 2 EOR 3=1, 1 EOR 3=2. This means that if you plot an object moving across the screen using EOR plotting, then you can erase it from its previous position with a repeat of the same plot that put it there, since the EOR action will reverse the colours back to their previous values.

Often this will just be a case of moving across a black background. For example, if your background is black (colour 0), and you plot an object using *GCOL 3,c* where c is the colour of the object, then the first plot will display the object (c EOR 0=c), while the second plot will remove it (c EOR c=0). However, the technique can also be used to move one object behind another.

Taking a 4-colour mode as an example (say mode 5), you might be using three actual colours in your animation - one for the background and one each for the static and the moving objects. What you must do is to map these colours to logical colours 0-2 (background=0, static=1, moving=2). So if, for example, you want static blue objects, moving red objects and a black background, you would use:

```
VDU 19,1,4,0,0,0
VDU 19,2,1,0,0,0
```

to set the blue and red to logical colours 1 and 2 (assuming colour 0 is already black). Now map the physical colour of the static object additionally onto logical colour 3:

```
VDU 19,3,4,0,0,0
```

so that you now have two logical colours linked to the same physical colour.

To illustrate how this works, we will amend our bouncing ball program from last month. The ball itself will now be red, while in the centre of the screen there will be a static blue triangle. If you run the following listing, you will see that where the path of the ball crosses the triangle, it appears to move behind it and then reappear on the other side.

```
   10 MODE1
   20 VDU23,240,0,0,7,31,63,127,255,255
   30 VDU23,241,0,0,224,248,252,254,25 5,255
   40 VDU23,242,255,255,127,63,31,7,0,0
   50 VDU23,243,255,255,254,252,248,2 24,0,0
   60 a$=CHR$240+CHR$241+CHR$10+CHR$8+CHR$8
+ CHR$242+CHR$243
   70 VDU5:step%=8:VDU19,1,4,0,0,0
   72 VDU19,2,1,0,0,0:VDU19,3,4,0,0,0
   74 GCOL0,1:MOVE 512,700
   76 MOVE 300,400:PLOT 85,724,400
   80 xs%=step%:ys%=-step%
   90 x%=400:y%=400:GCOL3,2
  100 REPEAT
  110 MOVE x%,y%:PRINTa$
  120 *FX19
  130 MOVE x%,y%:PRINTa$
  140 x%=x%+xs%:y%=y%+ys%
  150 IF x%>1248 x%=1248:xs%=-xs%
  160 IF x%<0 x%=0:xs%=-xs%
  170 IF y%>1024 y%=1024:ys%=-ys%
  180 IF y%<64 y%=64:ys%=-ys%
  190 UNTIL FALSE
```

The new program differs very little from the one given last month. Firstly, mode 1 is used instead of mode 0, in order to give 4 colours. Next, the colour mapping is done in lines 70-72 as described above. The triangle is plotted at lines 74-76, and finally the colour is set in line 90 using GCOL 3,2 this time. These are the only alterations from last month.

The effect of using EOR plotting is as follows: when the ball crosses the background, the first plot at line 110 EORs the existing colour (0) with the ball colour (2) to produce 2, displaying the ball in red. The second plot at line 130 does exactly the reverse (since the existing colour is now 2), thus erasing the ball from its present position before moving it on. When the ball crosses the triangle, the existing colour is now 1, which when EORed with 2 becomes 3. We have already redefined colour 3 to be the same as the static object, and so the ball will seem to disappear behind the triangle.

This is a very simple example, using only 4 colours, but the principle can be applied equally well to 16-colour modes. With some careful planning to work out which colour combinations will give the correct result when EORed together, you can make multi-coloured objects disappear behind others. This technique is widely used by games programmers to achieve realistic animation.

FLOOD FILL

Finally in this series on graphics programming, I want to look at the problem of filling a shape with colour. As I described last month, most straight line shapes can be made up from filled triangles, but for some complex shapes the only way to achieve a solid object is to use flood fill after the object has been drawn. There are flood fill plot codes available on the Master (for details of these see the Reference Manual), and on the model B there is a group of codes (72-79) which will plot a horizontal line left and right until a non-background colour is reached, though only in OS 1.2.

It is not difficult to devise a general-purpose flood fill routine, but it does have the disadvantage of being time-consuming. Wherever possible, it is better to design objects that can be plotted with filled triangles (or rectangles or circles on the Master), but if you *do* have to resort to flood fill, there is a fairly straightforward way to achieve it. You must bear in mind, though, that there are limitations to what such a routine can do, as you will see later.

First of all, you have to specify a start point somewhere within the shape you want to fill. The exact location of this will depend on the nature of the shape, as I will describe in a moment, and may or may not be critical. At the same time, the routine has to be clear about which colour should be recognised as the boundary of the shape; in other words, where does the fill stop. Next, you work outwards from this point in both directions horizontally until you meet the boundary colour on each side, and draw a line in the fill colour between these two points. Then you repeat the process line by line above and below the start point, until you reach the boundary colours at the top and bottom.

Now you can begin to see the limitations. If the shape is very complex, the fill routine could end prematurely in any direction if it hits a boundary pixel which forms part of an indentation. For the best results, the start point should lie vertically on the line of the tallest part of the shape. In some cases, however, you may need to call the fill routine a second or even third time from different start points to fill in odd corners of the shape.

The following procedure will fill a shape in the manner described, and takes five parameters: the x and y co-ordinates of the start point, the fill colour, the boundary colour, and the number of columns in the current mode (i.e. 20, 40 or 80).

```
1000 DEFPROCfill(x%,y%,fc%,bc%,cols%)
1010 pix%=160/cols%
1020 GCOL 0,fc%
1030 MOVE x%,y%:x1%=x%:y1%=y%
1040 FOR i%=-4 TO 4 STEP 8
1050 REPEAT
1060 REPEAT x1%=x1%-pix%
1070 p%=POINT(x1%,y1%)
1080 UNTIL p%=bc% OR p%=-1
1090 MOVE x1%,y1%:x1%=x%
1100 REPEAT x1%=x1%+pix%
1110 p%=POINT(x1%,y1%)
1120 UNTIL p%=bc% OR p%=-1
1130 DRAW x1%,y1%
1140 x1%=x%:y1%=y1%+i%
1150 p%=POINT(x1%,y1%)
1160 UNTIL p%=bc% OR p%=-1
1170 x1%=x%:y1%=y%
1180 NEXT
1190 ENDPROC
```

The routine works as follows: firstly, *pix%* in line 1010 is set to the number of OS units per pixel in the current mode. This ensures the maximum speed when scanning left and right to find the boundary. The graphics cursor is then placed at the start point (line 1030). In all graphics modes there are 4 OS units per pixel, and so the step size for our vertical movement loop is set to 4 (line 1040). The routine moves left (1060-1080) and then right (1100-1120), and draws a line between the two end points (1130). Line 1140 moves up and then down from the start point to repeat the process, until a boundary pixel is detected in the vertical axis (line 1160). Note that we must detect if the shape goes off the screen, otherwise the routine will hang while searching for the boundary colour. Thus our repeat loops look for both a boundary colour pixel (p%=bc%) and a value of -1 (off screen).

The ideas and routines presented in this and the previous three articles should enable you to liven up your graphics. Next month, *1st Course* moves on to another topic.

Public Domain Software

This month, Alan Blundell goes to some trouble to acquire the oldest public domain software in existence.

Before I launch into the main subject for this issue, I should give the latest news about the 'Spriter' story which I related in the October issue.

I was pleased and surprised by the feedback readers gave on the 'Spriter' topic, following my brief request to Master 128 users in a recent column - thanks again to one and all! I think that the reason for the request was fairly well covered in the October issue, but there have been a couple more developments which are important to relate. The first is that, having written that column close to the editorial deadline, I based the comments on the responses which the early birds amongst you sent in. There were plenty of you, but over the next few weeks I received even more information. From this, it was obvious that not all Master 128 Welcome discs do contain 'Spriter'. About 1% of the responses were from people who did not have a copy of the file. There was no apparent pattern to them, being connected with computers bought anytime from 1986 to 1990, so I was lost for a rational explanation.

Fortunately, a response to my latest letter to Acorn solved the question (to Acorn Customer Services, I'll stop pestering you now, if you're reading this!). Briefly, the situation is that 'Spriter' has been present on all Master 128 'Welcome' discs manufactured since sometime in 1986; the exact date is unknown to Acorn. Discs manufactured before that date do not contain the file, discs made later do. The Master bought in 1990 which was supplied without the file was obviously old stock!

Hopefully, that ends the 'Spriter' saga. There is one more thing to say, though. Many of the responses to my plea for information asked if it would be possible for an article explaining the use of 'Spriter' to appear in a future issue of BEEBUG. Well, work is now in hand and hopefully information to help readers who are interested in making use of sprites in their own programs is not too far away.

THE BIBLE

I'm not really sure how the term 'software' should be defined; my own understanding used to be that it was the stuff which made the hardware work but the word seems to have been extended in use to mean anything which can be recorded on computer storage media (i.e. discs or tape). If this latter definition is acceptable, then I have recently come across the oldest item of computer software in the world.

I recently had a request for a copy of the Bible on disc, from someone who was unable to read the book for himself, so that he could 'read' it via a speech synthesiser attached to his Master. I had no knowledge of such an item being available and didn't fancy the prospect of quickly tapping in a few million words, so I reluctantly said that I couldn't help. A short time later, I received in the post a set of 6 MS-DOS discs containing the full text of the King James version of the Bible (both old and new Testaments), together with an index, an efficient search and text retrieval program, and a lengthy text file which constituted a full manual for the software. The whole package is available as public domain software for PCs.

My first reaction was to see if the package would work on a Master 512 co-processor. The short story is that I don't think it does, but the long story is as follows. The 6 DOS discs contained a set of compressed archive files, which first had to be extracted from their archives. The archiving utility used was PKZIP, a utility which should be familiar to all Master 512 users. PKZIP is, at least in my opinion, the best of a range of utilities for DOS which efficiently compresses and combines any set of files into a single 'archive' file. The result is a massive saving on disc space, often around 60%-70%, together with a tidy means of distributing a package (only one file to copy instead of possibly dozens).

PKZIP is shareware and is fully compatible with the Master 512, using DOS Plus 2.1 and a second copy of COMMAND.COM. It is surprising if no similar PD or shareware utility has been produced for the BBC; apart from the benefits of compression, it would be an ideal means of overcoming the 31 file limit for DFS users. I am aware of two such utilities for the BBC which are 'going the rounds' but have never been able to confirm that either is actually public domain. If anyone can help, I would be pleased to hear from them.

Back to the Bible. PKZIP was actually only the first hurdle to overcome; once the archives had been restored (with PKUNZIP), I was left with a set of installation files which were themselves compressed, this time using LHarc, another DOS archiving utility. This in itself would not have been a problem (LHarc also works on the 512), but the arcane installation program actually made it impossible simply to unarchive those files and be left with a usable set of programs and data. The installation program itself is not 512 compatible, producing a range of unusual error messages. The reason I am not fully certain that the package does not work on the 512 is that, if these installation problems can be worked around, the package itself may actually work.

Having made a mental note to have another go at that sometime, I changed tactics and installed the package on a 'real' PC's hard disc. Having done that, I found that the package itself was quite impressive in its features and speed of searching for specific passages of text. However, here came the next problem; the Bible itself was stored in yet another compressed form, this time keyword coded with an index file of the relevant words, so it took some time to find a way of extracting the full text. I eventually managed to get there (or else there would have been no point in telling the story) with a single file about 5Mb long. The Bible is quite long, but obviously such a file was no practical use on your average BBC Micro!

The final result of my efforts was a set of 8 ADFS L-format discs containing the Bible separated into the individual Books, from Genesis to Revelations. A few of these are split again into more manageable files; the half megabyte of Psalms is split into three files. The most impressive part of the whole thing was the fact that all of this text, together with a search program, a 150K manual, a comprehensive index and cross-reference information was all originally squeezed onto just 6 360K DOS discs!

That was probably (no, hopefully) the biggest conversion job I have ever undertaken. The final part is still not done: if anyone fancies an ambitious programming challenge which could be put to good use, how about producing a search and text retrieval program to work on these text files now that they are available on the BBC? The BBC Micro has a number of blind users who benefit from the use of a speech synthesiser. I know of one person who would make use of such a program, and there may be many more.

NEXT ISSUE

In the next issue, I will be able to give the latest news about my campaign to get ex-commercial BBC software re-released as PD or shareware. All I need now is a more catchy name for the campaign...

Using Sideways RAM (Part 2)

Mr Toad shows you how to score with headers.

This month we bring you the first of the three promised sideways ROM headers. Type the listing in and give it a useful name (or why not buy the magazine disc?) and keep it until you next need a bit of machine code for some little job. Then you can haul it out, dust it off and tack your routine on the end. Last month we went into some of the advantages of using sideways RAM as machine code storage; but I'll just repeat the simple points that sideways RAM is safe from corruption, doesn't impinge on precious main memory, can be called from a Basic program by a star-command (or several such), and can as easily be called from any other environment. The only drawback is the complicated header code and general 'framework', which we've now done for you.

As we said last time, this one is a small version designed to respond to only one star-command. This is currently *ABC (line 600), but of course you can alter that to a nice mnemonic command for the job in hand. You will probably also want to alter the title (320), the copyright string (340) and the *HELP text (580). You insert your own routine from 630 onwards, the present line 630 being a test function. When you've typed the listing in and run it, doing *ABC (and only *ABC) should produce a dollar sign and a beep. You must finish up with a claim-and-exit (as in line 640) then a closing bracket, a NEXT and an ENDPROC, as in line 650. If the assembled code is long you might just need to change line 230 to allow more space for this, though the present DIM of &500 bytes is pretty generous: the present listing takes up about 10% of that.

"This is a *small* one?", I hear you croak, "sixty-two lines and it doesn't do anything useful yet - what does a big one look like, for Gawd's sake?" Well you don't have to type in all sixty-two lines: you can do without a *HELP facility, for a start, and you may not need all the labels for all the MOS calls (lines 180 - 220). In your place, though, I would put it all in. The whole point of this program is that it should cover contingencies which you haven't yet thought of, and a bit of typing now may well save work later on, when you want to concentrate on your working routine and not faff about with the nuts and bolts. A *HELP facility is always useful, because it only works if the ROM is initialised, whereas *ROMS will show it as long as it's in a slot at all; this is a great aid to debugging. In any case, you ought to show your ROM's existence on *HELP, otherwise things can get confusing. If you do omit the *HELP part (lines 410 - 450) you can also skip the help text (570 - 580) but you must include the actual command at 590 - 600, as it is also used by the interpreter. Failure to type in lines 10 - 50 will incur the wrath of Mr Williams and bring down on you the curse of the mummified reptile-god Kopiritis.

THERE IS A THEORY.....

Let's now take a look at the theory of the ROM header, using this as an example, and then we'll discuss the support code and the Basic.

To begin at the beginning: line 270, BRK:BRK:BRK. That's really three zeros; typing BRK is easier than EQUB 0. The first three bytes of a ROM are the 'language entry point': if your ROM is a language and it gets one of the usual A=4 calls to &8003 which turns out to be the command to select itself as current language, it will issue OSBYTE &8E, which does some internal housekeeping and re-enters the ROM at &8000, the code there being a JMP to the entry proper. In a non-language ROM, Acorn tells us to put three zeros there, the point presumably being that an erroneous JMP to a zero would cause a break, which is better than a hang-up. As is often the case, Acorn doesn't follow its own advice in this respect; if you reserve an SRAM slot with *SRDATA or *SRROM, the MOS puts an RTS at &8000. What &8001-2 contain is irrelevant, naturally. Three NOPs would result in a jump to the service entry point; it doesn't much matter, since an attempt to select a service ROM as a language is a foul-up in any event.

Line 280, JMP checkCalls. With all this header code of indeterminate length, the MOS needs a set way to get to where the real code begins, so Acorn specifies that &8003-5 contain a jump to the start proper.

Line 290, EQUB &82. This is the type byte; for full details see my article in BEEBUG Vol 9 No 5. On initialisation, the ROM copies the contents of this location, &8006, to the table at &2A1. As regards the number, &82 has bit 7 set, and that tells the MOS that the ROM has a service entry point at &8003. In fact, all ROMs have, except Basic which responds to no star-commands - the command *BASIC is dealt with by the MOS - but an SRAM slot might well be reserved by *SRDATA or even be in use by the ROM filing system, and the MOS doesn't want to go offering commands to those. If their type bytes don't have bit 7 set, it won't.

Line 300, copyright MOD &100. As I said last month, on power-up or a hard break the MOS checks all the ROM slots to see if there's anything in them, and only if there is a proper ROM in the slot does it put that ROM's type byte in the ROM table at &2A1. This avoids crashes which would be caused if the MOS, when issuing a call to all ROMS, did a JSR to something invalid. It's not as simple as checking whether there's any old non-zero byte at &8003: we need to be sure that it's not just random junk. Auntie Acorn, when designing a check routine, cunningly killed two birds with one stone here. The MOS reads this location, the eighth in the slot, adds the number contained therein to &8000 and checks the four locations beginning at that address (&8000+contents of &8007) for the sequence zero, opening bracket, capital C, closing bracket. This sequence is extremely unlikely to be in exactly that place by chance, so its presence is close to a guarantee that there's a ROM in the slot. If the ROM turns out to be a mass of bugs, that's your problem. At the same time, of course, the copyright message also performs its more obvious, legal function. It's worth knowing that you do need a zero plus the string "(C)", but that's all you need for technical purposes.

Line 310, EQUB &93. This is the 'binary version number'. In response to *ROMS it is printed after the title as a two digit hex number: 1 comes out as 01, 32 as 20 and so on. It's meant to let you see the version of a given ROM present in the machine, but it can't cope with things like '2.3', and to me it looks untidy. Programmers could have put the version number as part of the title string. Anyhow, there's no way you can stop the byte at &8008 being printed, so I always put in the year in hex.

Line 320. On a *ROMS call the MOS prints as a 'title' the sequence from &8009 to the next zero, taken as a string - the zero is the BRK in line 340, which also serves as part of the copyright sequence. Then it prints a space, then the byte in &8008, the 'binary version number'. After the title and the zero you may put an optional 'ASCII Version string', followed by another zero, the main reason for which is that you can use it in your *HELP routine. Many ROMs print the title on *HELP, but by the time you've added a couple of OSNEWL calls onto the print routine, it's as easy to use a separate string, and you get a choice of wording. Then we come to the copyright string, discussed above.

There's nothing after that which concerns service ROMS. In the case of a language, if you put a zero after the copyright string, the next three bytes will be taken as a relocation address on the far side of the tube. If the type byte has bit 5 set and if a co-processor is operating, then the MOS will copy the ROM across the tube to this address as part of the initialisation process.

Next I have put the service entry proper, lines 360 onwards. It doesn't have to be anywhere in particular, but this is as good a place as any.

Now we said last month that, except in the event that you wish your ROM to grab workspace and maybe a vector on initialisation, there is no need for a Ctrl-Break to initialise a ROM image in SRAM, all you do is to insert the type byte into the table at the time you load the ROM. On the Compact, you just put the parameter 'I' after the *SRLOAD or *SRWRITE and it's done for you. It follows that a ROM can function perfectly well without the header code, so long as you don't care if it's de-initialised by a hard Break. You don't even need a command interpreter if you can accept that any star-command which gets down to the SRAM slots unclaimed will call your code.

What happens after your ROM has done its stuff will depend on how you code the exit. Now this is worth knowing, because if you need a temporary bit of code for some specific job, and don't have a standard header about your person, just write BRK:BRK:BRK or begin with P%=&8003, then start your code. You still need the *SRWRITE and the initialisation, and to stop it answering all the different types of call it is best to begin with:

```
    CMP #4:BEQ go:RTS
    .go
```

but that's all you need to get a working ROM.

The rest of the code in this listing is pretty straightforward. At *.checkCalls* we save the registers and test the accumulator: A=4 causes a branch to the interpreter, and if the test for A=9 fails we branch to a 'no claim' exit. If A does hold 9, we get to the same point via the *HELP routine - you mustn't claim a call such as *HELP which is for all ROMs. The interpreter simply checks that the command string at (&F2),Y matches our command held in the *HELP data, skipping the two initial spaces as described above. Any mismatch causes a branch to the no-claim exit, and if we get as far as the zero in our string then it all matches, including the final &0D, and we branch to .go where you begin your working routine. Note that the star-command doesn't have to have three characters like the specimen, any length is catered for. In fact, for temporary purposes you can use a simpler method:

```
      LDA (&F2),Y:AND #&DF
      CMP #ASC "Q":BNE noClaim
```

will give you a response to *Q, and you can cater for a longer command by duplicating the above. Add:

```
      INY:LDA (&F2),Y:AND #&DF
      CMP #ASC "U":BNE noClaim
```

and it now responds to *QU, and so on. You'll see a useful variant of this technique in the medium-length header next time.

The Basic in lines 110 to 150 is almost self-explanatory; the loop looks for a free slot in the table; only if all four slots are non-zero does it pass the first NEXT with N%=3 and end with the 'full up' message. Otherwise it *SRWRITES the code to the first free slot found, sticks the type byte into N%?&2A1 and clears the NEXT stack (as the routine amounts to jumping out of a loop). This is very simple stuff, but you'd be amazed at the competent programmers who don't do it. There was an otherwise excellent ROM in BEEBUG in 1992 which actually worked out the text of the necessary *SRWRITE and printed it for you to type in! Sorry, but I'm far too lazy to type in anything that the computer can do instead.

A DRAWBACK

This method does have one drawback, however: it can be a nuisance during development work. You run the assembler and it puts the ROM in slot 7. After a trial, you make some changes and RUN the program again. This time, Basic finds that slot 7 is occupied and puts the new version into slot 6. When you try it out, the earlier version in slot 7 responds and claims the call. You think your alteration was ineffective and have another go: version 3 ends up in slot 5. You have a breakdown. To get round this, I always run the program via one of the function keys, which does:

```
      ?&2A8=0|M RUN|M
```

or you can insert a temporary line:

```
      105 ?&2A8=0
```

To clear an SRAM slot - say, slot 6 - fill it with junk and remove it from the table:

```
      *SRWRITE 7000+100 8000 6
      ?&2A7=0
```

While we're on the subject, one very silly and unnecessary drawback to the Beeb system is that there is no way of assembling SRAM code directly into sideways RAM; you have first to assemble it to main memory - hence the DIM at the start - and then *SRWRITE it to an SRAM slot. This waste of space becomes a real pain when the object-code gets to somewhere around 5K; in a later issue we'll look at ways round the problem.

That's it for now. Next month we bring you the second header.

```
   10 REM Program Small ROM Header code
   20 REM Version B 1.0
   30 REM Author  David Holton
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
  100 PROCassem
  110 FOR N%=7 TO 4 STEP-1
  120 IF N%?&2A1 NEXT:PRINT''"No free SR
AM slot."':END
  130 OSCLI "SRWRITE "+STR$~Z%+" "+STR$~
(O%+1)+" 8000 "+STR$ N%
  140 PRINT''"READY IN SLOT ";N%
  150 N%?&2A1=&82:N%=4:NEXT:END
  160 :::::::::::::::::::::
 1000 DEF PROCassem
```

```
1010 osasci=&FFE3
1020 osnewl=&FFE7
1030 oswrch=&FFEE
1040 osword=&FFF1
1050 osbyte=&FFF4
1060 DIM Z% (&500)
1070 FOR N%=4 TO 6 STEP 2
1080 P%=&8000:O%=Z%
1090 [ OPT N%
1100 BRK:BRK:BRK
1110 JMP checkCalls
1120 EQUB &82
1130 EQUB copyright MOD &100
1140 EQUB &93
1150 EQUS "My Very Own Sideways ROM"
1160 .copyright
1170 BRK:EQUS"(C) me 1992"
1180 :
1190 .checkCalls
1200 PHA:PHX:PHY
1210 CMP #4:BEQ isItOurs
1220 CMP #9:BNE noClaim
1230 :
1240 LDX #&FF
1250 .helpLoop
1260 INX:LDA helpText,X:PHP:JSR osasci
1270 PLP:BNE helpLoop
1280 :
1290 .noClaim
1300 PLY:PLX:PLA:RTS
1310 :
1320 .isItOurs
1330 LDX #0
1340 .checkLoop
1350 LDA starCommand,X:BEQ go
1360 LDA (&F2),Y:AND #&DF
1370 CMP starCommand,X:BNE noClaim
1380 INX:INY:JMP checkLoop
1390 :
1400 .helpText
1410 EQUW &0D0D: EQUS "My...Rom - Type
*"
1420 .starCommand
1430 EQUS"ABC":EQUB &0D:BRK
1440 ::::::::::::::::::::::::::
1450 .go
1460 LDA #ASC"$":JSR oswrch:LDA #7:JSR
oswrch \ test routine
1470 PLY:PLX:PLA:LDA #0:RTS
1480 ]:NEXT:ENDPROC
```

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

DESIGNING NLQ CHARACTERS (Vol.11 Nos.4, 5 & 6)

The author of this series, Elaine Kemp, has written to add a reminder regarding part 3 - that you need to have a buffer in your printer for NLQ characters to be downloaded. Printer buffers sometimes come as standard, sometimes they are an optional extra.

THE COOL CURSOR (Vol.11 No.6)

A small error in the listing prevents the cool cursor from being cool when other interrupts are occurring. To fix this, change line 360 to:

```
360 LDA &FE4D:AND #2:BEQ quit
```

MR. TOAD'S MACHINE CODE CORNER (Vol.11 No.6)

The program Random will not run on a BBC B as it stands. You must change the following for the model B:

```
110 P%=&A00
170 LDA (&8E),Y
```

NUMERICAL FORTUNES (Vol.11 No.6 Disc Only)

There is a small problem if your date adds up to 10 - the 10 is not reduced to 1. To correct this, change line 1510 to:

```
1510 REPEAT:number$=STR$(number%)
```

and add to the end of line 1540:

```
:UNTIL LEN(STR$(number%))=1
```

Mr Toad's Machine Code Corner

This month Mr Toad goes backwardly compatible and finds that there's no law against it.

Greetings, milk-munchers. The other week, Mr T was reminded just how old a computer the BBC B is: re-reading King Alfred's famous letter on the state of learning in England in the 9th century, I pondered over the celebrated passage "swithe feawe waeron buton Beebug, the hiora listingas cuthen understondan on 6502, ond ic waene, thaette noht monige aet Beebug naeren." How true those words are, even today.

You will all be aware that a lot of software written for the Master, which came out several centuries later in the reign of Aethelraed Amstraed, will not run on the original steam-powered Beeb. There are many reasons why - 'illegal' programming, use of Basic 4 instructions like ON PROC, shortage of memory, gassy coal et quid habes.

If the program is in assembler, the difficulty is often the use of the new instructions that work on the 65C02 processor in the Master, but not on the earlier 6502. Quite often a rewrite of the assembler to replace the "naughty" instructions with 6502 code will give you a useful program for very little work. You do need the source code, though; altering any large amount of object-code is a formidable task, to put it mildly.

Even if you know all the naughty instructions, it is often a pain to winkle them all out. You can keep running the assembler and using the error messages to find the next naughty, but for personal use Mr T has written a Sideways ROM image which will run on any model with SRAM and gives a neat printout of any occurrences of the new instructions together with the numbers of the offending lines. You'll find this on this month's subscription disc as *MBLAST*.

What really would be useful would be a program which could work automatically on the object code instead of the assembler, but alas, it could only be used after establishing which areas are data and which are code. In fact, writing such a program would never repay the immense labour, assuming the task turned out to be possible.

What we're going to do here is try to identify fixes for these problems which are of fairly general application and can be used without being fully understood. For an experienced programmer there will be many cases where the obvious solution will be quite different to the present suggestions. There may be a few readers who, having never used a Master, don't know the meaning of the new mnemonics, so at the risk of boring the rest of you to death we'll do Mr T's mug's guide as we discuss tactics.

There are about 20 "naughty" instructions (depending on what you define as a separate instruction), and for only one is there a single fix guaranteed to work in all cases: the unconditional BRAnch can always be replaced by JMP, provided there is room for an extra byte. If only they were all that easy. I don't know why anyone would use BRA, anyway, unless they were literally down to their last few bytes of memory, because it's actually slower than JMP if a page boundary is crossed; if you alter your code you may go out of range, and you're cutting yourself off from the older computer - all for the sake of one byte.

CLR and STZ: alternative mnemonics for 'store zero in the specified address'. This is best replaced by setting one of A, X or Y to zero and writing it to the address. There are other ways, but all involve a register. STZ... becomes LDA #0:STA... or LDX #0:STX... or LDY #0:STY... If none of these work, try PHA:LDA #0:STA...:PLA. There is another, unofficial answer: on many 6502 chips STZ exists as an 'undocumented instruction' - try EQUB &9C:EQUW address. If it works on your machine, then fine, but it won't work on all, and it's only good for one sort of STZ instruction, that with absolute addressing.

DEA, DECA and DEC A all mean 'decrement acumulator' (how on earth did it get left off the earlier chip?). First, if X or Y are free, do TAX:DEX:TXA or TAY:DEY:TYA. Failing that, you'll have to copy A to an address in RAM, decrement that address then copy it back into A - STA storeA:DEC storeA:LDA storeA. Clumsy, but not as bad as PHX:TAX:DEX:TXA:PLX.

INA, INCA and INC A all mean 'increment accumulator'. Treat these like DEA - TAY:INY:TYA or TAX:INX:TXA, else use RAM: STA storeA:INC storeA:LDA storeA.

PHX, PHY, PLX and PLY are nasty. PHX/PHY stand for Push X or Y, just like PHA on the old chip. Since the 6502's accumulator can be pushed, the usual fix is TXA:PHA or TYA:PHA to push and PLA:TAX or PLA:TAY to pull. The problem for those used to pushing only the accumulator is keeping track of the order of the pushes. Here's a common example.

A COMMON EXAMPLE

At various points in a Sideways ROM it is usual to push or pull all three registers at once. If the command PHA comes *after* the PHX and PHY, you'll have to alter the running-order so as to push A first, else its contents will be lost when you do TXA:PHA or TYA:PHA. In other words, the sequence TXA:PHA:TYA:PHA:PHA makes the last PHA meaningless, since you have lost the value contained in A before the sequence began. Alter to PHA:TXA:PHA:TYA:PHA. Y is still preserved for LDA (&F2),Y, the start of the command interpreting routine. When it comes to pulling the values back off the stack, remember that the *last* value pushed onto the stack is pulled off *first*. To restore all three registers if they were pushed by the last-mentioned sequence, therefore, you need PLA:TAY:PLA:TAX:PLA. This whole area is fraught with danger since, of course, the stack is also used by JSR/RTS, and it must never become *unbalanced*, which is to say that every push made after a JSR must be balanced by a pull before an RTS, even if the programmer no longer needs the number stored on top of the stack. Errors in this area often lead to interesting crashes.

Don't forget that you can always store a register in RAM instead of pushing it. This is about as fast, safer and means that you can dispense with pull instructions put in only to balance the stack. The bad news is that if there is a second push before the next pull, you have to use a second store. You also need to work out which pull matches which push - after a branch this is not always obvious. Note, too, that you can push one register and pull the contents into another; PHY:PLX is therefore the same as the nonexistent TYX, if you ever need such an instruction.

TRB and TSB are quite rarely used, as far as Mr T has observed. TRB ANDs the complement of A with the memory location, setting the Z flag on zero; TSB (the command that likes to say yes) ORs A with the memory location and sets or resets the Z flag. TRB address should succumb to EOR #&FF:AND address:STA address, though you may need to save A first, of course. TSB address should be the same as ORA address:STA address. This fix won't necessarily work if the next bit of code tests a flag after the TRB/TSB: in that case some CMP instruction would be needed at some point, which is a blind-'em-with-science way of saying you're own your own, Buster.

There are eight instructions which are only naughty if followed by a zero page

continued on page 55

All Together Now!

Chris Robbins pulls together BEEBUG's IBM transfer programs.

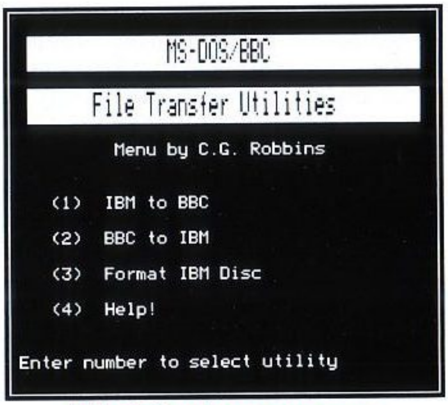
Due to the number and size of the programs referred to in this article they are provided only on this month's subscription disc.

Over the last few years BEEBUG has published some very useful programs that make it possible both to format IBM PC discs and to transfer disc files between the BBC series computers and an IBM PC. Updated versions of these programs were published just over a year ago (BEEBUG Vol.10 No.4). More recently the Postbag column showed how they could be used with that ubiquitous filing system for the Beeb, Watford Electronics' DDFS.

Despite their usefulness and the modifications suggested by P.J.Lawrence in Postbag, using the formatting and transfer facilities can prove a mite fraught, especially if, like me, you use them infrequently. So, I thought it was about time that everything was brought together under one programming roof, so to speak, and made easier to handle.

THE MENU PROGRAM

Although some modifications have been made to the BBC/IBM programs, most importantly to produce versions that work with Watford's DDFS, large scale changes have been avoided by building a simple 'front end' menu program - *IBMMENU*. This provides options for selecting and running the various utilities, and setting up the appropriate BBC drive number and directory. There's also a simple HELP feature that, hopefully, will tell you all that you want to know about controlling the formatting and transfer facilities.



```
MS-DOS/BBC
File Transfer Utilities
Menu by C.G. Robbins

(1)  IBM to BBC
(2)  BBC to IBM
(3)  Format IBM Disc
(4)  Help!

Enter number to select utility
```

IBM - BBC main program menu

BBC DRIVE/DIRECTORY SELECTION

The operation of the menu program is simplicity itself, so I'll omit any detailed explanation. However, it's probably worth mentioning the BBC disc drive/directory selection feature. It's an additional request for information displayed by the program when selecting either the IBM to BBC or BBC to IBM transfer facilities. It not only allows you to select in advance the appropriate drive and directory from or to which you wish to transfer BBC files, but also helps prevent the wrong drive being selected. The default drives/directory used for the BBC format files are : BBC to IBM - 1/$, IBM to BBC - 0/$

HELP FEATURE

This is implemented via a separate simple control program, *HELP*, chained by the menu program, that allows you to page through a collection of notes concerning the BBC/IBM utilities. Simple it maybe, but it comes in very handy for infrequent users such as myself.

I've deliberately avoided tailoring it exclusively for the transfer utilities, so that it can, if required be 'lifted' and used for any other application you may have in mind. The REMs at the head of the program can easily be changed or deleted as required - they have no bearing on program operation.

HELP NOTES

In order to make the HELP feature as general purpose as possible, the associated HELP notes are held in an ASCII format file i.e. a spooled text file called *NOTES*. Alternative files can easily be prepared with any suitable word processing package capable of producing ASCII files. This avoids having to change the HELP program itself, except perhaps to change the file name *NOTES*, if you so choose.

UTILITIES DISC

To make things even easier, and just as importantly, to minimize possible 'finger problems' when using the utilities, I suggest that like me you keep all the relevant files on one disc reserved solely for that purpose, set up a tiny !BOOT file to chain the menu program, and set the disc option to *EXEC this when Shift-Break is pressed.

The files you'll need are :-

!BOOT	**- EXEC file to CHAIN "$.MENU"**
MENU	**- 'front end' menu program**
HELP	**- HELP program to display $.NOTES**
NOTES	**- ASCII format notes**
IBM-BBC	**- IBM to BBC file transfer program**
BBC-IBM	**- BBC to IBM file transfer program**
PCFORM	**- IBM disc formatting program**

Once the utilities disc has been set up, subsequent operation merely involves putting it in drive 0/2 and pressing Shift-Break to bring up the menu.

WATFORD DDFS

The original transfer/formatting programs, although requiring the 1770 disc controller, did not work with one of the most popular disc filing systems for the Beeb, Watford Electronics Double Density Disc Filing System (DDFS), which also used that controller as the basis of its 8271 emulator. However, the modifications suggested by P.J.Lawrence have got round that problem, albeit imposing limitations on drive selection.

Both the menu program and the HELP notes have been tailored to fit an 80 track dual drive system using Watford's DDFS, and incorporating Mr. Lawrence's modifications. However, there is nothing to prevent them being modified for other combinations.

I have also made some slight modifications to the latest versions of the transfer/formatting programs as published on the magazine disc for BEEBUG Vol.10 No.4 August/September 1991. The modifications include 'fixes' to allow Watford's DDFS use (see Postbag, BEEBUG Vol.10 No.10), true dual drive operation, together with amended user messages to make the programs more compatible, and to minimize 'finger problems'.

Modifications are as follows:

IBM PC Disc Formatting - file PCFORM

```
  255 ENDPROC:REM Disables drive
selection
  760 IF M%=251 OR M%<1 cmd%=&FE84:
ctrl%=&FE80: M%=FALSE: fc%(0)=32:
fc%(1)=34
  930 PRINT'CHR$136"   Ensure disc to be
formatted"
  925 PRINT CHR$136"   is in drive 0/2"
```

IBM to BBC file transfer - file IBM-BBC

```
  140 D=0
 1020 PRINTT$'T$'LEFT$(T$,2)+CHR$130+"<4
spaces>(Drive 1/3 to drive 0/2)"
 1510 val=rst
 1680 wd=&FE84:ctrl=&FE80:sel=2:dden=8:
rst=20
 1710 ?ctrl=rst:ENDPROC
 1860 REM deleted
 3270 PRINT TAB(0,14)"Remove IBM/BBC
file transfer utilities disc. Then put
the IBM disc in drive 1/3and the BBC
disc in drive 0/2."
```

BBC to IBM file transfer - file BBC-IBM

```
    0 ON ERROR GOTO 610
   55 REM Basic IBM file extension
   56 E$="AA"
  160 D=0
 1240 PRINTT$'T$'LEFT$(T$,2)+CHR$130+"<4
spaces>(Drive 1/3 to drive
0/2)"'CHR$129;CHR$157
 1390 ch=ASC"A":REPEAT
 1400 new$=FNuc(name$)+"."+E$+CHR$ch:f=0
 1460 $(dir+ifn*32)=FNuc(name$)+
E$+CHR$ch
 1960 wd=&FE84:ctrl=&FE80:sel=2:dden=8:
rst=&20
 1990 ?ctrl=rst:ENDPROC
 2120 REM deleted
32703 PRINT TAB(7,21)CHR$130;"Press any
key to continue"
32704 PRINT TAB(10,22)CHR$130;"or
Escape to exit."
32705 PRINTTAB(0,15)"Remove IBM/BBC file
transfer utilities disc. Then put the IBM
formatted disc indrive 0/2 and the BBC
disc in drive 1/3."
32706 REM deleted
```

SINGLE DRIVE OPERATION

Single drive operation is not catered for. It is, to put it mildly, inefficient and tedious in the extreme, requiring as it does repeated removal and re-insertion of discs to swap between BBC and IBM.

Mr Toad's M/C Corner (continued from page 52)

address in brackets with no X or Y mentioned - this is the 65C02's new mode, 'indirect zero page'. The list is as follows: ADC, AND, CMP, EOR, LDA, ORA, SBC and STA.

Now, f'rinstance, LDA (&F2) is the same as LDA (&F2),Y if Y=0, so the easy fix is to set Y to zero, then put ',Y' at the end of the instruction - e.g. LDY #0:CMP (address),Y or LDY #0:EOR (address),Y. If this fails, it must be because Y has been corrupted, so push it or save it to memory first.

Finally, there is JMP (ind,X) where ind is a 16-bit address. This ought to be fixable by TXA:CLC:ADC ind:STA ind:LDA ind+1:ADC #0:STA ind+1:JMP (ind), though there is a bug in the 6502 which will mess that up if ind=&FF. You may need to save the contents of A, as usual, and you will probably need to save the original contents of ind and ind+1 and restore them before the next call to this routine.

So now you know. I hope that Jorge Alvoeiro of Hull, who wrote in recently to ask about the 65C02, finds himself adequately answered. All you need to do before attempting a modest-sized conversion, Jorge, is to book a fortnight off work, lay in large stocks of gin, Valium and fags and retain a good divorce lawyer. A bucket of hair dye could come in handy, too.

That's about it for now, mammal types. This month's competition: what does the quotation from Alfred mean? First wrong answer wins a copy of Sweet's Anglo-Saxon Primer and a long test-paper. Aelcum menn gebyreth, the aenigne godne craeft haefth... Next month we discuss Prof. Hodenschmerz's revolutionary theory: was the Cotton MS of Beowulf really produced on View, and not on Locoscript as Zucker maintains?

```
TRING$(M%,".");STRING$(M%,CHR$8);:PROCb(
1):REPEAT:K%=FNb:IFK%=127:IFA$<>"":VDU8,
46,8:A$=LEFT$(A$,LENA$-1)
 1320 IFK%>31ANDK%<127:IFLENA$<M%:VDUK%:
A$=A$+CHR$K%
 1330 UNTILK%=13:PROCb(0):PRINTSTRING$(M
%-LENA$,".");:=A$
 1340 DEF PROCg:PROCd:REPEAT:PROCl("Data
Boss  -  By Jon Ribbens",5):PROCh:IFY%=7
ORY%=8:PROCn
 1350 IFX%>2IFX%<21IFY%>2IFY%<6PROCl("En
ter record number to go to",0):CLS:PRINT
:I%=VALFNc(18):CLS:IFI%<1ORI%>M%ORFNf(I%
)=0:PROCl("No such record/Record searche
d out",0):PROCd:PROCj ELSEIFX%>2IFX%<21I
FY%>2IFY%<6R%=I%:PROCd
 1360 UNTIL0
 1370 DEF PROCn:IFX%>40ANDX%<50:IFR%<(M%
+1)IFR%<T%PROCv:PROCd ELSEIFX%>32ANDX%<3
9:PROCp
 1380 IFX%>51ANDX%<62IFR%>1PROCw:PROCd E
LSEIFX%>63ANDX%<71:IFR%>1:PROCx:PROCd EL
SEIFX%>72ANDX%<79:IFR%<>M%:IFM%>0:PROCy:
PROCd
 1390 IFX%>22ANDX%<31:PROCq ELSEIFX%>1AN
DX%<11:PROCr ELSEIFX%>12ANDX%<21:PROCt
 1400 ENDPROC
 1410 DEFFNe(M%):LOCALA$,I%:FORI%=1TOM%:
A$=A$+CHR$(BGET#H%):NEXTI%:=A$
 1420 DEF PROCo(A$,M%):LOCALI%:IFLENA$>0
:FORI%=1TOLENA$:BPUT#H%,ASCMID$(A$,I%):N
EXTI%:IFLENA$=M%:ENDPROC
 1430 FORI%=1TOM%-LENA$:BPUT#H%,46:NEXTI
%:ENDPROC
 1440 DEF PROCp:PROCl("Enter new data or
 press RETURN to reinstate old line",3):
FORI%=1TOF%:VDU31,0,I%-1:A$=FNc(l%(I%)):
IFA$<>"":f$(I%)=A$:IFR%>M%:M%=R%
 1450 IFA$="":VDU31,0,I%-1:PRINTf$(I%);
 1460 NEXTI%:PROCm(R%):IFR%=T%PROCl("Wai
t ...",3):FORI%=1TOF%:f$(I%)="":NEXTI%:F
ORI%=T%+1TOT%+20:PROCm(I%):NEXTI%:T%=T%+
20:PTR#H%=D%-4:BPUT#H%,T%MOD256:BPUT#H%,
T%DIV256
 1470 PTR#H%=D%-2:BPUT#H%,M%MOD256:BPUT#
H%,M%DIV256:PROCd:ENDPROC
 1480 DEF PROCq:IFFNy("Delete this recor
d (Y/N) ?")=0:ENDPROC
 1490 FORI%=1TOF%:f$(I%)="":NEXTI%:IFR%=
M%:IFR%>1:M%=M%-1:PTR#H%=D%-2:BPUT#H%,M%
 1500 PROCm(R%):PROCd:ENDPROC
 1510 DEF PROCr:PROCa(3):CLS:RESTORE:FOR
I%=1TO5:READA$:PRINTTAB(27-LENA$/2,I%+2)
;A$:NEXTI%:PRINTTAB(21,1);"General Menu"
:PROCh:IFX%<24ORX%>78ORY%<13ORY%>17:CLS:
PROCd:ENDPROC
 1520 IFY%=13CLS:PROCd:REPEATPROCp:R%=R%
+1:PROCd:UNTILFNy("Enter another (Y/N) ?
")=0 ELSEIFY%=14:PROCs ELSEIFY%=16:CLS:P
RINT"*";:A$=FNc(54):PRINT:OSCLIA$:PROCl(
"Press a key",3):I%=FNb
 1530 IFY%=17:CLOSE#0:FORI%=0TO4:PROCa(I
%):CLS:NEXTI%:PROCl("DataBoss ended.",3)
:END ELSEIFY%=15FORI%=a%TOa%+508 STEP4:!
I%=-1:NEXTI%
 1540 CLS:PROCd:ENDPROC
 1550 DATAAuto entry,List technical info
rmation,Reset search,OS Command,Exit Dat
aBoss
 1560 DEF PROCs:CLS:PRINTTAB(16,1);"Tech
nical information"''"DataBoss file #";H%
'"FDR length ";D%'"Record length ";L%'"F
ile length ";EXT#H%'"Fields ";F%'"Last r
ecord entered ";M%'"Last record created
";T%:PROCl("Press a key",3):I%=FNb:ENDPR
OC
 1570 DEF PROCt:PROCl("Enter string to s
earch for",3):CLS:PRINTTAB(20,1)"Search
Routine"':A$=FNc(55):CLS:PROCl("Searchin
g for "+A$+" ...",3):J%=0:FORI%=1TOM%:PR
OCu(I%):NEXTI%:IFJ%=0:PROCl("No matches
found",3) ELSEIFJ%=1:PROCl("1 match foun
d",3)
 1580 IFJ%<>0ANDJ%<>1:PROCl(STR$J%+" mat
ches found",3)
 1590 CLS:PROCx:PROCd:PROCj:ENDPROC
 1600 DEF PROCu(R%):IFFNf(R%)=0ENDPROC E
LSELOCALI%:PROCz(R%,0):PROCk(R%):FORI%=1
```

```
TOF%:IFLENf$(I%)>LENA$:IFINSTR(f$(I%),A$
):J%=J%+1:PROCz(R%,1):I%=99
 1610 NEXTI%:CLS:ENDPROC
 1620 DEF PROCv:Z%=R%:REPEATR%=R%+1:UNTI
LFNf(R%)<>0ORR%>(M%+1):IFR%>(M%+1)R%=Z%
 1630 ENDPROC
 1640 DEF PROCw:Z%=R%:REPEATR%=R%-1:UNTI
LFNf(R%)<>0ORR%=1:IFFNf(R%)=0R%=Z%
 1650 ENDPROC
 1660 DEF PROCx:Z%=R%:R%=0:REPEATR%=R%+1
:UNTILFNf(R%)<>0ORR%>(M%+1):IFR%>(M%+1)R
%=Z%
 1670 DEF PROCy:Z%=R%:R%=M%+1:REPEATR%=R
%-1:UNTILFNf(R%)<>0ORR%=1:IFFNf(R%)=0R%=
Z%
 1680 ENDPROC
 1690 DEFFNf(R%)=-SGN((a%?(R%DIV8))AND(2
^(R%MOD8)))
 1700 DEF PROCz(R%,A%):IFA%=0:?(a%+(R%DI
V8))=?(a%+(R%DIV8))AND(255-(2^(R%MOD8)))
:ENDPROC ELSE:?(a%+(R%DIV8))=?(a%+(R%DIV
8))OR(2^(R%MOD8)):ENDPROC
```

Pointb

```
   10 REM Program Pointb
   20 REM Version B 1.0
   30 REM Author  J. Ribbens
   40 REM BEEBUG  December 1992
   50 REM Program subject to copyright
   60 :
  100 FORp=0TO2STEP2:P%=&900:[OPTp
  110 JMP showpointer
  120 JMP deletepointer
  130 :
  140 .showpointer
  150 JSR shadow:JSR calcaddr
  160 LDY#0:.lp1
  170 LDA(&70),Y:STAstordat,Y
  180 ANDmaskdat,Y:ORApoindat,Y
  190 STA(&70),Y:INY:CPY#8:BNElp1
  200 JSRnewline:LDY#0:.lp2:LDA(&70),Y
  210 STAstordat+8,Y
  220 ANDmaskdat+8,Y:ORApoindat+8,Y
  230 STA(&70),Y:INY:CPY#8:BNElp2
  240 .main:LDA&8F:BEQnomain:LDA#108
  250 LDX#0:LDY#0:JMP&FFF4:.nomain:RTS
  260 :
  270 .poindat
  280 EQUD &60400000:EQUD &7E7C7870
  290 EQUD &0C0C1818:EQUD &00000606
  300 :
  310 .maskdat
  320 EQUD &070F1FFF:EQUD &00000103
  330 EQUD &E0C1C100:EQUD &FFF0F0E0
  340 :
  350 .stordat
  360 EQUD &00000000:EQUD &00000000
  370 EQUD &00000000:EQUD &00000000
  380 :
  390 .deletepointer
  400 JSR shadow:JSR calcaddr
  410 LDY#0:.lp3
  420 LDAstordat,Y:STA(&70),Y
  430 INY:CPY#8:BNElp3:JSRnewline
  440 LDY#0:.lp4:LDAstordat+8,Y
  450 STA(&70),Y:INY:CPY#8:BNElp4
  460 JMPmain
  470 :
  480 .calcaddr
  490 LDA#&3000 MOD256:STA&70
  500 LDA#&3000 DIV256:STA&71
  510 CPX#0:BEQ doy
  520 .lp5:LDA&70:CLC:ADC#8:STA&70
  530 LDA&71:ADC#0:STA&71:DEX
  540 BNElp5:.doy
  550 CPY#0:BEQ endcalc
  560 .lp6:JSRnewline:DEY
  570 BNElp6:.endcalc RTS
  580 :
  590 .newline
  600 LDA&70:CLC:ADC#&80:STA&70
  610 LDA&71:ADC#&02:STA&71
  620 RTS
  630 :
  640 .shadow:PHX:PHY
  650 LDA&8F:BEQnoshadow
  660 LDA#108:LDX#1:LDY#0:JSR&FFF4
  670 .noshadow:PLY:PLX:RTS
  680 ]:NEXT
  690 *SAVE Pointer 900 9E0
```

Personal Ads

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

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

Mega 3 £35, AMX SuperArt £10, System Delta £10, Edword2 £5, Dumpmaster £6, Printwise £4, Printmaster £5, Toolkit £4, Exmon £4, Viewstore £15, Printer Driver (View) £3, Artist (Peartree) £8, Spellcheck II £6, GrandPrix £4, AMX Pagemaker £8, Mouse £8, 2way printer switch £6, Brother IF-50 typewriter interface £15, all originals with handbooks etc. Tel. (0623) 27423.

WANTED: New members for BBC/Master user group, for free introductory disc magazine & PD software send disc & return p&p to 8-Bit software, 1 Oakwood Drive, Heaton, Bolton BL1 5EE.

WANTED URGENTLY: Latest versions 5.25"; Shibumi Problem Solver + Tull mouse driver, Gem arrive v200, Microsoft flight simulator v2.12 or later, Gem 1st Word Plus, Volkswriter Deluxe, Gem 3 Interface, Deluxe Paint II Gem DTP, Ventura Publisher V2.0 + professional extension, Eurolink/Pace Linnet Autodial/Answer modem, + comms software, may also require Watford BBC hard disc. Tel. (0621) 815162 eves and ask for Chris.

Dead Beeb. Lots of Beeb software and hardware all boxed with manuals, please call for list. Tel. (0442) 214157.

Master 80186 ('512') 2nd processor with Essential Software 1Mb upgrade, DOS+ 2.1, all discs and mouse (inc. Tull driver), plus books and software (WP, DTP, utilities etc.) £125, 20Mb hard disc (ADFS/DOS partitions) £80, Beebug Master ROM £15, View Professional V2.0 (all media) £20, Acorn Master MOS upgrade £20, Viglen PC style keyboard case etc. for Master £12, Le Modem £25, Ventura Publisher V1.0 (runs on expanded M512) original discs and manuals £50, Adaptor to fit 3.5" drive in 5.25" bay £7.50, Wordwise Plus inc. manuals £15, Baksoft DOS/CPM/BBC copy program £15, other beeb software please call for more details. Tel. (0992) 558965.

Beebug magazines vols.1 to 10 complete i.e. 100 issues £35, plus postage for the lot. Tel. (0766) 513270.

ROMs with manuals/discs: Micro prolog, Discaid, Beebfont, Printmaster, Buffer and Backup, £5 each or £20 the lot, Viewspell £10, Viewstore £10, View Professional £20, AMX mouse and SuperArt £20, WE mouse and Quest Paint £20, 10x8k blank EPROMS £12, 10x16k blank EPROMS £15, all add £1 for p&p. Tel. (0934) 842410 eves.

Hardware with ROMs/manuals/discs: Softlife keypad £10, Cumana Touch pad £5, Digital Graph Pad II £25, WE Light Pen £10, Mini Office II (ROM card) £12, Demon modem (auto dial etc.) £20, Delta 14B joystick/keypad with port splitter £10, Acoustic coupler with micronet/prestel ROM £5, external quick change ROM box £10, all add £2 for p&p. Tel. (0934) 842410 eves.

Internals with manuals: WE DDFS (1772 controller) £20, Solidisc DDFS (1772 controller) £15, HCR side RAM board fitted 32k £10, Mini ROM board (4 extra ROM sockets) £5, all add £1.50 p&p. Also Micro User complete vols.1-8 £25, BEEBUG complete vols. 1-11 £35. Tel. (0934) 842410 eves.

Watford Electronics Video Digitiser £60, EMR Midi Interface & leads (incl. Pro-performer, Performer, Link & Editor software £60, Casio HT-3000 keyboard/synthesiser (5 oct, full-size keys, incl. sustain pedal, foot volume and power pack £200, Casio MT-540 keyboard (4 oct, mini keys) £40, Casio FX-700P Programmable Calculator (incl FA-3 cassette interface) £30 ACP 0.25Mb RAM cartridge & Adv 1770 DFS £50, lots of hardware & software, please ring for further details. Tel. (0883) 345294 eves.

Master 512 & colour monitor, double 40/80T 5.25" disc drive, plinth, mouse, all manuals, Star dot matrix printer £400 o.n.o. Tel. (0223) 245263 after 6pm.

BBC B with Wordwise + 40/80T drive CUB monitor, printer £250, also CST Procyon IEEE interface £15, 9" NEC mono monitor £25, Wordwise ROM £10, also lots of tape and disc based software - ring for details. Tel. (0483) 480632.

BBC B/Master, Prestel adaptor complete with ROM, handbook and connectors, nearest offer to £45 secures. BBC B/Master Acorn Viewspell ROM and handbook £15. Tel. 086 732 8776.

Master 128 with Interword, Interbase, DumpOut3 ROMs, Care dual 8 quad cartridges, welcome manuals, reference manuals 1&2, various books, software £220, send for complete list. Tel. (0286) 880997.

BBC B issue 7, os 1.20, DFS, Opus 40/80T disc drive, manuals, many books, lots of software on disc and tape, tape recorder, T/D ROM, magazines £120, Wordwise Plus ROM £12, Disc Doctor £8, Caretaker £8, Exmon II £8, Buffer/Backup £8, Accelerator & G Code £10, all with manuals, ATPL ROM board £20, Pace Nightingale modem with Commstar ROM and manuals £15, send for complete list. Tel. (0286) 880997.

Master 128 Cumana 40/80T, 5.25" drive, Watford 3.5" drive, black and white monitor, Wordwise plus Spellmaster ROMs fitted, all manuals, welcome software £250. Tel. (0345) 740660.

Master 128, all leads etc., cassette recorder, twin 40/80T reference manuals, joystick, three cartridges, installed ROMs, Interword, BEEBUG Master ROM, MB Copy, View suite etc. programs on disc include Master Elite, Masterfile II, Printwise and Viewplot, books include Mastering Assy. Code, BBC Micro ROM book, Understanding Interword, 30hr Basic, Wordwise Plus, Mastering Disc Drives etc. BEEBUG magazines 1987-present plus approx 160 discs, blank or with programs. £400 o.n.o. Buyer collects. Tel. (0454) 778061.

Master 128 with RGB colour monitor and single 40/80T 5.25" disc drive, excellent condition, also Acorn software: Overview I & II £395 o.n.o. Tel. 081-445 9403.

WANTED: Copy of Viewsheet /Viewstore - A Dabhand Guide. Tel. (0442) 256940.

BBC B, DFS, ROM board, dual disc drive, 6502 second processor, many ROMs any offers considered. Tel. 081-204 0424 eves.

Master 128, dual drives, printer, DTP, mouse, Viewspell, Smart cartridge, ADT, many games, all manuals and much more £350. Tel. 061-4425158.

Master 128 with mono monitor, Pace twin 40/80T disc drive plus all necessary cables, plenty of games, utilities and educational discs, most boxed and/or with original

Please do keep sending in your hints for all BBC and Master computers. Don't forget, if your hint gets published, there's a financial reward.

BOOTING A WORDWISE FILE

David Polak

To boot a file *MYFILE* into Wordwise, I find the following useful. It can be typed into Wordwise (ending the file with a Return) and saved as *!Boot*.

```
*BASIC
*WORD.
:SELECT TEXT
:LOAD TEXT "MYFILE"
:DISPLAY
```

Don't forget to *OPT 4 3 the disc to enable the boot procedure.

JUSTIFIED PROPORTIONAL TEXT WITH VIEW

D.Jowers

It may be possible to output justified proportional text from View by using the software inside your printer. For example, using a Panasonic KX-P1124i (which is compatible with the Epson LQ-850), the following set of escape codes is required:

```
ESC "x" 1
ESC "p" 1
ESC "l" 1
ESC "Q" 74
ESC "a" 3
```

These commands do the following: set NLQ mode, set proportional spacing, set left-hand margin to 1 (note that it is a lower case L, not a 1, in the quotes), set right-hand margin to 74 and set justification. As the margins are being set on the printer it is essential that margins are not set within View using the LM command. The ruler should be set to give a reasonable 'fill' with proportional spacing - I have found a line length of 80 suitable with the above printer settings. It is important to note that with proportional spacing the margins are based on 10 characters per inch, and that margins do need to be explicitly specified.

To obtain tabulated text it is necessary to send an ASCII 9 for each tab and, if the tab settings required are different to the printer default settings, to set the printer tabs with the following escape sequence:

```
ESC "D" n1 n2 ... nx 0
```

where *n1 n2* ... nx are the required tab positions (based on 10cpi).

FUNCTION KEY CORRECTION

Brian Lowe

I have discovered an error in my *Programming the Function Keys for Mode 7* hint in BEEBUG Vol.10 No.10. Shift-Ctrl-F6 gives an unfilled block (i.e. a space) and not a filled block as described. The filled block can be obtained by typing:

```
VDU 255
```

followed by Return. The displayed square can then be inserted, using the Copy key, between two quotation marks. For example, to program the F7 key to produce a block, one would use:

```
*KEY 7 "<block>"
```

where *<block>* is the block produced by the above.

MORE UNDEFINED DEFINITIONS

Simon Myers

There is another use for the undefined variable (see Hints in Vol.10 No.9 and Vol.11 No.3). If there is a routine in a program which dimensions some memory locations, then it is sensible to dimension the workspace once only, and only if it is needed. The following example routine cures both these problems neatly:

```
DEF PROCallocate_memory
mem%=mem%
IF mem%=0 THEN DIM mem% 1000
...
rest of procedure
...
ENDPROC
```

The variable *mem%* will only be allocated the first time the routine is called; the second line prevents the third line from producing a "No such variable" error.

documentation. Books included:- Master Welcome Guide, Dabhand Guides' View, BBC Microcomputer User Guide, Discovering BBC Micro Machine Code, Advanced Machine Code Techniques for the BBC Micro, BEEBUG mags, March 1991 to date £250. Acornsoft ISO Pascal (ROMs & Disc), boxed with manuals plus cartridge as new £30. Tel. (0254) 823902.

Panasonic KX-P1080 printer, manual £90 or make an offer, Chameleon Palette - 8 colours from 4096 with BBC £25. Tel. (0707) 325034 eves.

BBC issue 7, DFS, Aries B32 Shadow RAM and B12 ROM board with 16k Sideways RAM, AMX mouse and Stop Press, other ROMs: Exmon II, Sciways, Advanced Disc Investigator, Watford double plinth, good condition, manuals £150. Tel. (0273) 844951.

Solidcad/superdump package £130 no offers, also BBC equipment disc drive GXR B+ ROM Dumpout 3 ROM Wordwise ROM Termulator ROM, 10 games plus other items, call for more details. Tel. (0621) 815162 after 6pm.

WANTED: Aries B-12 Sideways ROM board, with fitting instructions if possible. To help elderly beginner to the world of computers. Tel. (0646) 681275.

Music software: Music Master with microphone interface, 5.25" disc, handbook Mupados Recorder Tutor with ensemble, duet and classroom Network packs (5x 5.25" discs, handbooks and cassettes), Micro Maestro with 5.25" disc and 6 cassettes, all for £32 plus postage (worth £179). Tel. (0256) 27018.

Blank discs 5.25" 80T D/S flip over, (can read/write both sides as drive 0 or use as normal D/S) £3/10 or £20/100. Tapes >20 full of games/programs/utilities etc. £15, Micro User monthly discs Feb '89/Mar '91 26 discs 80T £15. Tel. (0934) 842410 eves.

Two BBC's for sale: 1 an issue 7, DFS, no drive £100, the other has the works; Viglen separated case, mono monitor 40 SS and 40/80T DS drives, Sidewise ROM PCB, Music 500/5000 synth: mouse + AMX S'art + PCB designer, joystick, Fleet Street Editor £29, ROMs inc WPs and spreadsheet, 142 discs in boxes, 7 books £200 o.n.o. Tel. (0284) 762302 eves.

BBC B issue 7, OS 1.20 in excellent condition £115 incl. choice of free software if wanted. Tel. (0821) 642652.

WANTED: Can anyone lend/sell me WE 'WordAid' instruction manual as I've lost mine. Tel. (0821) 642652.

Wordwise + ROM, FKey strip, introduction, reference manual, fitting instructions and Norwich Computer Services 40T disc of segment programs £15, Spellcheck II ROM & 40T disc £7, Printwise 40T disc & manual £10, Acorn User Loose Copies, Nov'82 to Dec '89 (July '87 missing) £10 + carriage, Micro User in binders, March '83 to June '86 £7 + carriage, Micro User Loose copies, July '86 to May '88 £4 + carriage. Tel. (0536) 770478 eves.

First Word Plus (II), Acorn DTP, Genesis all £20 each, Pacmania £10, Spreadsheet (Mk5) £5. Tel. 061-442 5158.

Font FX, Draw Bender £2 each, Glimpse, Help Companion £3 each Mad Professor Mariarti, Pacmania, Arcade Soccer, Fish, Fireball II £5 each, FunSchool 2 (under 6), FunSchool 2 (6-8) £6 each, Droom, Personal Accounts, Interdictor 2, Apocalypse 2 £10 each, Pendown £25. Tel. 081-292 6870 eves.

Public Domain Software and Shareware for the BBC Micro, Master and Master 512 co-processor

Amongst the latest additions are - the sequel to Acornsofts' 'Gateway to Karos' and Howard Spurr's 'Disc Duplicator III'

Send £1.50 for catalogue and sampler disc to;

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

Make cheques payable to A Blundell or send an A5 s.a.e. for more details

Master 128 including View & Viewsheet, boxed as new, only light home use £99, CC Interchart ROM and manual, boxed as new, never used £7.50, CC Print Master ROM and manual, boxed as new, never used £7.50, Fourth Dimension Holed Out Golf, BBC B/Master 5.25" disc, boxed as new £4.99, BBC soft VU-Type Professional - Pitman typing tutor, BBC B/Master 5.25" disc and manual, boxed as new £3.99, all originals and with full documentation, will sell separately or together, ideal for a beginner or school, make me an offer! This is all I have left now from selling a complete system, no reasonable offer will be rejected. Tel. 081-694 9340 eves or w/ends or 081-697 6726 during office hours.

NEC 3D multisync monitor £225, LISP £30 boxed Brother 1709 wide carriage Epson-compatible NLQ printer with manual £95, First Word Plus £40, ROM/RAM podule £20, A3000 midi interface £25. Tel. (0483) 480632.

BEEBUG mags Vol.1 No.1 to Vol.10 No.10 (2 missing) offers to include postage £10 (16.5lbs) will deliver free to East Central Scotland area. Tel. 031-539 4820.

Interbase £35, Interword £25, Mini Office II £7, Novacad + Plotter £25, NTQ £12, Publisher £20, Slave++ £5, Sleuth £12, Speech £5, Stop Press £20, Studio 8 £8, System Delta £30, Reference Guide £8, Inter/View Link £10, Mail Shot £10, Reporter £10, System Gamma £25, Toolkit Plus £12, Viewstore £20, Word-Ex £10, Wordpower £20, Wordwise Plus £10, WYSIWYG £10, Solidisk SWR 128 £15, Teletext Decoder + ATS £45, Z80 + Perfect software £60, 32k Shadow RAM board £30, Watford ROM board £15. Much more software, books etc. please send large s.a.e. for list. Tel. (0423) 886359.

Software for BBC Master and 512 - Essential software Fastboot £5, Co-processor filing system £12, Miscellaneous utilities £5, Commandline mouse £5, RAM disc and utilities £8, Printscreen utilities £7, Memopad £5, Dabs Press Shareware collection No.1 £5, No.2 £5, or £8 for both, Master 512 User guide book and disc £6, Sidewriter £3, ROMs - Spellmaster with manual £12, Master ROM with manual £12, Questpaint with utilities disc and manual £12, Disc - Masterfile II £5, Keyword (Thesaurus) £2, Dumpmaster £2, Hershey characters (2 discs) £3, Magscan £2, Mewsoft (1) Faxfile organiser (2), A4 forms designer (3), Fancy labels £2 each or £ the set, books with accompanying discs - Advanced sideways RAM User Guide £6, Wordwise Plus, a Users Guide £6, Master 512 User Guide £6, BEEBUG mags Vols.1 to 10 £4 per volume or £25 the lot, BEEBUG discs Vols.7 to 11 details on request. Tel. (0440) 702311.

WANTED: ROM/RAM board and ROMs suitable for BBC B. Please write to; Mr T.C Brown, 8 Flemming Gardens, Tilbury, Essex RM18 8JR with details and prices.

WANTED: ADE+ for BBC micro, must have all documentation, also any interested SWR machine code programmers with OPUS DDOS systems to share new ideas, I have developed 16k support ROMs to use with DDOS/View. Tel. 091-268 4609 day.

Switched mode PSU model EXT90/12 £30, Spellcheck III + ROM £17, Toolkit ROM £7, Spellcheck discs £10, Quickcal disc £9, OverView EPROMs £60, Master Operating System ROM £18, BEEBUG discs list available £3 each, 2764 new EPROMs £.250 each 2732a new EPROMs £3.25 each, 6264 LP RAM £3.75 each, 19" rack with guides £30, 6809 CPU single board computer, OS software available - on disc (ring for details). Tel. (0254) 701573.

BBC model B issue 7 with DFS and Watford ROM/RAM expansion board with various ROMs fitted £125 o.n.o. printer Riteman Inforunner 80 col. needs new ribbon £25, also a Master Compact keyboard unit for spares (OS ROM is missing!) £20. Tel. (0622) 756071.

A3000 colour monitor, PC Emulator, stand, original boxes and manuals £490. Tel. (0323) 735841.

Computer Concepts Intersheet spreadsheet - boxed, 2 No. ROMs complete with instruction manual £20. Tel. (0732) 863105.

BEEBUG magazines complete set from Vol.2 No.9 (March '84) to Vol.8 No.9 (March '90), offers please. Tel. (0684) 572295.

Omniscope relieves eye strain and magnifies text £50. Tel. 081-318 5155.

GOOD HOMES FOR OLD MACHINES

I read with interest Alex Belton's hint about converting an idle Master to an AM radio. Can I, though, put in a plea to owners of Masters and model Bs who are considering putting out their now defunct machines to consider donating or selling their machines at a peppercorn amount to their local primary or special schools.

Many small schools will not be able to upgrade to Archimedes systems to any great extent given the increasing effect of cuts in education budgets, etc. Special schools in particular will consider staying with their non-Archimedes machines for as long as possible given the adequate memory capacity available for their needs.

B.Kemp

DATA TRANSFER WITH WORDWISE PLUS

My letter is inspired by the article on page 26 of the November issue of BEEBUG (Wordwise User's Notebook, BEEBUG Vol.11 No.6). I supply systems which generate the legal documentation used in debt collection. These systems do fairly complex interest calculations in Basic and then pass the data on to WW+ to create the documents themselves.

Although resident integer variables are very useful for storing and transferring certain types of data (such as the date), the easiest way to pass anything but the most basic integer data is by using the function keys.

For example, if I want to pass A$, B$ and C$ to a WW+ program, assuming their total length is less than 250 bytes, I can use:

```
    OSCLI"KEY1 "+A$+"|M"+B$+"|M"+C$+"|M"
```

I can also use:

```
    *KEY *WORDWISE|M:SE.SEG.0|M:LO.T
E."WWPROGRAM"|MX|!|@
```

followed by *FX3,2 (to turn off the screen), then *FX138,0,128 (to effect the 'pressing' of function key 0), and finally END. The system will shift invisibly to Wordwise Plus, where *WWPROGRAM* will be loaded into segment 0 and run. Function key 1 can then be 'pressed' by using *FX138,0,177, and its contents read by:

```
    A$=GLK$:B$=GLK$:C$=GLK$
```

Then *FX3,0 can turn the screen back on for an interactive WW+ program. I can go back to the Basic program with:

```
    *EXEC !BOOT:*WORDWISE:END
```

where !BOOT contains:

```
    *BASIC:CHAIN"BPROGRAM"
```

I have also written routines which 'read' outline documents and fill in the blanks either from individual files or by asking questions of the operator on-screen.

Eventually, I am sure that I will have to go over to a DOS machine, but I doubt that I will find a better combination than BBC Basic and Wordwise Plus to do the job.

Tom Hervey

UNDERSTANDING THE TERMINOLOGY

I always read the 512 Forum pages, which I think are for owners of the Master 512, but I am continually confused by reader's ads selling things like 512 co-processors, Z80 boards, 65C02 processors, M512 boards, 80186 processors, Turbo boards etc. Can you, for the benefit of readers, explain what they all mean? I am sure I am not the only one confused by the terminology.

T.D.Parsons

Briefly, a Master 512 is a Master 128 fitted with a 512 co-processor, and a Master Turbo is a Master 128 fitted with a 65C02 co-processor. Similar co-processors (referred to originally as 2nd processors) were available as separate units for connecting to a model B (Z80 CP/M, 65C02 etc.). The Cambridge Workstation, a Master fitted with an 32-bit 32016 co-processor, was a quite different system intended for use in higher education and research, and with a totally different operating system.

BEEBUG MEMBERSHIP

SPECIAL OFFERS TO BEEBUG MEMBERS

BEEBUG members can purchase BEEBUG and RISC Developments' products at special discounted prices. For additional offers see the centrefold in the magazine.

0077b	C - Stand Alone Generator	14.56	0084b	Command	29.88
0081b	Masterfile ADFS M128 80 T	16.86	0073b	Command(Hayes compatible)	29.88
0024b	Masterfile DFS 40 T	16.86	0053b	Dumpmaster II	23.76
0025b	Masterfile DFS 80 T	16.86	0004b	Exmon II	24.52
0074b	Beebug C 40 Track	45.21	0087b	Master ROM	29.88
0075b	Beebug C 80 Track	45.21	1421b	BEEBUG Binder	4.20

SUBSCRIPTION RATES

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

BEEBUG		BEEBUG & RISC USER
£18.40	1 year UK, BFPO, Ch.I	£28.90
£27.50	Rest of Europe & Eire	£42.90
£33.50	Middle East	£53.10
£36.50	Americas & Africa	£58.40
£39.50	Elsewhere	£62.50

BACK ISSUE PRICES (per issue)

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

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

POST AND PACKING

Please add the cost of p&p when ordering. When ordering several items use the highest price code, plus half the price of each subsequent code.

Stock Code	UK, BFPO Ch.I	Europe, Eire	Americas,Africa, Mid East	Elsewhere
a	£1.00	£1.60	£2.40	£2.60
b	£2.00	£3.00	£5.00	£5.50

BEEBUG 117 Hatfield Road, St.Albans, Herts AL1 4JS
Tel. St.Albans (0727) 40303, FAX: (0727) 860263
Manned Mon-Fri 9am-5pm (for orders only 9am-6pm and 9am-5pm Saturdays)
(24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by RISC Developments Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Technical Editor: Mark Moxon
Editorial Assistance: Marshal Anderson
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Sue Baxter
Managing Editor: Sheridan Williams

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

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

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

In all communication, please quote your membership number.

RISC Developments Ltd (c) 1992

ISSN - 0263 - 7561

Printed by Arlon Printers (0923) 268328

Magazine Disc

December 1992

DATABOSS - DataBoss is a standard text database, which will allow you to keep addresses and fairly simple data which needs little manipulation. The disc contains an example data file containing a number of addresses.

JOBLOG (PART 1) - This is the first part of the Job Log Personal Organiser program. The program on this disc enables the so-called Principal Index to be set up for your data, and next month's installment will implement many more features.

USING SIDEWAYS RAM (PART 2) - The program ROMHead assembles a small sideways ROM image as an example of how to write your own images. The program has been structured so you can create sideways ROMs easily and without having to learn all the details of the ROM header.

WORD-SEARCH PUZZLE GENERATOR - The program WordSq generates a word-search grid containing up to thirty words of your choice. The word-search can be printed, both in full and 'skeleton' form (for the solution).

BEEBUG WORKSHOP - The program on the disc shows inorder tree traversal using a threaded tree structure defined in DATA statements at the end of the program. The program can be customised to define your own tree structure.

STYLE CHECKER - The style checker analyses text to help improve its style. Also included on the disc are two text files, giving examples of good and bad style, and a machine code version of the style checker which works at a much higher speed than its Basic counterpart.

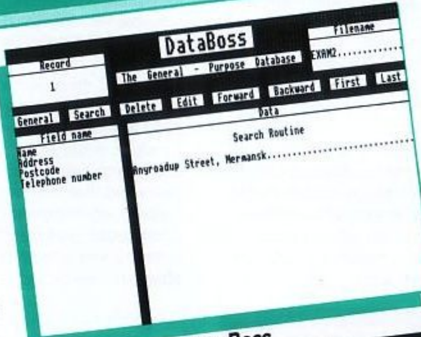
IBM/BBC FILE TRANSFER - The BBC/IBM transfer programs and the PC formatter have been brought together under one menu system, and a help option is also included to remind you how to use the programs. This complete suite of programs will allow you to transfer files between DFS and MS-DOS without any problems.

MR TOAD'S MACHINE CODE CORNER - The program MBlast is the Master Blaster ROM described in this month's article. It analyses programs and prints out any assembler mnemonics which will only work on the Master and not the BBC B.

THE TWELVE DAYS OF CHRISTMAS - The first of two Christmas bonuses, this nostalgic program from BEEBUG Volume 2 Issue 7 plays 'The 12 Days of Christmas' in three part harmony, while scrolling the lyrics up the screen. Sing along with your computer this Christmas!

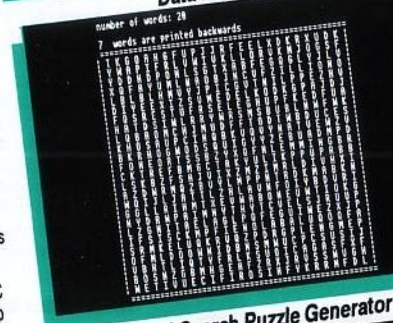
CHRISTMAS CAROL - The second of our Christmas bonuses shows the flocks grazing in the fields of the Biblical shepherds, with the star of Bethlehem shining in the background. There are three dulcet carols included to soothe you gently into the Christmas spirit.

Bibliography for this issue (Vol.11 No.7).



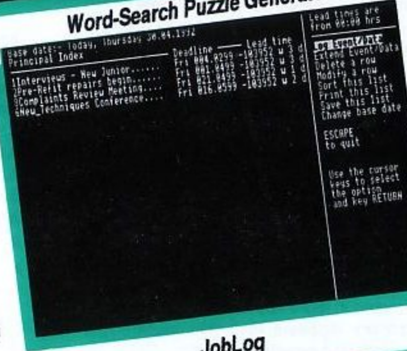
```
                              DataBoss                 Filename
Record                                                 EXAM2............
                   The General - Purpose Database
1                  Delete  Edit  Forward  Backward  First  Last
General  Search                       Data
Field name                       Search Routine
Name
Address            Anyroadup Street, Mermansk..........................
Postcode
Telephone number
```

Data Boss



```
number of words: 20
7  words are printed backwards
```

Word-Search Puzzle Generator



```
Base date:- Today, Thursday 30.04.1992                          Lead times are
Principal Index                    Deadline ——— Lead time       from 00:00 hrs
1Interviews - New Junior......     Fri 004.0299 -103952 w 3 d
2Pre-Refit repairs begin......     Fri 001.0499 -103952 w 3 d   Log Event/Data
3Complaints Review Meeting....     Fri 021.0499 -103952 w 1 d   Extend Event/Data
4New_Techniques Conference....     Fri 016.0599 -103952 w 2 d   Delete a row
                                                                Modify a row
                                                                Sort this list
                                                                Print this list
                                                                Save this list
                                                                Change base date

                                                                ESCAPE
                                                                to quit

                                                                Use the cursor
                                                                keys to select
                                                                the option
                                                                and key RETURN
```

JobLog

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + £1 P&P (50p FOR EACH ADDITIONAL ITEM)

Back issues (5.25" and 3.5" discs from Vol.6 No.1) available at the same prices.

DISC (5.25" or 3.5") SUBSCRIPTION RATES	UK ONLY	OVERSEAS
6 months (5 issues)	£25.50	£30.00
12 months (10 issues)	£50.00	£56.00

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

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

Upgrading to an Archimedes

We know that many BEEBUG readers have already upgraded to an Archimedes, and no doubt many more will choose to follow a similar route. For their benefit we offer our advice to help them make a sensible decision on whether to upgrade and if so, what path to take.

Any prices quoted relate to our associated company Beebug Ltd., but note that all prices, particularly those on trade-ins and secondhand items, are likely to change without notice. You should always telephone or write for the latest information.

Archimedes A5000

What System to Choose

All new Archimedes systems are now supplied with the RISC OS 3.10 operating system. Any secondhand system should be upgraded to this. Based on the experience of existing users, we would strongly recommend a minimum of 2Mb of RAM. Most users find a hard disc adds significantly to the convenience of using an Archimedes, but you can always add a low-cost hard drive later, and more memory, but check on the likely price of future expansions - it is not necessarily the same for all machines. If you might be interested in more specialised add-ons (scanners, digitisers, etc.) then check the expansion capability of your preferred system.

Compatibility and Transferability

You will need to decide to what extent you wish to continue using existing discs and disc drives on an Archimedes. An Archimedes and a BBC micro can be directly connected for transfer of files. You can also connect a 5.25" drive to an Archimedes via an additional interface to continue to access 5.25" discs (ADFS format). Our DFS Reader will also allow files to be transferred to the Arc from DFS format discs. However, none of this is possible with the latest A3010/A3020/A4000 systems.

Much BBC micro software will run directly on an Archimedes, or via the 6502 emulator. However, consider this carefully; in our experience, despite prior misgivings, most Archimedes users find that they rapidly adjust to the Desktop environment of the Archimedes, and quickly abandon the software and data of their old system after an initial period.

Software for the Archimedes

The Archimedes is supplied complete with a range of basic applications software. Before embarking on any further purchases it may be better to familiarise yourself with the new machine. Most users look for a word processor (or DTP package), maybe a spreadsheet, or a database, plus other more specialist software. We cannot give detailed guidance here, but back issues of RISC User contain a wealth of useful information - we can advise on suitable issues.

General Advice

It is advisable to discuss your requirements with the BEEBUG technical team before making a final decision on what you want. Try to anticipate future expansion needs at the outset. Note: the price on some systems includes a monitor; in other cases a choice of monitor is available at an additional cost. The details given in the table are minimum specifications of the different Archimedes models.

The A3010

It may also be possible to trade in an existing monitor and/or disc drive, but check if your existing monitor is suitable for use with an Archimedes first. You may find it better to advertise your BBC system in BEEBUG and sell privately - this applies particularly to any software and hardware add-ons which cannot be accepted for a trade-in. In future, all personal ads for Archimedes systems in RISC User will also be included in BEEBUG. You may also defer a trade-in until a later date provided you make this clear at the time of purchase.

Archimedes Systems - Typical or Current Prices

		Secondhand	New
+	A310 1Mb RAM	£350	
+	A410/1 1Mb RAM	£565	
+	A420/1 2Mb RAM, 20Mb hard drive	£650	
+	A440/1 4Mb RAM, 40Mb hard drive	£725	
+*	A3000 1Mb RAM	£350	
*	A3010 1Mb RAM, Family Solution		£ 499.00
*	A3020 2Mb RAM, 60Mb hard drive		£1056.33
*	A4000 2Mb RAM, 80Mb hard drive		£1115.08
*	A5000 2Mb RAM, 80Mb hard drive		£1643.83
+*	Acorn standard colour monitor	£145	£ 258.50

All systems above include a single floppy disc drive.
New (*) and secondhand (+) - all prices inc. VAT.
The A5000 price includes a multiscan colour monitor,
A3020/A4000 price includes standard colour monitor.

BBC Micros - Typical Trade-in Prices

Model B (Issue 7)	£ 35
Model B (issue 7) + DFS	£ 75
Master 128	£125
Master Compact	£ 50

For further information on all Archimedes systems contact:
BEEBUG Ltd. 117 Hatfield Road, St Albans, Herts AL1 4JS. Tel. 0727 40303 Fax 0727 860263