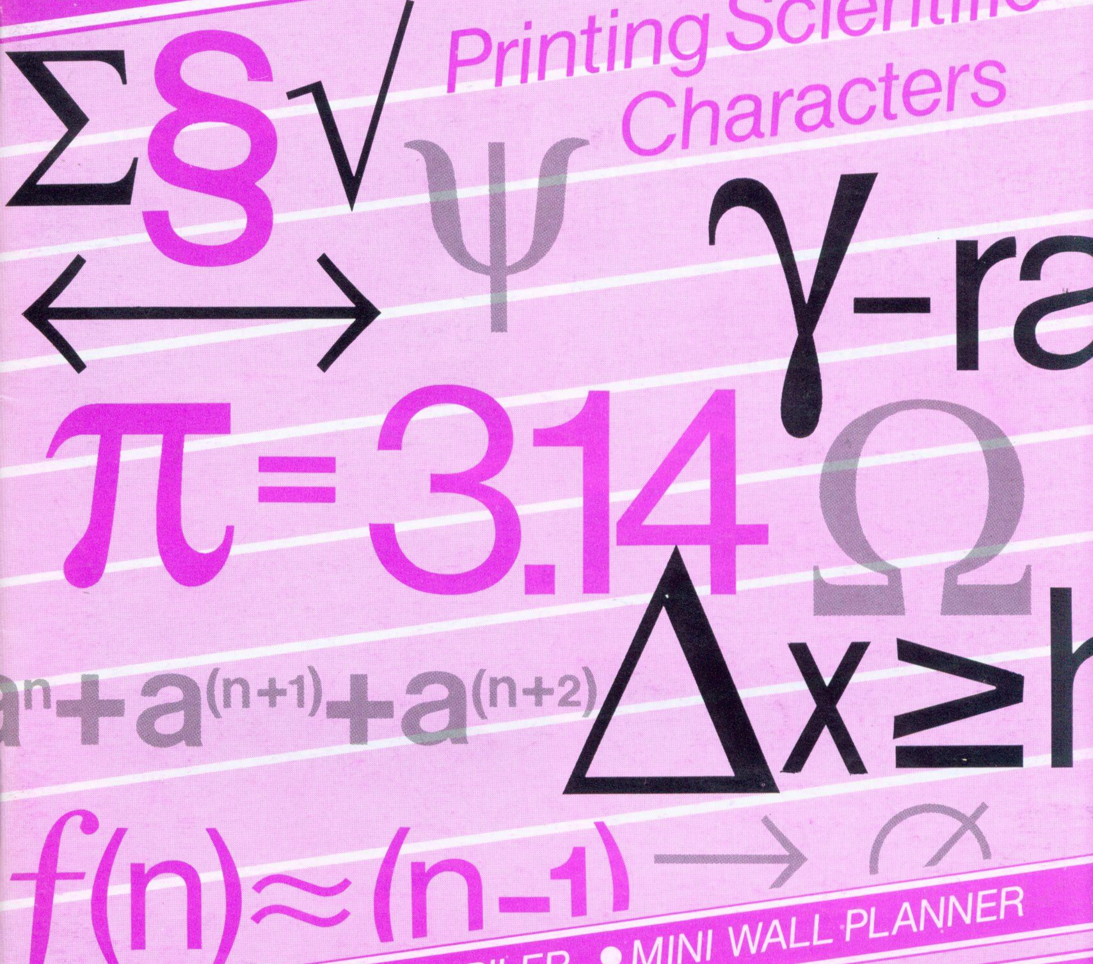
Vol.10 No.1 May 1991

BEEBUG

FOR THE
BBC MICRO &
MASTER SERIES

Printing Scientific Characters



- CROSSWORD COMPILER
- MINI WALL PLANNER
- NEW FONTS FOR THE MASTER
- VIEW TO ASCII

BEEBUG Vol.10 No.1 May 1991

FEATURES

REVIEWS

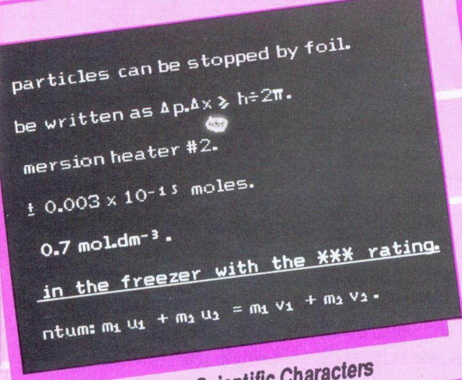
REGULAR ITEMS

HINTS & TIPS

PROGRAM INFORMATION

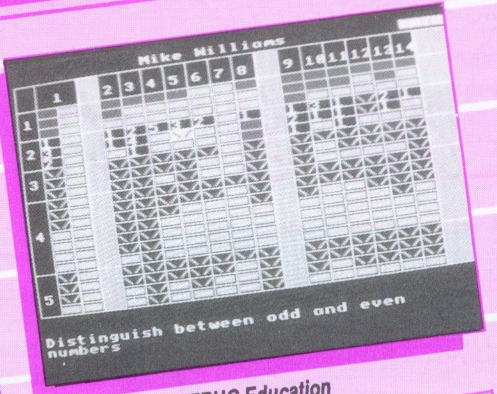
All listings published in BEEBUG magazine are produced directly from working programs. They are formatted using LISTO 1 and WIDTH 40. The space following the line number is to aid readability only, and may be omitted when the program is typed in. However, the rest of each line should be entered exactly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



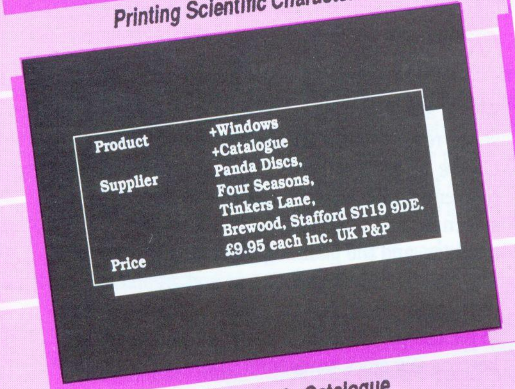
```
particles can be stopped by foil.
be written as Δp.Δx ≥ h÷2π.
mersion heater #2.
± 0.003 x 10-15 moles.
0.7 mol.dm-3.
in the freezer with the *** rating.
ntum: m1 u1 + m2 u2 = m1 v1 + m2 v2.
```

Printing Scientific Characters



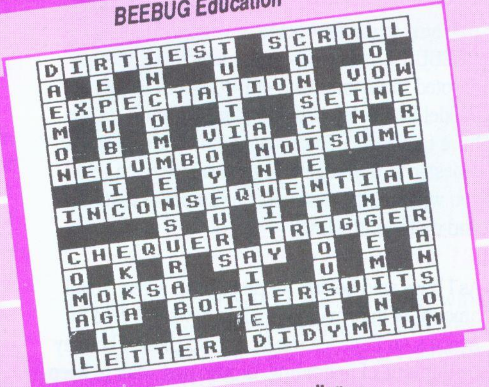
```
Mike Williams
 1  2 3 4 5 6 7 8   9 10 11 12 13 14
1
2
3
4
5
Distinguish between odd and even
numbers
```

BEEBUG Education



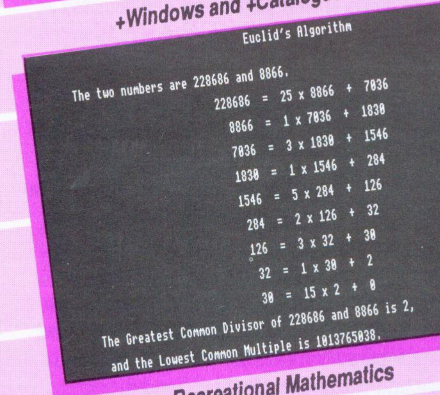
Product	+Windows +Catalogue
Supplier	Panda Discs, Four Seasons, Tinkers Lane, Brewood, Stafford ST19 9DE.
Price	£9.95 each inc. UK P&P

+Windows and +Catalogue



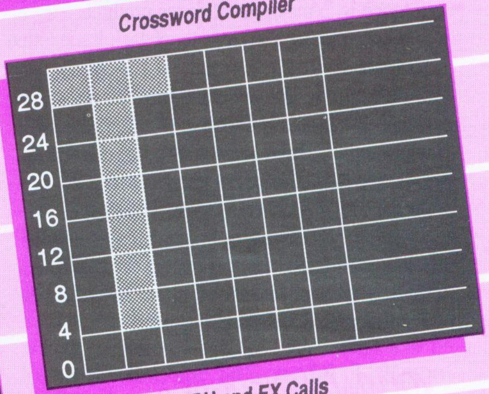
```
D I R T I E S T # S C R O L L
A # E # N # # U # # O # # O #
E X P E C T A T I O N # V O W
M # U # O # # T # # S E I N E
O # B # M # V I A # C # N # R
N E L U M B O # N O I S O M E
# # I # E # Y # N # E # # # #
I N C O N S E Q U E N T I A L
# # # # S # U # I # T # N # #
C H E Q U E R # T R I G G E R
O # K # R # S A Y # O # E # A
M O K S A # # I # # U # M # N
A G A # B O I L E R S U I T S
# L # # L # # E # # L # N # O
L E T T E R # D I D Y M I U M
```

Crossword Compiler



```
Euclid's Algorithm

The two numbers are 228686 and 8866.
          228686  =  25 x 8866  +  7036
            8866  =  1 x 7036  +  1830
            7036  =  3 x 1830  +  1546
            1830  =  1 x 1546  +  284
            1546  =  5 x 284  +  126
             284  =  2 x 126  +  32
             126  =  3 x 32  +  30
              32  =  1 x 30  +  2
              30  =  15 x 2  +  0
The Greatest Common Divisor of 228686 and 8866 is 2,
and the Lowest Common Multiple is 1013765038.
```

Recreational Mathematics



VDU and FX Calls

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

VOLUME TEN

Welcome to the first issue of Volume 10 of BEEBUG, the only magazine which is entirely devoted to the interests of BBC micro users (Model B, Master 128 and Master Compact). We have taken the opportunity to introduce a minor redesign of our page layouts, and we hope that you will find the new look both more readable and more interesting.

As I have said before, we do publish a very small amount of information (mainly news) relating to Acorn's Archimedes range, to help those who may be thinking of upgrading, or who just wish to keep in touch with major Archimedes product announcements. The majority of each issue (often 100%) is devoted to the model B and Master range of BBC micros. Owners of the model B should be aware that probably over half of all current BBC micro owners use a Master. Although we always seek to ensure that as much as possible is relevant to all machines, some articles (and programs) will pertain only to the Master, and some, indeed, are relevant only to the model B. We have to cater for all BBC micro users in order to maintain a flourishing magazine.

VOLUME NINE INDEX

To mark the fact that this is the first issue of Volume 10, we are including two extra items with your magazine at no extra cost. As is always the case with the first issue of a new BEEBUG volume, we have included a printed index to the whole of Volume 9 of BEEBUG to provide a handy reference. Remember, though, that our updated computerised index, *MagScan*, which we relaunched last month, provides even faster and more efficient access to information via keyword searching. And, of course, all volumes can be accessed just as easily in a single search.

FREE PLANNING SHEET

We are also including a free screen planning sheet, relating to the current article in our *First Course* series. This deals with some of the problems which arise when trying to design screen layouts incorporating both text and graphics, and the planning sheet greatly helps in this task by including both text and graphics measurements on the same sheet. Please feel free to photocopy this sheet if required; we will not be keeping any stock for general sale.

NO VAT PRICE INCREASES

As most BEEBUG readers will remember, the Chancellor of the Exchequer raised the rate of VAT from 15% to 17.5% in his budget this year. However, there will be no consequent increase in BEEBUG subscription rates (as these are zero rated anyway), and we have decided to absorb the additional VAT on BEEBUG magazine discs, and all other BEEBUG magazine products, the prices of which remain unchanged. However, we have separately had to increase our charges for post and packing on all orders, and the new rates, where appropriate, are given in this issue.

We certainly expect that throughout 1991 and beyond BEEBUG will continue to offer the best possible support for all BBC micro owners.

M.W.

BEEBUG News BEEBUG News BEEBUG News BEEBUG News

ACORN HOLDS PRICES ON LEARNING CURVE

Despite the increase of VAT from 15% to 17.5% imposed by the Chancellor in his budget, Acorn has announced that it will maintain the existing price on its Learning Curve offers by reducing the ex-VAT price. In addition Acorn is introducing a 0% financing package for its Learning Curve systems, which gives buyers the opportunity to spread the cost over 12 months.

The Learning Curve consists of either an Archimedes A420/1 with 20 MByte hard disc, or an A3000 with floppy disc drive, together with supporting software. The VAT-inclusive prices of the A3000 Learning Curve will remain at £799, and that of the A420/1 learning Curve at £1489.

Full details on both Learning Curve packages are available through Acorn dealers, including BEEBUG.

Acorn has also confirmed that it is developing an enhanced version of its PC emulator for the Archimedes range. No further details or price are as yet available, but release date is expected to be within the next three months.

CONTROLLING YOUR BEEB

Paul Fray Ltd is a specialist company which produces a CMOS Eurocard version of the Acorn Master 128 microcomputer. The card is, of course minus the Master 128 keyboard, and monitor screen. One version, called *Beetle 1*, has 32K of RAM, plus additional 4K and 8K chunks of workspace, and is supplied with Basic IV, DFS and ADFS under license from Acorn. *Beetle 2* will support up to 128K of RAM including shadow and sideways RAM. Both versions of the Eurocard are intended for a wide variety of control applications, and various other interfaces and control systems are available from the same source.

Other products include *Spider*, an add-on in both model B and Master versions providing real-time control for both of these BBC micros, and Termite which greatly increases the number of control inputs available to any BBC micro.

Termite control system on a Master 128

Latest development is the *Arachnid*, which provides a real-time programming environment for the Acorn Archimedes range of computers. For full details of the complete range of products, and prices, contact Paul Fray Ltd., 4 Flint Lane, Ely Road, Waterbeach, Cambridge CB5 9QZ, tel. (0223) 441134.

ANOTHER ADVENTURE WITH TOPOLOGIKA

After concentrating more recently on its educational software, Topologika has returned to the adventure trail by announcing that it will be releasing a brand new adventure game (in the autumn): the long-promised *Spysnatcher* written by Jon Thackray and Jonathan Partingdon, to be priced around £14.95 inc. VAT. This will be available for the BBC model B, Master, Archimedes and various other home micros.

For more information contact Topologika at P.O.Box 39, Stilton, Peterborough PE7 3RL, tel/fax. (0733) 244682.

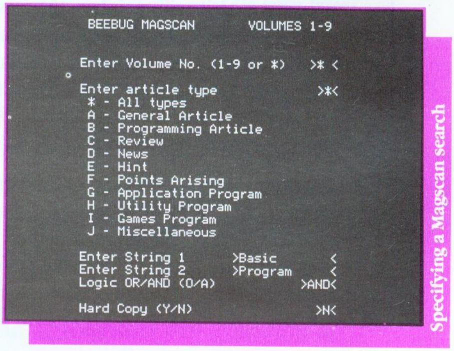
RESOURCING A PHONE NUMBER

Readers should note that the phone number quoted in *News* (Vol.9 No.9) for Doncaster-based Resource is in fact its fax number. The correct numbers are voice: (0302) 320331, fax. (0302) 328735. B

Magscan

Comprehensive Magazine Database for the BBC Micro and the Master 128

An updated version of Magscan, which contains the complete indexes to all BEEBUG magazines from Volume 1 Issue 1 to the latest Volume 9 Issue 10



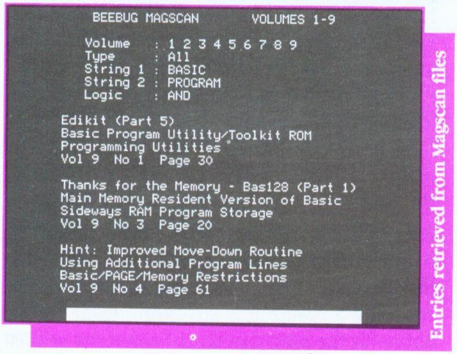
```
        BEEBUG MAGSCAN        VOLUMES 1-9

  Enter Volume No. (1-9 or *)     >* <

  Enter article type                >*<
   * - All types
   A - General Article
   B - Programming Article
   C - Review
   D - News
   E - Hint
   F - Points Arising
   G - Application Program
   H - Utility Program
   I - Games Program
   J - Miscellaneous

  Enter String 1       >Basic      <
  Enter String 2       >Program    <
  Logic OR/AND (O/A)             >AND<

  Hard Copy (Y/N)                   >N<
```

Specifying a Magscan search

Magscan allows you to locate instantly all references to any chosen subject mentioned anywhere in the 90 issues of BEEBUG magazine.

Just type in one or two descriptive words (using AND/OR logic), and you can find any article or program you need, together with a brief description and reference to the volume, issue and page numbers. You can also perform a search by article type and/or volume number.

The Magscan database can be easily updated to include future magazines. Annual updates are available from BEEBUG for existing Magscan users.



```
     BEEBUG MAGSCAN        VOLUMES 1-9

     Volume    : 1 2 3 4 5 6 7 8 9
     Type      : All
     String 1  : BASIC
     String 2  : PROGRAM
     Logic     : AND

  Edikit (Part 5)
  Basic Program Utility/Toolkit ROM
  Programming Utilities
  Vol 9  No 1  Page 30

  Thanks for the Memory - Bas128 (Part 1)
  Main Memory Resident Version of Basic
  Sideways RAM Program Storage
  Vol 9  No 3  Page 20

  Hint: Improved Move-Down Routine
  Using Additional Program Lines
  Basic/PAGE/Memory Restrictions
  Vol 9  No 4  Page 61
```

Entries retrieved from Magscan files

Some of the features Magscan offers include:

- full access to all BEEBUG magazines
- rapid keyword search
- flexible search by volume number, article type and up to two keywords
- keyword entry with selectable AND/OR logic
- extensive on-screen help
- hard copy option
- easily updatable to include future magazines
- yearly updates available from BEEBUG

Phone your order now on (0727) 40303

or send your cheque/postal order to the address below. Please quote your name and membership number. When ordering by Access, Visa or Connect, please quote your card number and the expiry date.

Magscan complete pack, contains disc, manual and release notes: **£9.95**+p&p

Stock codes:	0005b	5.25"disc 40 track DFS		1457b	3.5" ADFS disc
	0006b	5.25"disc 80 track DFS			

Magscan update, contains disc and release notes: **£4.75** +p&p

(for update, please return old disc, label or evidence of purchase)

Stock codes:	0011a	5.25"disc 40 track DFS		1458a	3.5" ADFS disc
	0010a	5.25"disc 80 track DFS			

Postage: **a** - 60p UK, £1 Europe + Eire, £2.40 Elswhere **b** - £1.50 UK, £2.50 Europe + Eire, £4.80 Elswhere

BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS. Telephone (0727) 40303 Fax (0727) 860263

Printing Scientific Characters with Word Processors (Part 1)

by Gareth Leyshon

As an A-level physics and chemistry student, I like to keep revision notes on the computer. However, physics requires the use of the Greek alphabet which is not a standard on my printer, and is not easy to input from the keyboard (unless, of course, you have a Master Compact). In chemistry, extensive use of super- and sub-scripts is required, easy to achieve with the right codes on the printer, but not something standard word processors easily display. Furthermore, some printers cannot print super/subscripts in NLQ mode, printing full-height characters instead.

I use a Panasonic KX-P1081 printer, which is almost but not fully Epson compatible. I feel that this is suitable for designing the following applications, as it is a common printer in schools, which is where it is often necessary to print physics and chemistry notes. The technical notes should allow conversion for Epson compatible (and other standards of) printers. For convenience, I'll refer to non-standard characters as *new characters*. While many applications requiring the use of new characters are scientific, these notes apply equally well to any other word processing task where printer UDGs (user-defined graphics) are required.

What are the options? There are scientific text processing packages on the market, but they can be fairly expensive, and I decided to try to implement my own solution. Built in to my Master 128 were View and Edit; I also had Inter-Word, supplied with the computer by BEEBUG. When just a few characters are required, there is often a simple fix: see, for instance, the "plus or minus" character utility (BEEBUG Vol.8 No.6 p.33, also No.9 p.57). The Master carries an expanded character set; a routine for printing characters 128 to 255 and expanding the Beeb to cover it was given in Vol.8 No.9 p.32, with better printer definitions in Vol.9 No.7 p.55).

But I wanted a fairly large set, including some characters which weren't in the Master's expanded set. As a practical maximum, I took 40 characters - the maximum number which a KX-P1081 can hold in its UDG buffer.

INTER-WORD TECHNIQUES

Firstly I turned to Inter-Word, since it is the word processor I am most familiar with. The simplest way to define non-standard characters is to redefine keyboard characters which you know you won't need for the current document - $; @ and \ are seldom used, for instance. There is no problem with redefining them on the printer, but Inter-Word seems to have its own integral screen character set which can't be redefined. This results in your UDGs appearing as $, @, \, etc., despite printing correctly on paper. The method is suitable as long as you are only using a few new characters, since you can remember what each substitute is.

How do you make sure that the correct definition is sent to the printer when you print out the text? The obvious method is to use an embedded command. Usually, Inter-Word is quite flexible about embedded commands, allowing you to insert a long sequence. But since printers require a number of codes (typically 9 or 11) for each character, you might not be able to fit the definition into the bytes available for a single embedded command. This rules out the definition being put in the "Pre-print" menu option.

There is a simple way round it, however. The character you wish to redefine is obviously a printable character, and frequently some of the other numbers in the definition sequence also correspond to printable characters. This allows longer code sequences to be embedded: the method is best described by means of an example.

Suppose you wish to redefine the @ symbol using the character matrix code sequence 1, 2, 3, 4, 65, 4, 3, 2, 1. Note that the 65 in the definition sequence happens to correspond to the ASCII code for the letter A. The P1081 code to set a UDG is ESC, "y", character, matrix codes.

To program this, you must arrange the top line (which will be printed as a blank line) as follows:

* Leave the first character as a space for clarity.
* Type @, then A, then another space.
* Move to the @, and press f1 to enter an embedded command, namely ESC,"y".
* Now move to the A and embed the sequence 1,2,3,4.
* Finally, move to the trailing space and embed "4,3,2,1".

The printer will receive the visible characters immediately after the embedded codes on top of them and treat them as part of the control sequence.

It is worth mentioning at this point that if you turn off the built-in pound patch (by making the patch character "£" itself), you can still print the pound symbol by typing a hash (#). The advantage of this is that the pound character will actually print a reverse apostrophe (`), which comes in useful if you need to use nested direct speech: "I told you she said,'You can nest direct speech like this` for example." With the pound patch activated, the patch character, if it appears in its own right in the text, is sent to the printer as it is, printing as a pound symbol, rather than being reverse-patched to the computer's ASCII code for pound. N.B. If you actually want to print #, you will have to change your international character set.

You can also check the printer's table of international characters to see if a symbol you require is listed there (it's easier than designing a matrix - the square bracket "]" in the Scandinavian character sets becomes the symbol for the Angstrom unit "Å" (one ten-thousandth of a micron)). Or you might be able to create a character such as a degree sign by using super- or sub-script mode.

Sometimes, selecting Near Letter Quality mode causes complications - for instance, if you try to use super- or sub-scripts in NLQ mode, they will be printed full-height. The trick is to select NLQ mode from software (rather than using the switch on the printer) so that you can include an NLQ turn-off in the same embedded command as the super/subscript. To prevent it looking light compared to the double-pass NLQ, you CAN usually use emphasised mode.

This is an elegant solution if you are prepared to remember what normal-looking characters translate to. I use it sometimes in scientific essays where I only need a few symbols (e.g. °C) and am willing to sacrifice clarity of characters to keep the powerful facilities of Inter-Word. However, it becomes impractical if any great number of characters are required.

CHARACTERS IN VIEW

Like Inter-Word, View cannot display the extended character set. However, if one of the standard characters (i.e. having a code between 31 and 127) is redefined, View (unless in mode seven) WILL show the screen character in its modified form. By not implementing any pound-patch, you can obtain the reverse apostrophe as before. The same general tips apply, but you do not have the same embedded command flexibility.

DESIGNING CHARACTERS

Designing screen characters is a simple process - many programs exist to help you do this (e.g. CHARDES on the Master Welcome disc). However, printer characters do not use the same 8x8 matrix, and usually need to be programmed column by column rather than row by row. In this case, listing 1 may prove useful.

The program, *GridPrint*, allows any character matrix up to size 9x12 (9 rows, 12 columns). After telling it the size of the matrix, you may move around the grid using Z, X, : and /. Space alternately fills and blanks a square.

The bottom row and the right-hand side column are given the value 1, the next-to-bottom and next-to-right 2, then 4, 8, 16 and so on. The total of the values for the filled squares is displayed to the left of each row and below each column. The values shown on screen are hexadecimal (necessary because there is no room to print four-digit decimal values) but when the values are printed, they appear on paper in both decimal and hex.

A miniature version of the display with one pixel per square is shown at the top right of the screen. This gives a better idea of what the final character will look like.

The other key-commands are:

C: Clear matrix.
M: Clear matrix, set up matrix of new dimensions.
Q: Quit program.
P: Print out codes for this character.
D: Call screen dump.

* The screen dump is not installed as standard - insert your own routine at line 1540 onwards.

Another short program, *Analysis*, is provided (listing 2) to allow you to determine the definition codes of any character (default or UDG) resident in the computer at the time. All the possible characters are displayed on screen and can be input using the cursor keys, or a key may be pressed directly. When four codes have been analysed, the screen will be full and the computer will bleep; press any key to clear the screen and continue.

Analysis will work on any Beeb; GridPrint will only run as it stands on a Master but may be converted for BBC micro use by

a) replacing the | in VDU commands with eight zeros;
b) replacing PLOT 149 and PLOT 101 with PLOT 5; and
c) changing the ON...PROC to ON... GOSUB, deleting the relevant DEFPROCs and changing the ENDPROCs to RETURNs or use a series of IF condition THEN PROCs etc.

SUMMARY

The programs above should be useful when designing any sort of printer UDG, whichever word processor you intend to use it with. However, if you have to write a text containing many UDGs, you will soon run out of substitution characters. Or you may find that switching NLQ on and off for a large number of super/subscripts is impractical. Coping with these situations requires a different approach. With this in mind I turned to the Master's Edit ASCII editor and developed a program to allow this to be used as a more flexible word-processor, which we'll look at next time.

Useful BEEBUG references:

- **Plus-or-minus character** *(Vol.8 No.6 p.33)*
- **Plus or minus by redefinition** *(Vol.8 No.9 p.57)*
- **Printing 128-255** *(Vol.8 No.9 p.33)*
- **Pound intercept** *(Vol.9 No.5 p.61)*
- **Improved printer matrix codes** *(Vol.9 No.7 p.55)*

Listing 1

```
   10 REM Program GridPrint
   20 REM Version A1.4/M128
   30 REM Author  Gareth Leyshon
   40 REM BEEBUG  May 1991
   50 REM Program subject to copyright
   60 :
  100 MODE 129:VDU 19,2,15|23,1,1|
  110 ON ERROR GOTO 2000
  120 REPEAT
  130 INPUT "Number of rows:"R%'"Number
of columns:"C%
  140 IF R%>0 AND R%<10 AND C%>0 AND C%<
13 THEN E%=TRUE ELSE E%=FALSE:COLOUR1:CO
LOUR131:PRINT''"MATRIX MUST NOT EXCEED 9
 x 12 !"''CHR$7:COLOUR3:COLOUR128
  150 UNTIL E%
  160 VDU 12,23,1|:*FX 4,2
  170 X%=C%-1:Y%=R%-1
  180 DIM A%(X%,Y%)
  190 H%=-50:W%=50+100*C%:GCOL 0,3
  200 FOR Z%=0 TO R%
  210 H%=H%+100
  220 MOVE 50,H%:DRAW W%,H%
  230 NEXT
  240 W%=-50:H%=50+100*R%
  250 FOR Z%=0 TO C%
  260 W%=W%+100
  270 MOVE W%,50:DRAW W%,H%
  280 NEXT
  290 U%=0:V%=0:PROCcursor:PROCzero
  300 REPEAT
  310 G%=INSTR("ZX:/ QMCDP@zx*? qmcdp",G
ET$)MOD11
  320 ON G% PROCmove,PROCmove,PROCmove,P
ROCmove,PROCflip,PROCquit,PROCnewm,PROCn
ewc,PROCdump,PROCprint ELSE VDU 7
  330 UNTIL SGN(G%)=-1
  340 IF G%=-3 THEN GOTO 160
  350 IF G%=-1 THEN RUN
  360 MODE 7:OSCLI("FX 4"):END
  370 :
 1000 DEF PROCquit:IF FNveri("Quit?") TH
EN G%=-2
 1010 ENDPROC
 1020 :
 1030 DEF PROCnewm:IF FNveri("New matrix
, new size?") THEN G%=-1
 1040 ENDPROC
 1050 :
 1060 DEF PROCnewc:IF FNveri("New matrix
, same size?") THEN G%=-3
 1070 ENDPROC
 1080 :
 1090 DEF FNveri(S$):VDU 7
 1100 COLOUR0:COLOUR131:PRINT TAB(1,0)CH
R$32;S$;" Y/N "
 1110 REPEAT:Q%=INSTR("YyNn",GET$):UNTIL
 Q%>0
 1120 COLOUR3:COLOUR128:PRINTTAB(0,0)SPC
(32):=(Q%<3)
 1130 :
 1140 DEF PROCcursor:GCOL 3,2
 1150 MOVE (C%-U%)*100,(1+V%)*100
 1160 PLOT 149,(C%-U%)*100,(1+V%)*100+25
 1170 GCOL 0,3:ENDPROC
 1180 :
 1190 DEF PROCmove:PROCcursor:U%=(C%+(U%
+(G%=2)-(G%=1)))MODC%
 1200 V%=(R%+(V%+(G%=4)-(G%=3)))MODR%:PR
OCcursor:ENDPROC
 1210 :
 1220 DEF PROCflip:GCOL 3,1:MOVE (C%-U%)
*100-40,(1+V%)*100-40
 1230 PLOT 101,(C%-U%)*100+40,(1+V%)*100
+36
 1240 GCOL 3,3:PLOT 69,1150-4*U%,980+4*V
%:GCOL 0,3
 1250 A%(U%,V%)=A%(U%,V%) EOR 1:PROCtotu
p:ENDPROC
 1260 :
 1270 DEF PROCzero:VDU 5:GCOL 0,0
 1280 FOR L%=1 TO C%:MOVE 100*L%+46,36:G
COL 0,129+2*(L%MOD2):PRINTCHR$127CHR$127
CHR$127"000":NEXT
 1290 FOR L%=1 TO R%:MOVE 36,100*L%+46:G
COL 0,129+2*(L%MOD2):PRINTCHR$127"0":MOV
E 36,100*L%+14:PRINTCHR$127"0":MOVE 36,1
00*L%-18:PRINTCHR$127"0":NEXT
 1300 VDU 4,23,1|:GCOL 0,3:GCOL 0,128:EN
DPROC
 1310 :
 1320 DEF PROCtotup:T%=0:FOR L%=0 TO R%-
1:T%=T%+A%(U%,L%)*(2^L%):NEXT
 1330 VDU 5:GCOL 0,0:L%=C%-U%:MOVE 100*L
%+46,36
 1340 GCOL 0,129+2*(L%MOD2):PRINTCHR$127
CHR$127CHR$127FNtrio
 1350 T%=0:FOR L%=0 TO C%-1:T%=T%+A%(L%,
V%)*(2^L%):NEXT
 1360 L%=V%+1:MOVE 36,100*L%+46:GCOL 0,1
29+2*(L%MOD2):PRINTCHR$127;LEFT$(FNtrio,
1)
 1370 MOVE 36,100*L%+14:PRINTCHR$127MID$
(FNtrio,2,1):MOVE 36,100*L%-18:PRINTCHR$
127RIGHT$(FNtrio,1)
 1380 VDU 4,23,1|:GCOL 0,3:GCOL 0,128:EN
DPROC
 1390 :
 1400 DEF FNtrio:T$=STR$~T%:=STRING$(3-L
EN(T$),"0")+T$
 1410 :
 1420 DEF PROCprint:COLOUR1:PRINTTAB(1,0
)"(P)rint or (A)bort?":COLOUR3
 1430 REPEAT:I%=INSTR("AapP",GET$):UNTIL
I%>0:IF I%>2 THEN PROCtype
 1440 PRINTTAB(0,0)SPC(25):ENDPROC
 1450 :
 1460 DEF PROCtype:VDU 2,21:PRINT"Charac
ter codes:"''"Codes for rows, from top t
o bottom."
 1470 FOR P%=R%-1 TO 0 STEP -1:T%=0:R$="
":FOR L%=0 TO C%-1:R$=CHR$(32+10*A%(L%,P
%))+R$:T%=T%+A%(L%,P%)*(2^L%):NEXT
 1480 PRINTCHR$38;FNtrio;CHR$32;CHR$(-32
*(T%<1000));CHR$(-32*(T%<100));CHR$(-32*
(T%<10));T%;TAB(12,VPOS)SPC(6);R$:NEXT
 1490 PRINT'"Codes for columns, from lef
t to right.":FOR P%=C%-1 TO 0 STEP -1:T%
=0:FOR L%=0 TO R%-1:T%=T%+A%(P%,L%)*(2^L
%):NEXT
 1500 PRINTCHR$38;FNtrio;CHR$32;CHR$(-32
*(T%<1000));CHR$(-32*(T%<100));CHR$(-32*
(T%<10));T%:NEXT
 1510 PRINT'':VDU 6,3:ENDPROC
 1520 :
 1530 DEF PROCdump
 1540 REM Put your dump routine here.
 1550 ENDPROC
 1560 :
 2000 REM Error handling routine
 2010 MODE 135
 2020 IF ERR<>17 THEN REPORT:PRINT" at l
ine ";ERL''
 2030 END
```

Listing 2

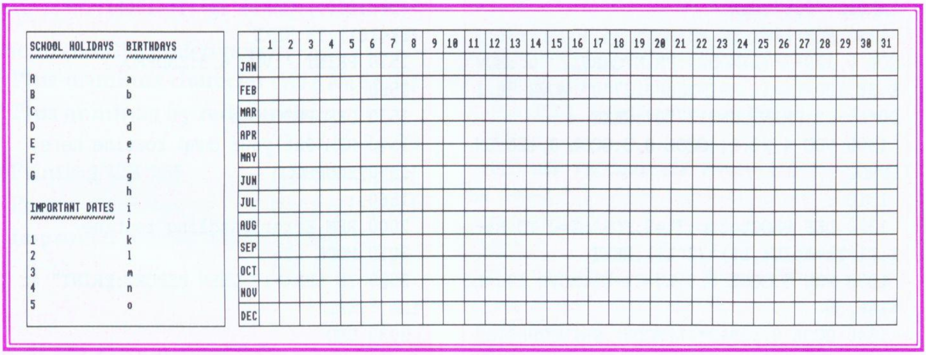
```
   10 REM Program Analysis
   20 REM Version B1.1
   30 REM Author  Gareth Leyshon
   40 REM BEEBUG  May 1991
   50 REM Program subject to copyright
   60 :
  100 REPEAT
  110 MODE3:FOR Z%=32 TO 255:VDU 32,Z%:N
EXT:PRINT
  120 REPEAT
  130 ?&70=GET:VDU 10,?&70,10,13
  140 X%=&70:Y%=0:A%=10:CALL &FFF1
  150 PRINT ?&71,?&72,?&73,?&74,?&75,?&7
6,?&77,?&78
  160 UNTIL VPOS>21:VDU 7:IF GET
  170 UNTIL FALSE
```

Mini Wall Planner

Organise your life with this useful program from Graham Cooley.

No company office would be complete without a year planner on the wall. Now you can print out your own personal year planner using the program *WalPlan* listed here. Significant dates, such as birthdays, school holidays etc., can be picked out on an index which is printed alongside the planner (see Figure 1).

Type in the listing carefully and save it as *WalPlan*. If you wish to incorporate significant dates, you must replace the existing DATA statements in lines 1850-1870, and if necessary the headings at lines 1180-1230. In both cases, the maximum number of characters for each entry is 15 (or less if you require space for a marker - see below).



```
SCHOOL HOLIDAYS  BIRTHDAYS

A                a
B                b
C                c
D                d
E                e
F                f
G                g
                 h
IMPORTANT DATES  i
                 j
1                k
2                l
3                m
4                n
5                o
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
JAN																															
FEB																															
MAR																															
APR																															
MAY																															
JUN																															
JUL																															
AUG																															
SEP																															
OCT																															
NOV																															
DEC																															

Printout produced by the Mini Wall Planner

RUNNING THE PROGRAM

When run, the program outputs the complete wall planner onto any Epson-compatible printer, using two sheets of continuous paper. Any items you have added, as described above, will be included in the index on the left. The program works by first drawing the design on the screen in three sections, and dumping these to the printer one by one.

USING THE PLANNER

Everyone will have his or her own ideas on how to use the wall planner, but one suggestion is to use colour-coded small self-adhesive labels (available from any good stationer). Place one of these beside the index entry and another of the same colour in the appropriate slot on the date chart. Alternatively you could use coloured pens or markers. You never need be disorganised again!

```
  10 REM         >WalPlan
  20 REM Program Mini Wall Planner
  30 REM Version B1.0
  40 REM Author  Graham Cooley
  50 REM BEEBUG  May 1991
  60 REM Program subject to copyright
  70 :
 100 ON ERROR PROCerror
```

```
  110 MODE 7
  120 VDU23,1,0;0;0;0;
  130 PROCtitle("MINI WALL PLANNER")
  140 PROCtitle("By Graham Cooley")
  150 PRINTTAB(0,8)"PLEASE AMEND THE DAT
A STATEMENTS AT THE END OF THE PROGRAM T
O SUIT YOUR NEEDS."
  160 PRINTTAB(10,15)"PRESS THE SPACEBAR"
  170 REPEAT UNTIL INKEY(-99)
  180 :
  190 MODE 4:VDU5,2:*FX3,64
  200 VDU23,1,0;0;0;0;
  210 DIM A$(12),H$(7),I$(5),N$(15)
  220 :
  230 PROCbox:PROCinfo:PROCprint
  240 CLS:PROCgrid(1160)
  250 PROCday(1,10,-120):PROCmonth
  260 PROCprint:CLS
  270 :
  280 PROCgrid(1160)
  290 PROCday(11,21,-236)
  300 PROCprint:CLS
  310 :
  320 PROCgrid(1044)
  330 MOVE 1160,70:DRAW 1160,980
  340 PROCday(22,31,-236)
  350 PROCprint:CLS
  360 VDU3
  370 END
  380 :
 1000 DEF PROCerror
 1010 VDU3
 1020 REPORT:PRINT" at line ";ERL:END
 1030 ENDPROC
 1040 :
 1050 DEF PROCtitle(t$)
 1060 A%=LEN(t$)/2
 1070 PRINTCHR$132;CHR$157;CHR$131;CHR$1
41;TAB(20-A%);t$
 1080 PRINTCHR$132;CHR$157;CHR$131;CHR$1
41;TAB(20-A%);t$
 1090 ENDPROC
 1100 :
 1110 DEF PROCbox
 1120 MOVE 80,70:DRAW 80,980
```

```
1130 DRAW 1200,980:DRAW 1200,70
1140 DRAW 80,70
1150 ENDPROC
1160 :
1170 DEF PROCinfo
1180 MOVE 100,950:PRINT"SCHOOL HOLIDAYS"
1190 MOVE 100,910:PRINT"~~~~~~~~~~~~~~~"
1200 MOVE 650,950:PRINT"BIRTHDAYS"
1210 MOVE 650,910:PRINT"~~~~~~~~~"
1220 MOVE 100,450:PRINT"IMPORTANT DATES"
1230 MOVE 100,410:PRINT"~~~~~~~~~~~~~~~"
1240 RESTORE 1850
1250 FOR chol=850 TO 550 STEP-50
1260 READ H$
1270 MOVE 100,chol:PRINT H$
1280 NEXT
1290 RESTORE 1860
1300 FOR icol=350 TO 150 STEP -50
1310 READ I$
1320 MOVE 100,icol:PRINT I$
1330 NEXT
1340 RESTORE 1870
1350 FOR ncol=850 TO 150 STEP -50
1360 READ N$
1370 MOVE 650,ncol:PRINT N$
1380 NEXT
1390 ENDPROC
1400 :
1410 DEFPROCmonth
1420 RESTORE 1810
1430 FOR col=885 TO 115 STEP -70
1440 READ A$
1450 MOVE 12,col:PRINTA$
1460 NEXT
1470 ENDPROC
1480 :
1490 DEF PROCday(S,F,Q)
1500 FOR L=S TO F
1510 MOVE Q,955:PRINT L
1520 Q=Q+116
1530 NEXT
1540 ENDPROC
1550 :
1560 DEF PROCgrid(x)
```

Continued on page 42

+Windows and +Catalogue

Reviewed by Ian Waugh

Product	+Windows +Catalogue
Supplier	Panda Discs, Four Seasons, Tinkers Lane, Brewood, Stafford ST19 9DE.
Price	£9.95 each inc. UK P&P

You can't keep a good word processor down. After seven years, Wordwise Plus is still arguably the most popular word processor on the BBC (no matter what View addicts say!), and companies are still producing support programs for it.

Two of the latest come from Panda Discs which is probably better known as the software arm of AMPLE DCT offering on-line support for the Hybrid Music System. Panda branched out a while ago with a crossword puzzle program (see BEEBUG Vol.9 No.3) and these two new programs further extend its catalogue.

+Windows is a green command embedder for WW+ which uses pull-down windows. It was originally written as an aid for students with special needs, and it has been designed to be particularly easy to use. Panda hopes this will make it appeal to other WW+ users, too.

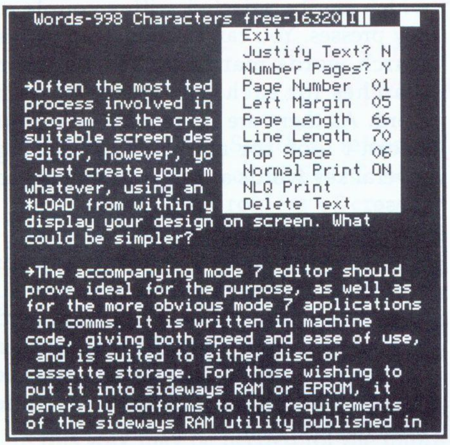
Like many Wordwise Plus support programs, +Windows makes use of the Wordwise Plus programming language, WPPL. The routines are supplied on a master disc which you are advised to copy in case you want to edit the files. On booting (pressing Shift-Break) you enter Wordwise Plus and the routines load into segments one to five. There are 10 files, so if you use DFS and decide to copy the files to your work disc you will lose a third of your disc's file capacity. A better solution is to create a boot disc since once the routines are installed you no longer need access to the files.

The routines are called, as usual, by pressing Shift and one of the function keys. If you press Shift-Lock, they can be accessed with only one finger. Each routine causes a menu to drop down from the top of the screen. You use the down cursor key to cycle through the options and Return to select one.

The f1 menu has three options. The first two insert a string of embedded commands at the start of the text which set justification, page numbering and page number, page length, line length, left margin and top space. One produces draft quality print, the other final quality. The cursor then moves automatically to the end of the text ready to insert a Begin Page command, the third menu option, if you so wish.

The f2 menu contains Centre, Underline On and Underline Off commands. A nice touch is that after selecting Underline On, the cursor moves to Underline Off when you next call the menu. f3 includes Bold and Italics On and Off commands, and these work in the same way.

f4 houses Line Space, Indent, Temporary Indent and Cancel Indents commands. The first three require a numeric parameter. After selection, the instruction appears in the text at the cursor position. You then enter the number and carry on typing (by pressing any non-numeric key) and the White embedded command end marker is inserted automatically. Nice.



```
 Words-998 Characters free-16320
                         Exit
                         Justify Text? N
                         Number Pages? Y
→Often the most ted      Page Number  01
process involved in      Left Margin  05
program is the crea      Page Length  66
suitable screen des      Line Length  70
editor, however, yo      Top Space    06
 Just create your m      Normal Print ON
whatever, using an       NLQ Print
*LOAD from within y      Delete Text
display your design on screen. What
could be simpler?

→The accompanying mode 7 editor should
prove ideal for the purpose, as well as
for the more obvious mode 7 applications
 in comms. It is written in machine
code, giving both speed and ease of use,
 and is suited to either disc or
cassette storage. For those wishing to
put it into sideways RAM or EPROM, it
generally conforms to the requirements
of the sideways RAM utility published in
```

The +Windows software in operation

f5 allows you to alter the default embedded commands used by f1 for draft and final printing. This is a temporary change but you can make any settings permanent default values by editing the Format file on the disc. This is a text file which can be loaded into and saved from WW+. The basic settings are clearly REMed and easily edited.

You can also alter the codes used to send commands such as underline and double strike to the printer. The defaults are for Epson-compatible printers but, armed with another printer's Escape code sequences, these can be easily changed, too. The Italics On and Off commands can be completely redefined to produce, say, Emphasised or Double Width effects. The new names will appear in the f5 menu.

The program uses OPS (Output Print Sequence) and RPS (Redefine Print Sequence) commands. You can save the embedded command settings with

```
:SAVE PARAMS <filename>
```

and load the settings back into a computer, even one which doesn't have +Windows installed.

+Windows works superbly and users with restricted mobility, for example, will find it very easy to use - one-finger operation is all that's required. However, if you don't have an Epson-compatible printer or wish to change the default settings then you may have to dig into your printer's manual which is one of the things the program was designed to avoid. But once set up it's plain sailing all the way.

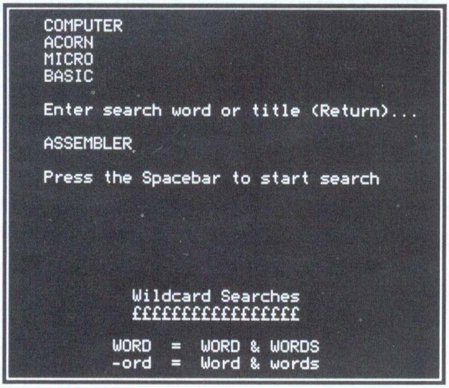
It's worth pointing out that all +Window's functions - and many more besides - can be found in WordWise Plus II (available from IFEL) although operation does require two simultaneous keypresses. If you can manage that, it would undoubtedly be my first recommendation. If you can't, then +Windows is as easy to use a program as you'll get.

+Catalogue is a simple database system designed for users with little or no knowledge of computing or WW+. The word 'database' may be a little ambitious, however, and the program describes itself more modestly as an information and data retrieval system. Suggested applications include the storage of information about record and slide collections, books and videos and even names and addresses. The manual also suggests that teachers may find it useful for storing and searching National Curriculum attainment targets. However, it has several shortcomings which, I'd venture to suggest, do not make it the ideal medium for some of these applications.

The program autoboots from disc and is completely menu driven. It uses WPPL and effectively takes over WW+ although you do have to drop into WW+ to create and edit the data files. It can handle up to

25 files, disc capacity permitting, although you can insert and catalogue another disc at any time.

I'm afraid the manual gets off to a rather poor start by jumping in with an explanation of how to edit the File Menu even before explaining how to use the program. I suspect the complete novice will be somewhat confused, although if you persevere and read through the rest of the manual, all will become clear.



```
COMPUTER
ACORN
MICRO
BASIC

Enter search word or title (Return)...

ASSEMBLER

Press the Spacebar to start search




                Wildcard Searches
                £££££££££££££££££

            WORD  =  WORD & WORDS
            -ord  =  Word & words
```

Performing a search using +Catalogue

Each file must be able to reside in the Beeb's memory. This may appear rather limiting but you should be able to get around 3000 words into a file and if you look upon +Catalogue as a cataloguing system (as its name suggests) and not a fully-fledged database you'll realise that the total storage capacity is considerable.

A file is simply a list of items. There are no separate records or data fields as such although you can separate items (records) in a file by blank lines for your convenience.

Having created your files, you can search one or all for key words. To help with this you can create a Keyword Index file containing commonly used search words which can be selected with a couple of key presses. You can also enter up to five search words manually. The program searches for each occurrence of each word. As there are no records as such it doesn't look for an entry containing several key words as this would make no sense.

You can load any file into memory for searching or you can elect to search all files. When a match is found the word can be optionally highlighted with a colour or simply indicated with the cursor. The search can then continue from the next line.

You can view the contents of the current file and edit it. All searches are case sensitive and the manual suggests that it would be good practice to use upper case only in all files. There is a wildcard facility so you could capitalise the initial letter of keywords and use the wildcard in place of the initial letter. Not an ideal solution, however.

All the function keys are used. When viewing a file, for example, function keys 6 to 9 are used to move around the text. You can toggle the colouring of the 'word found' on and off and send a section of the current file to a printer.

The Break key is used to re-enter +Catalogue after editing a file. This may seem rather dangerous but WW+ is incredibly stable. However, pressing Break may have an adverse effect on some programs lodged in sideways RAM such as printer buffers and spoolers.

Well, that's basically what +Catalogue does. You'll probably have realised that it lacks a sort function. I'd suggest that the

Continued on page 38

View to ASCII

Andrew Rowland explains how to convert View word processing files to a plain ASCII format.

Further to D.J. Smith's letter in December's issue of BEEBUG (Vol.9 No.7) about the problem of creating plain text (ASCII) files from View, there is another way of removing formatting commands etc. The idea is that as View sends plain text to the printer when producing hardcopy, a utility masquerading as a printer driver can spool printer output to disc. Files created in this way may be needed for the multi-column printing program MultCol (see BEEBUG Vol.7 No.1) or for inserting into another word processor or DTP package.

Simply type in the short listing given here. When run the program produces a printer driver called VSpool, which is then saved to disc. Subsequently when using View, the driver should be installed as usual using:

```
        Printer VSpool
```

An output file can then be created by typing:

```
        Print
```

(or PR). It will prompt you for a filename, and you may abort at this stage by pressing Escape.

The advantage of this method is that it is fully automatic, and the only change you may need to make to your View document is to inhibit paging, either by entering the embedded command *PB OFF* at the beginning if you have View 3, or setting top and bottom margins to zero, turning off headers and footers.

The routine suppresses any blank lines at the start of the document, ignores the left margin (including indents) and filters out all printer highlights. It replaces single carriage returns (marking the end of lines) with a space. Multiple carriage returns - the blank lines between paragraphs, for example - are retained. In doing so, I have assumed that any multiple blank lines are deliberate, but if this doesn't suit you, delete line 1630, and all multiples will be replaced by a single blank line.

I have also assumed that multiple spaces (produced by TABs etc.) are undesirable as they result in large gaps opening up in multiple column text. They are all reduced to one space. However, if your purposes are better served by retaining them, delete line 1380.

VSpool is adapted from *V_strip* which was included in the introductory DTP package for foreign languages, *Front Page Europe*, published by Newman Software Ltd. It is reproduced here by kind permission of the publishers.

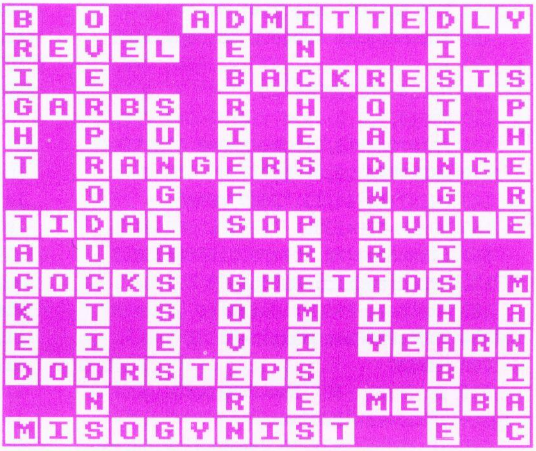
```
   10 REM Program VSpool_B
   20 REM Program View Spooler
   30 REM Version 0.04
   40 REM Install as Printer Driver
   50 REM Author  Andrew Rowland
   60 REM (C) Newman Software Ltd 1989
   70 :
  100 ON ERROR PRINT':REPORT:PRINT" at l
ine ";ERL:END
  110 PROCmc:fn$="VSPOOL"
  120 a$="SAVE "+fn$+" "+STR$~Q%+"+FF 40
```

```
0 400"
 130 PRINT a$:OSCLI a$
 140 VDU 14:END
 150 :
1000 DEF PROCmc
1010 oswrch=&FFEE:osbyte=&FFF4
1020 osasci=&FFE3:osword=&FFF1
1030 osfind=&FFCE:osbput=&FFD4
1040 osnewl=&FFE7:Q%=&900
1050 len%=&44:buffer=&500
1060 FOR pass=4 TO 7 STEP 3
1070 P%=&400:O%=Q%
1080 [OPT pass
1090 .print   JMP char
1100 .on      JMP pron
1110 .off     JMP close
1120 .micro   RTS:NOP:NOP
1130 .option  RTS
1140 :
1150 .pron LDA &70:PHA:LDA &71:PHA
1160 LDX #LEN"Filename? "-1
1170 .loop LDA mess,X:JSR oswrch
1180 DEX:BPL loop
1190 PLA:STA &71:PLA:STA &70
1200 JSR getline:BCC notesc
1210 BRK:EQUB 17:EQUS "Escape":BRK
1220 .notesc
1230 LDX #buffer MOD &100
1240 LDY #buffer DIV &100
1250 LDA #&80:JSR osfind \ OPENOUT
1260 STA handle:CMP #0:BNE openok
1270 BRK:BRK:EQUS "Can't!":BRK
1280 .openok INC startflag
1290 .zerocr LDA #0:STA crflag:RTS
1300 :
1310 .char STA currchar
1320 TXA:PHA:TYA:PHA
1330 LDA currchar
1340 CMP #13:BEQ cr
1350 CMP #32:BNE notsp
1360 LDX startflag:BNE finish
1370 LDX margin:BNE finish \ skip
1380 LDX spflag:BEQ sp:BNE finish
1390 .notsp LDX #0:STX spflag
```

```
1400 CMP #128:BCS finish
1410 CMP #32:BCC finish
1420 LDX #0:STX margin:STX startflag
1430 LDX crflag:BEQ nocr:JSR lines
1440 .nocr LDA currchar:JSR bput
1450 JSR zerocr
1460 .finish PLA:TAY:PLA:TAX
1470 LDA currchar:RTS
1480 :
1490 .cr INC margin
1500 LDA startflag:BNE finish
1510 INC crflag:JMP finish
1520 :
1530 .sp INC spflag:BNE nocr
1540 :
1550 .getline LDX #pblk MOD &100
1560 LDY #pblk DIV &100
1570 LDA #0:JMP osword
1580 :
1590 .lines CPX #2:BCS linov
1600 LDA #32:JMP bput
1610 .linov LDA #13
1620 .linlp JSR bput
1630 DEX:BNE linlp:RTS
1640 :
1650 .bput JSR osasci
1660 LDY handle:JMP osbput
1670 :
1680 .close LDY handle
1690 LDA #0:JSR osfind \ close file
1700 JMP osnewl
1710 :
1720 .mess      EQUS " ?emaneliF"
1730 .crflag    EQUB 0
1740 .startflag EQUB 0
1750 .spflag    EQUB 0
1760 .margin    EQUB 0
1770 .currchar  EQUB 0
1780 .handle    EQUB 0
1790 .pblk      EQUW &500 \ buffer
1800            EQUB &44  \ length
1810            EQUB 32   \ min char
1820            EQUB 126  \ max char
1830 ]NEXT:ENDPROC
```

Crossword Compiler

We present a program by John Webb which enables you to construct your own crosswords with ease.



```
B # O # # A D M I T T E D L Y
R E V E L # E # N # # # I # #
I # E # # # B A C K R E S T S
G A R B S # R # H # O # T # P
H # P # U # I # E # A # I # H
T # R A N G E R S # D U N C E
# # O # G # F # # # W # G # R
T I D A L # S O P # O V U L E
A # U # A # # # R # R # I # #
C O C K S # G H E T T O S # M
K # T # S # O # M # H # H # A
E # I # E # V # I # Y E A R N
D O O R S T E P S # # # B # I
# # N # # # R # E # M E L B A
M I S O G Y N I S T # # E # C
```

The program listed here is undoubtedly lengthy but will appeal to all crossword enthusiasts, as it not only allows you to construct the basic crossword layout, but to include the answers in the crossword as well. Take care when entering the program, and save it before attempting to run it.

When run, the program displays a default crossword grid, with the option to choose a different size if you so wish (in the range 3 to 19 - remember that all good crossword designs use an odd number of squares). With the blank grid on display, you can now enter the black squares (using the Copy key), or remove them with Delete. The cursor keys enable you to move around the grid. The program also preserves the usual crossword symmetry as far as black squares are concerned.

At any time you can enter letters into the blank squares to form the answers to the crossword. Both black squares and letters can be edited at will until you are satisfied. To help, Ctrl-? will delete all the existing words, and Ctrl-@ will clear the grid completely ready to start again. Other operations are controlled by the function keys as follows:

f0 Display list of functions
f1 Load a saved crossword
f2 Save the current crossword
f4 Print options
f5 Show answers
f6 Count squares
f9 Exit from the program

Key f5 calculates the clue numbers and displays this list on the screen, with the words if these have been entered. Key f6 counts the numbers of black and blank squares in the crossword.

There are effectively five print options:

1. The crossword grid only
2. The crossword grid with numbers
3. As currently displayed
4. Final print
5. Answers only

Where a crossword is to be printed you have the choice of a large (full paper width) or small size. Option 6 returns to the main display. The print options assume an Epson compatible Kaga KP810 printer, but PROCprinting (line 3990) and associated procedures can be modified to suit an alternative.

The program is also able to take advantage of Spellmaster if fitted to assist in finding words to fit a given pattern. Use the '*' key, and then enter the sequence with space or '#' for the missing characters. The magazine disc also contains a completed crossword ready to load into the program (for printing, modification, or what you choose).

```
   10 REM         >Crosswd
   20 REM Program Crossword compiler
   30 REM Version B1.10
   40 REM Author  John Webb
   50 REM BEEBUG  May 1991
   60 REM Program subject to copyright
   70 :
  100 MODE4:*FX200,1
  110 VDU19,1,0;0;19,0,7;0;
  120 PRINTTAB(12,7)"CROSSWORD COMPILER"
  130 PRINTTAB(19,9)"by"
  140 PRINTTAB(16,11)"JOHN WEBB"
  150 LOMEM=TOP+1280
  160 DIMspace%(20,20):PROCinitial:CLS
  170 PROCdrawgrid:PROCdump
  180 PROChelp:PROCgridsize
  190 VDU5:PROCmksq:exit%=FALSE
  200 REPEAT
  210 IF INKEY(-58) PROCunmksq:height=he
ight+1:PROCvert:PROCmksq
  220 IF INKEY(-42) PROCunmksq:height=he
ight-1:PROCvert:PROCmksq
  230 IF INKEY(-122) PROCunmksq:side=sid
e+1:PROChoriz:PROCmksq
  240 IF INKEY(-26) PROCunmksq:side=side
-1:PROChoriz:PROCmksq
  250 IF INKEY(-106) PROCfillsquare:GOTO
400
  260 IF INKEY(-90) PROCunfillsquare
  270 IF INKEY(-120) exit%=TRUE
  280 inkey=INKEY(1):IF inkey<>32 inkey=
inkey AND 223
  290 IF (inkey>64 AND inkey<91) OR inke
y=32 THEN ?FNspace(side,height,0)=inkey:
PROCunmksq:PROCmksq
  300 IF INKEY(-33) PROChelp
  310 IF INKEY(-117) PROCpanswers("D")
  320 IF INKEY(-114) PROCload
  330 IF INKEY(-115) PROCsave
  340 IF INKEY(-73) PROCcommand
  350 IF INKEY(-92) PROCcommand
  360 IF INKEY(-21) PROCprinting
  370 IF INKEY(-118) PROCfilledsquares
  380 IF INKEY(-2) AND INKEY(-72) Q%=0:P
ROCclr
  390 IF INKEY(-2) AND INKEY(-105) Q%=1:
PROCclr
  400 UNTIL exit%
```

```
  410 MODE7:*FX200,0
  420 END
  430 :
 1000 DEF PROCdrawgrid
 1010 acrossmax=across+grid*square
 1020 upmax=up+grid*square
 1030 FOR N%=1 TO grid+1
 1040 MOVE across+N%*square,up
 1050 DRAW across+N%*square,up+square*gr
id
 1060 NEXT
 1070 FOR N%=0 TO grid
 1080 MOVE across+square,up+N%*square
 1090 DRAW across+square*(grid+1),up+N%*
square
 1100 NEXT:PROCtitle
 1110 ENDPROC
 1120 :
 1130 DEF PROChoriz
 1140 IF side>grid side=grid
 1150 IF side<1 side=1
 1160 ENDPROC
 1170 :
 1180 DEF PROCvert
 1190 IF height>grid height=grid
 1200 IF height<1 height=1
 1210 ENDPROC
 1220 :
 1230 DEF PROCfillsquare
 1240 ?FNspace(side,height,1)=ASC"F"
 1250 oppside=grid+1-side:oppheight=grid
+1-height
 1260 ?FNspace(oppside,oppheight,1)=ASC"
F"
 1270 IF load=FALSE:IF oppside=side AND
oppheight=height ENDPROC
 1280 PROCunmksq
 1290 MOVE (across+square*oppside)+6,up+
oppheight*square-4:MOVE (across+square*(
oppside+1))-4,up+oppheight*square-4:VDU2
5,81,-square+8;-square+8;25,81,square-8;
0;
 1300 VDU18,0,1:PROCopprint:VDU18,0,3
 1310 PROCmksq:ENDPROC
 1320 :
 1330 DEF PROCunfillsquare
 1340 ?FNspace(side,height,1)=ASC"U"
 1350 oppside=grid+1-side:oppheight=grid
```

```
+1-height
 1360 ?FNspace(oppside,oppheight,1)=ASC"
U"
 1370 PROCprint:PROCopprint:PROCunmksq
 1380 MOVE (across+square*oppside)+6,up+
oppheight*square-4:MOVE (across+square*(
oppside+1))-4,up+oppheight*square-4:VDU2
5,83,-square+8;-square+8;25,83,square-8;
0;
 1390 PROCopprint:PROCmksq
 1400 ENDPROC
 1410 :
 1420 DEF PROCprint
 1430 MOVE across+square*side+(square-24
)/2,up+height*square-(square-24)/2
 1440 IF Q%<>2 PRINTCHR$(?FNspace(side,h
eight,0))ELSE PRINTCHR$32
 1450 ENDPROC
 1460 :
 1470 DEF PROCopprint
 1480 MOVE across+square*oppside+(square
-24)/2,up+oppheight*square-(square-24)/2
 1490 PRINTCHR$(?FNspace(oppside,oppheig
ht,0))
 1500 ENDPROC
 1510 :
 1520 DEF PROCunmksq
 1530 IF ?FNspace(side,height,1)=ASC"F"
MOVE (across+square*side)+2,up+height*sq
uare-2:MOVE (across+square*(side+1))-4,u
p+height*square-2:VDU25,81,-square+6;-sq
uare+6;25,81,square-6;0;:VDU18,0,1:PROCp
rint:VDU18,0,3:ENDPROC
 1540 MOVE (across+4+square*side)+2,up-4
+height*square-2:MOVE (across+4+square*(
side+1))-6,up-4+height*square-2:VDU25,83
,-square+6;-square+8;25,83,square-8;0;
 1550 IF puz=FALSE PROCprint
 1560 ENDPROC
 1570 :
 1580 DEF PROCmksq
 1590 VDU18,3,1
 1600 MOVE (across+square*side)+6,up+hei
ght*square-6:PLOT1,square-10,0:PLOT1,0,1
0-square:PLOT1,10-square,0:PLOT1,0,squar
e-10
 1610 VDU18,0,3:PROCprint:ENDPROC
 1620 :
```

```
 1630 DEF PROCload
 1640 VDU26,4,15,28,0,30,39,1:CLS
 1650 *Cat
 1660 load=TRUE:REPEAT
 1670 VDU4,28,0,31,39,29
 1680 CLS:INPUT'"Load file name: "F$;
 1690 UNTIL LENF$<11
 1700 IF F$="" CLS:GOTO1800
 1710 F=OPENIN(F$)
 1720 IF F=0 CLS:GOTO1800
 1730 CLS:PRINT"Loading ";F$;", please w
ait a moment"CHR$7;CHR$5
 1740 PROCinitial
 1750 INPUT#F,square,grid,side,height
 1760 FOR P%=0 TO 1:FOR N%=1 TO grid:FOR
 M%=1 TO grid
 1770 INPUT#F,Q$:?FNspace(N%,M%,P%)=ASC(
Q$)
 1780 NEXT:NEXT:NEXT
 1790 CLOSE#0:load=FALSE:name$=F$
 1800 PROCdisplay:ENDPROC
 1810 :
 1820 DEF PROCdisplay
 1830 VDU5,26:CLS:PROCdrawgrid
 1840 DEF PROCpartdisplay:VDU5
 1850 FOR side=1 TO grid:FOR height=1 TO
 grid
 1860 IF puz=FALSE IF (?FNspace(side,hei
ght,0)<>32 AND ?FNspace(side,height,1)=A
SC"U")PROCprint ELSE PROCunmksq
 1870 NEXT:NEXT
 1880 side=1:height=1:PROChelp
 1890 IF puz=FALSE PROCmksq
 1900 ENDPROC
 1910 :
 1920 DEF PROCsave
 1930 REPEAT:VDU4,15,28,0,31,39,31
 1940 INPUT'"Save file name: "F$
 1950 UNTIL LENF$<11
 1960 VDU5:IF F$=""ENDPROC
 1970 F=OPENOUT(F$)
 1980 PRINT#F,square,grid,side,height
 1990 FOR P%=0 TO 1:FOR N%=1 TO grid:FOR
 M%=1 TO grid
 2000 PRINT#F,CHR$(?FNspace(N%,M%,P%))
 2010 NEXT:NEXT:NEXT
 2020 CLOSE#0:name$=F$
 2030 VDU4:PROCtitle:VDU5
 2040 ENDPROC
 2050 :
 2060 DEF PROCtitle
 2070 VDU4,26
 2080 line=up/(32*2):len=LENname$
 2090 div=32:IF puz=TRUE div=16
 2100 len=((grid*square)/div-len)/2+acro
ss/div:IF len<0 len=0
 2110 PRINTTAB(0,line)SPC20;
 2120 PRINTTAB(len+1.6,line)name$
 2130 VDU5:ENDPROC
 2140 :
 2150 DEF PROCcommand
 2160 PROCpos:PRINT"Search$ :"
 2170 TIME=0:REPEAT:UNTIL TIME>15
 2180 *FX21,0
 2190 A$="":REPEAT:get$=GET$
 2200 IF get$=" "ORget$="#" A$=A$+"#":GO
TO2260
 2210 IF ASCget$=13 THEN 2260
 2220 M=223 AND ASCget$:get$=CHR$M
 2230 IF M=95:A$=LEFT$(A$,LENA$-1)
 2240 *FX21,5
 2250 IF M>64 AND M<91 A$=A$+get$ ELSE V
DU7
 2260 PRINTTAB(0,2)A$;" "
 2270 UNTIL ASCget$=13:PRINT:VDU7
 2280 IF A$="" CLS:GOTO2330
 2290 ON ERROR PRINT"None Found":GOTO190
 2300 OSCLI"CROSS."+A$
 2310 ON ERROR OFF
 2320 VDU15:PRINT'"Thats all"
 2330 VDU5:ENDPROC
 2340 :
 2350 DEF PROCinitial
 2360 VDU23;11,0;0;0;0:*FX211,1
 2370 Q%=0:grid=15
 2380 FOR N%=65 TO 1270 STEP3:N%!TOP=&00
5520:NEXT
 2390 *FX212,0
 2400 name$="Crossword Compiler":*FX213,
94
 2410 square=48
 2420 side=1:height=1:load=FALSE:puz=FAL
SE:both=FALSE:*FX214,12
 2430 across=20:up=(1023-(square*grid))/
2:ENVELOPE1,3,0,0,0,0,0,0,121,-10,-5,-62
,78,78
```

```
 2440 ENDPROC
 2450 :
 2460 DEF PROChelp
 2470 PROCpos
 2480 VDU4,14,28,V,31,39,3
 2490 VDU23,33,254,254,254,254,254,254,2
54,0,23,35,254,130,130,130,130,130,254,0
 2500 PRINT"  Help"'
 2510 PRINTTAB(0,3)"f0-Help"
 2520 PRINTTAB(0,4)"f1-Load"
 2530 PRINTTAB(0,5)"f2-Save"
 2540 PRINTTAB(0,6)"f4-Prints"
 2550 PRINTTAB(0,7)"f5-Show"'"   answers
"
 2560 PRINTTAB(0,9)"f6-No.!:#"
 2570 PRINTTAB(0,11)"* -Search"
 2580 PRINTTAB(0,12)"f9-Exit"
 2590 PRINTTAB(0,14)"Cursor keys-"'"   M
ove"
 2600 PRINTTAB(0,16)"Copy  -!"
 2610 PRINTTAB(0,17)"Delete-#"
 2620 PRINTTAB(0,19)"Ctrl @"'"  Refresh"
 2630 PRINTTAB(0,21)"Ctrl ?"'"  Delete"'
"  words"
 2640 VDU5:ENDPROC
 2650 :
 2660 DEF PROCgridsize
 2670 VDU4,15,28,0,31,39,31
 2680 COLOUR1:PRINT"Change grid size? (Y
/N)";:COLOUR3:VDU7:*FX21,0
 2690 REPEAT:get=GET AND 223
 2700 IF get<>89 CLS:ENDPROC
 2710 REPEAT:CLS:INPUT"New grid size: "g
rid:grid=INT(grid):UNTIL grid MOD2=1 AND
 grid>2 AND grid<20
 2720 up=(1023-(square*grid))/2
 2730 CLS:VDU5:CLS:PROCdrawgrid:PROChelp
 2740 ENDPROC
 2750 :
 2760 DEF PROCpos
 2770 R%=39:V=grid*square/32+3:IF V<25 V
=25
 2780 IF puz=TRUE V=V+V:R%=79
 2790 VDU4,14,28,V,31,R%,3
 2800 CLS:ENDPROC
 2810 :
 2820 DEF PROCfilledsquares
```

Continued on page 54

BEEBUG Education

by Mark Sealey

Product	CASS, Curriculum Analysis Support System, a primary school admin package for the BBC Master etc.
Supplier	RESOURCE Exeter Road, Wheatley, Doncaster DN2 4PY. Tel (0302) 328735
Price	£29.95 plus VAT

Whatever else is true about the National Curriculum, two things can be fairly stated:

> Firstly, because it will probably be with us for some time at least, anything which will assist teachers in administering it is a definite boon.
>
> Secondly, the National Curriculum is an attempt to describe learning in a modular and subject-based way.

Not all the teaching matter of all subjects has yet been published. So far, though, it consists of from five to sixteen 'Attainment Targets' (ATs). These split up a subject horizontally into distinct areas of knowledge and skills etc., and are themselves each separated vertically into levels, all of which contain one or more (usually three to six) 'Statements of Attainment'. The material to be taught is more closely detailed in corresponding 'Programmes of Study'.

Good and imaginative teachers will want to integrate the work which they plan and carry out for pupils across subjects, though; they will want to keep track of the nature of the whole experience. This is particularly true at primary and lower secondary levels, where learning has not yet become entirely compartmentalised into subjects.

The program under review this month is intended for primary schools; yet it could also be useful at lower secondary level - for integrated cross-curricular project work, for example.

WHAT THE SOFTWARE DOES

CASS, Curriculum Analysis Support System, from established specialists Resource, is an administrative and planning aid which aims to do two things.

Firstly, it seeks to provide just such an overview - by regrouping all the applicable Statements of Attainment under the heading of a topic or a child's work, and then presenting not just a formal (if rather cold) record of areas covered, or "visited" (to use the current jargon).

Secondly it aims to present a picture (visually as well as textually) of what will have been achieved if everything planned for is indeed visited.

The product - which is very reasonably priced - comes on a main 'starter' disc (80 track ADFS only), with an exceptionally clear and well-written 16 page set of A4 tutorial cards and a ring-bound keystrip guide, which is really more of a simple yet easy-to-use reference booklet.

Once you have entered the name of your school and up to 40 pupil names, Curriculum Analysis Support System lets you enter preset levels of attainment which correspond to those of the National Curriculum in the core subjects. That is, you decide to which children each of the levels in the various subjects is going to apply.

You can then add, separately, the degree of success of visiting each and every statement. CASS aggregates all of this into what it calls an 'analysis', which is akin to blocking off a matrix which has the ATs at the top and the Levels at the side for all those areas to be taught or which have been successfully visited etc.

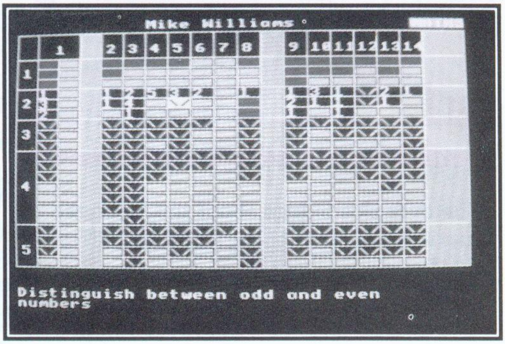
It could all be done on paper. Indeed, many hundreds of teachers are probably busily doing something like it at this moment. But a set of programs like CASS makes it all much, much easier and quicker - especially since it has to be done individually for each child in a class or group.

This same analysis can also result in a printable 'report' - for parents for example. It also neatly compiles quite acceptable 'records' of the ground covered.

THE SOFTWARE'S STRUCTURE

CASS is composed of several inter-related modules. The appropriate code is loaded from disc as required. On booting, a main menu appears with four options: recording, compiling the analysis, applying it to a class and a 'Toolkit' for backing up, configuring the printer and similar housekeeping tasks.

Finding your way about these is straightforward; the logic behind the module layout quickly becomes clear, second nature indeed. There are clear distinctions - often by using different colours and screen modes - between working on data relating to children, to subjects and on appıying the one to the other.



Mike Williams

Distinguish between odd and even numbers

Example of a typical visual record of achievement across 14 ATs for Maths

Various of the clear and near foolproof menus lead you to other pages; in each case this is clear and a 'previous' operation can be returned to - usually by pressing Escape.

It is worth noting at this point that the only other documentation takes the form of a 13cm-deep keystrip-length ring-bound card 'crib'. This is context sensitive (menu by menu) and contains in a clearly laid out format all the other information you will need to run the package.

USING CASS

Because the number crunching involved is heavy and the route through the software at times intricate - though never unclear - you are advised to follow the tutorial closely until you become familiar with CASS.

This leads you through the business of entering details of a mock class list, setting the levels of performance aimed for and logging the actual experience of the material to be taught (or already taught). This is done for each child expressing it in terms of those Statements of Attainment that you select.

This results, finally, in a summary (perhaps 'analysis' is rather inflated a term) of this work in various forms - most usefully a View file of the programmes of study. To use such a text in planning lessons would currently be considered excellent practice and - again - represents a massive saving of time.

Of course, many such summaries can be saved to disc and loaded as and when required. Effectively, a school could build a bank of age-related topics and be able to know instantly which parts of the National Curriculum were relevant as well as having all relevant Programmes of Study instantly to hand.

SOME SMALL NIGGLES

Given the huge quantity of textual data relating to the National Curriculum ATs and Statements of Attainment etc. which has had to be incorporated into this package, it would have been surprising if a few minor blemishes of layout had not crept in: paragraphs too close to their header numbers "1.Can recognize letters of the alphabet"; and commas left in after lists of digits "1,2,3,4," sic.

The 'Hal'-like messages such as "I feel tired now" or "that was a pleasure" during or just after a set of updates or a particularly vigorous number crunching operation might become tiresome once you've got used to working with CASS and just want to clock off as quickly as possible and forget school altogether.

More serious at one point is the possibility of clearing an entire set of data with a single keypress because there is no prompt for confirmation. That could be a costly move if any length of time has been spent setting it all up. Watch it!

CONCLUSIONS

CASS is well written and produced, meets a real need and will save hundreds of hours of time. It is very competitively priced and will pay for itself in less than a school term. The blurb for it claims an almost "intelligent" analysis, akin to some sort of expert-system. This it is not. Where the main strength of the package does lie, though, is in juggling masses of textual and numerical data and outputting it in presentable and useful form.

Similar programs exist already for the Archimedes (e.g. NStore from HS Software) and many more are sure to follow, spawned by the highly structured nature of the National Curriculum.

One criterion which will be increasingly important is that of flexibility as the National Curriculum is added to and (hopefully) at least its most unworkable features amended. Any software which purports to aid its use must be similarly alterable. Resource shows every sign of having addressed this issue with CASS. There are plans for a release called "Small Steps" where Statements of Attainment are broken down for children with special needs, for instance.

This, then - and the overall robust and professional feel to CASS - make it likely to be a winner.

Converting ASCII to Binary

Use this program by Kai S. Ng to convert a sequence of strings, multi-byte numbers etc. into a straightforward binary file.

The program *AscBin* listed here allows you to generate a binary file from an ASCII file of "description" text. In other words, a straightforward text file, which may contain any combination of strings or decimal and hexadecimal numbers in single, double or quad byte form, is converted into a file containing the correct sequence of bytes. An example of its use is the generation of printer control codes without needing to convert Escape codes into VDU or PRINTCHR$() sequences.

The syntax of the source text follows these conventions:

- the quotation mark " signifies the beginning of a textual type on which no conversion is necessary.
- a double quotation mark within a textual type is translated as a single quotation mark.
- a textual type can be closed by another single quotation or a carriage return (ASCII 13) in the source file.
- the separation between 2 types can be achieved by either a space or a comma.
- a type can be either textual or any one of the following if the appropriate prefix is supplied:

	alternative	
prefix	prefix	type
none	U	single byte decimal
#	V	double byte decimal
%	W	quad byte decimal
&	X	single byte hexdecimal
~	Y	double byte hexdecimal
!	Z	quad byte hexdecimal

- there must not be a space between the prefix and the description after it.

Thus, for example, a source file containing the following text:

```
4,&21 !50 "ABC"
```

would convert to a binary file with the following hex dump:

```
04 21 00 00 00 50 41 42 43
```

An example of a source file to test a printer might look like this:

```
2,"This is Pica",13,10,10
27,"M","This is Elite",13,10,10
27,"P",15,"And this is condensed",3
```

The source description can be written on any text processor, but must be filed as a pure ASCII file. Note that the file *must* end with a carriage return or an error will be generated. Entering Basic and CHAINing *AscBin* will now prompt you for the names of the source and destination files, and will generate the latter.

If the binary file is suitable for direct transmission to the printer, you could use the following short program to send it:

```
10 F%=OPENIN"filename"
20 VDU 2:REPEAT
30 VDU 1,BGET#F%
40 UNTIL EOF#F%
50 CLOSE#F%:VDU 3
```

Note that, because Basic treats integers as signed, the largest value that can be used with the quad byte type is &7FFFFFFF. If this is a problem it can be resolved by using other types.

```
   10 REM          >AscBin
   20 REM Program ASCII-Binary converter
   30 REM Version B1.00
   40 REM Author  Kai S. Ng
   50 REM BEEBUG  May 1991
   60 REM Program subject to copyright
   70 :
  100 DIM V(4):@%=&C0C:L%=-1
  110 ON ERROR PROCerr
  120 CLS
  130 REPEAT
  140 INPUT'"Name of source file: "S$
  150 if=OPENIN(S$)
  160 IF if=0 PRINT'"Doesn't exist"
  170 UNTIL if
  180 INPUT'"Name of destination file: "
D$
  190 of=OPENOUT(D$)
  200 L$="":L%=L%+1:REPEAT
  210 C%=BGET#(if):C$=CHR$C%:L$=L$+C$
  220 UNTIL C%=13 OR EOF#(if)
  230 F$="0"
  240 C$=LEFT$(L$,1):C%=ASCC$
  250 L$=RIGHT$(L$,LENL$-1)
  260 IF C%=13 GOTO470
  270 IF F$<>"""" GOTO320
  280 IF C%<>34 PROCdoc:GOTO240
  290 IF ASCLEFT$(L$,1)<>34 GOTO230
  300 PROCdoc:L$=RIGHT$(L$,LENL$-1)
  310 GOTO240
  320 IF INSTR("uvwxyz",F$)>0:GOTO440
  330 IF C%=32 GOTO240
  340 IF INSTR("UVWXYZ",F$)>0 GOTO410
  350 IF F$="," AND C$="," PROCerr
  360 IF INSTR("uvwxyz",C$)>0 C$=CHR$(C%
-32)
  370 IF INSTR("UVWXYZ,""",C$)>0 F$=C$:G
OTO240
  380 IF INSTR("0123456789",C$)>0 F$="U"
:GOTO430
  390 C%=INSTR("#%&~!",C$):IF C%=0 PROCe
rr
  400 F$=MID$("VWXYZ",C%,1):GOTO240
  410 IF INSTR("0123456789",C$)>0 GOTO43
0
  420 IF INSTR("ABCDEF",C$)*INSTR("XYZ",
F$)=0 PROCerr
  430 F$=CHR$(ASCF$+32):V%=0
  440 IF INSTR(", ",C$)>0 PROCdov:GOTO23
0
  450 IF INSTR("uvw",F$)>0 V%=V%*10+EVAL
C$ ELSE V%=V%*16+EVAL("&"+C$)
  460 GOTO240
  470 IF INSTR("UVWXYZ""",F$)>0 PROCerr
  480 IF INSTR("uvwxyz",F$)>0 PROCdov
  490 IF EOF#(if):CLOSE#0:END
  500 GOTO200
  510 :
 1000 DEF PROCerr
 1010 CLOSE#0:REPORT
 1020 PRINT" at input line ";L%;", progr
am line ";ERL
 1030 END
 1040 :
 1050 DEF PROCdoc
 1060 PROCshowc:PROCsend(C%)
 1070 ENDPROC
 1080 :
 1090 DEF PROCdov
 1100 n=2^(INSTR(" uxvywz",F$)DIV2-1)
 1110 FOR I%=1TOn:V(I%)=(V%DIV(256^(I%-1
)))MOD256:NEXT
 1120 PROCshowv:FORI%=1TOn
 1130 PROCsend(V(I%)):NEXT
 1140 ENDPROC
 1150 :
 1160 DEF PROCshowv
 1170 PRINT~V%,V%"   |";,;
 1180 FORI%=1TOn:PRINTCHR$(-32*(V(I%)<32
)-V(I%)*(V(I%)>31));:NEXT:PRINT,;
 1190 FORI%=1TOn:PRINT ;V(I%);",";
 1200 NEXT:PRINT
 1210 ENDPROC
 1220 :
 1230 DEF PROCshowc
 1240 PRINT~C%,C%"   |";,;CHR$(-C%*(C%>3
1));,;C%;","
 1250 ENDPROC
 1260 :
 1270 DEF PROCsend(B%)
 1280 BPUT#of,B%
 1290 ENDPROC
```

Recreational Mathematics
Euclid's Algorithm and Modular Arithmetic

by Michael Taylor

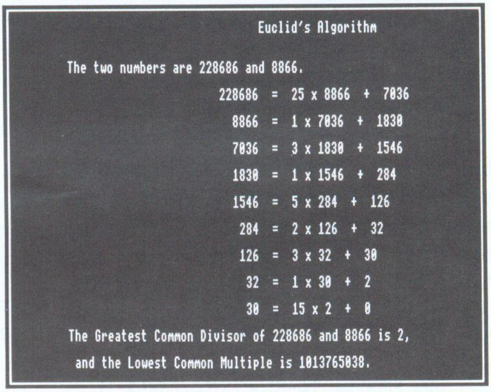
EUCLID'S ALGORITHM

Most readers will remember from school how to find the greatest common divisor, *gcd*, of two numbers (it is also called the highest common factor or *hcf*). The gcd of two numbers is the biggest number that will exactly divide each of them. It is the product of those prime factors (including repeats) which are factors of one number AND the other. The lowest common multiple, or *lcm*, is the product of those prime factors which are factors of one number OR the other. However there is a much faster way of finding the gcd and lcm - much faster than using last month's program to find factors and select the ones in common.

Euclid's algorithm is ingenious, fast, of importance for the RSA deciphering to be explained in a future article - and over two thousand years old! Unlike Euclidian geometry it never seems to have found its way into schools.

The very short program given in listing 1 is not only meant to be useful, but to give an elegant display of the algorithm as it works. Of course, if the reader includes it in another program there is no need to display the intermediate stages of the calculation - as is done here for fun. It can be easily typed in and run. With two large numbers it is worth keeping fingers on the Shift and Control keys to temporarily pause printing to follow the stages of the calculation more easily. The following two numbers will give a short display without scrolling. When asked for the larger number key in 228696 and for the smaller try 8866.

The last remainder is the gcd, 26, and dividing it into the product of the two number, gives their lcm, 77985336. An important use of the algorithm is to test whether two numbers are relatively prime which will be the case if their gcd is 1. We will use an extended version of the algorithm in another article for finding inverses in modular arithmetic.



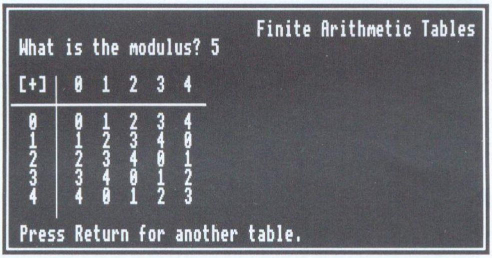
```
                         Euclid's Algorithm

 The two numbers are 228686 and 8866.
                   228686  =  25 x 8866  +  7036
                     8866  =  1 x 7036  +  1830
                     7036  =  3 x 1830  +  1546
                     1830  =  1 x 1546  +  284
                     1546  =  5 x 284  +  126
                      284  =  2 x 126  +  32
                      126  =  3 x 32  +  30
                       32  =  1 x 30  +  2
                       30  =  15 x 2  +  0
 The Greatest Common Divisor of 228686 and 8866 is 2,
  and the Lowest Common Multiple is 1013765038.
```

Finding the lcm and gcd of 228696 and 8866

MODULAR ARITHMETIC

Our second program this month allows you to play with what are sometimes called modular or finite arithmetics (or arithmetics of residues). It displays addition, multiplication and power tables for finite arithmetics with moduli up to 23. There is only space here for a brief explanation: the remainders on division by a number (the *modulus*), can take part in a full arithmetic (with the operators +, -, x and /) provided that the modulus is prime. Rather than read further formal explanation, the reader is invited to glance at some tables.

The program is easily typed in. Here is an example to show what it does and to illustrate finite arithmetic mod 5. When asked for the modulus key in '5', and when asked for the operator key in '+'.



```
                              Finite Arithmetic Tables
What is the modulus? 5

[+] | 0 1 2 3 4
----+----------
 0  | 0 1 2 3 4
 1  | 1 2 3 4 0
 2  | 2 3 4 0 1
 3  | 3 4 0 1 2
 4  | 4 0 1 2 3

Press Return for another table.
```

Modulus 5 arithmetic using the '+' operator

It can be seen that 1 + 2 = 3 (not surprisingly), but also that 3 + 4 = 2 (in the program we write, rather more correctly, 3 + 4 ≡ 2 (mod 5), but provided we know what we mean, it is easier to print an equals sign). Of course, the operands do not have their usual meaning and the numeral 2 can be taken to denote the remainder when 3 + 4 is divided by 5.

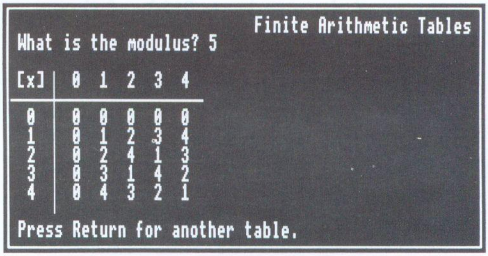
More correctly, 2 denotes the equivalence class of all those natural numbers which when divided by 5 have a remainder of 2, or, which amounts to the same thing, the equivalence class of all those natural numbers which when 2 is subtracted from them are divisible by 5. The structure as a whole is an example of a field in which operations analogous to all those of ordinary arithmetic are possible.

Now run the program, again with modulus 5, but with the 'multiplication' operator 'x'.

It can be seen that, 5 being a prime number, every 'number' in the table can be multiplied by some other number to give a product of 1. This other number is its inverse. It makes 'division' possible, for example 1/2 = 3 (mod 5). From now on, we will omit quote marks from 'number', 'division' and so on, understanding that they are so called by analogy. The reader may want to check why division is not possible with a composite modulus such as 6.

By far the most interesting table is the one with powers. Choose a bigger prime number - such as 23 and the operator '^'.

For convenience the base number is in the left-hand column, and one can then imagine that the exponent, which is 'above' it, is to be selected from the top row. Notice the column of 1's showing that the 22nd power of any number (mod 23 and not a multiple of 23) is 1 - Fermat's little theorem again. Notice how the inverse of a number in the left hand column is to be found in the second column from the right, its inverse (mod 23) being equal to its 21st power (the inverse of N (mod m) is equal to N to the power of the Euler number minus one, and here, with 'm' a prime number, 23, the Euler number is 22 and hence the inverse is the 21st power).



```
                              Finite Arithmetic Tables
What is the modulus? 5

[x] | 0 1 2 3 4
----+----------
 0  | 0 0 0 0 0
 1  | 0 1 2 3 4
 2  | 0 2 4 1 3
 3  | 0 3 1 4 2
 4  | 0 4 3 2 1

Press Return for another table.
```

Modulus 5 arithmetic using the 'x' operator

This table of powers is rich in patterns and the student of number theory can have fun explaining them!

It may be disappointing that the screen only allows moduli up to 23. However, your printer may accept larger tables. A future program will cope with modular arithmetic using moduli as big as 1,000,000,000 that allows addition, subtraction, multiplication, division and powers. Before attempting division it will use Euclid's algorithm to check that the quotient is relatively prime to the modulus, and then an extension of it to find the inverse.

PROGRAM OUTLINE FOR EUCLID'S ALGORITHM

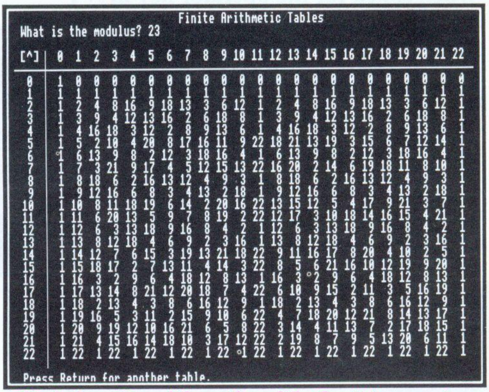
The larger of two numbers is divided by the smaller, and the remainder is found. Then this process is repeated where the larger number is what was previously the smaller number, and the smaller number is what was the remainder. Eventually there is no remainder. As is illustrated, the last (non-zero) remainder is the gcd of the two original numbers.

A rough explanation is that, in the first line of the display, any factor of 228696 and 8866 will also be one of 7046. The second time that this process is repeated the same common factors, 2 and 13 will be 'caught' in the remainder, now 1820. The remainders get smaller and smaller until the last one, apart from zero, just consists of the product, 2*13, of these common factors - the gcd, 26.

Once the gcd has been found, the lcm 77985336 is easily found by dividing the product of the two numbers 2027618736 by their gcd (of course, with large numbers the lcm is imprecisely represented in floating point form).

PROGRAM OUTLINE FOR FINITE ARITHMETIC TABLES

Hardly any explanation is needed: the MOD operator of Basic is used, and most of the program is devoted to producing a clear table. However, there is a problem that powers could exceed the limits for integer variables long before operated on by MOD. To stop this, the power is found by repeated multiplication and continued use of MOD to 'prune' the numbers before they get too big.



```
                     Finite Arithmetic Tables
What is the modulus? 23

[^] |  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
----+---------------------------------------------------------------------
  0 |  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  1 |  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  2 |  1  2  4  8 16  9 18 13  3  6 12  1  2  4  8 16  9 18 13  3  6 12  1
  3 |  1  3  9  4 12 13 16  2  6 18  8  1  3  9  4 12 13 16  2  6 18  8  1
  4 |  1  4 16 18  3 12  2  8  9 13  6  1  4 16 18  3 12  2  8  9 13  6  1
  5 |  1  5  2 10  4 20  8 17 16 11  9 22 18 21 13 19  3 15  6  7 12 14  1
  6 |  1  6 13  9  8  2 12  3 18 16  4  1  6 13  9  8  2 12  3 18 16  4  1
  7 |  1  7  3 21  9 17  4  5 12 15 13 22 16 20  2 14  6 19 18 11  8 10  1
  8 |  1  8 18  6  2 16 13 12  4  9  3  1  8 18  6  2 16 13 12  4  9  3  1
  9 |  1  9 12 16  6  8  3  4 13  2 18  1  9 12 16  6  8  3  4 13  2 18  1
 10 |  1 10  8 11 18 19  6 14  2 20 16 22 13 15 12  5  4 17  9 21  3  7  1
 11 |  1 11  6 20 13  5  9  7  8 19  2 22 12 17  3 10 18 14 16 15  4 21  1
 12 |  1 12  6  3 13 18  9 16  8  4  2  1 12  6  3 13 18  9 16  8  4  2  1
 13 |  1 13  8 12 18  4  6  9  2  3 16  1 13  8 12 18  4  6  9  2  3 16  1
 14 |  1 14 12  7  6 15  3 19 13 21 18 22  9 11 16 17  8 20  4 10  2  5  1
 15 |  1 15 18 17  2  7 13 11  4 14  3 22  8  5  6 21 16 10 12 19  9 20  1
 16 |  1 16  3  2  9  6  4 18 12  8 13  1 16  3  2  9  6  4 18 12  8 13  1
 17 |  1 17 13 14  8 21 12 20 18  7  4 22  6 10  9 15  2 11  3  5 16 19  1
 18 |  1 18  2 13  4  3  8  6 16 12  9  1 18  2 13  4  3  8  6 16 12  9  1
 19 |  1 19 16  5  3 11  2 15  9 10  6 22  4  7 18 20 12 21  8 14 13 17  1
 20 |  1 20  9 19 12 10 16 21  6  5  8 22  3 14  4 11 13  7  2 17 18 15  1
 21 |  1 21  4 15 16 14 18 10  3 17 12 22  2 19  8  7  9  5 13 20  6 11  1
 22 |  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1

Press Return for another table.
```

Modulus 23 arithmetic with the '^' operator

In a future program, where large numbers are used in modular arithmetic, great care will have to be taken to keep integer variables below 2^31 - 1 and yet allow numbers as big as 1,000,000,000 to be used for multiplication - and, with even more difficulty, powers. The reader may like to anticipate how this can be achieved.

Some appropriate references on number theory (given last month) are repeated here for convenience:

1. *The Higher Arithmetic*, by H. Davenport (1952), Cambridge University Press.
2. *Number Theory and Its History*, by Oystein Ore (1948), Dover Publications Inc.
3. *Think of a Number* (1990), by Malcolm E Lines, Adam Hilger.
4. *Numbers, Groups and Codes* (1989), by J.F. Humphreys and M.Y. Prest, Cambridge University Press.
5. *"The Penguin Dictionary of Curious and Interesting Numbers"* (1986), by David Wells.

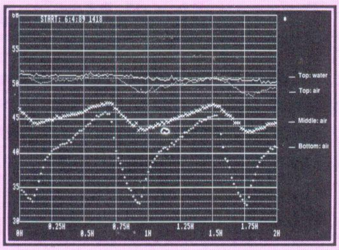
Listing 1

```
   10 REM Program Euclid
   20 REM Version B1.3
   30 REM Author  Michael Taylor
   40 REM BEEBUG  May 1991
   50 REM Program subject to copyright
   60 :
  100 MODE0
  110 VDU19,0,7,0;0;0;
  120 VDU19,1,0,0;0;0;
  130 PRINT'SPC30;"Euclid's Algorithm"
  140 @%=0:VDU28,0,31,79,3
  150 REPEAT
  160 PRINT: VDU10,10,10,11,11,11
  170 INPUT " Bigger  Number? " B%
  180 INPUT " Smaller Number? " A%
  190 CLS:PRINT'" The two numbers are ";
B%;" and ";A%;"."
  200 Y% = B%: X% = A%
  210 REPEAT
  220 PRINT'TAB(30-LENSTR$Y%);
  230 PRINTY%;"  =  ";Y%DIVX%;" x "X%;"
 +  ";Y%MODX%
  240 YN% = X%: XN% = Y% MODX%
  250 Y%=YN%: X%=XN%
  260 UNTIL XN%=0
  270 PRINT '" The Greatest Common Divis
or of ";B%;" and ";A%;" is ";Y%",";''"
and the Lowest Common Multiple is ";A%*B
%/Y%;".":UNTILFALSE
  280 END
```

Listing 2

```
   10 REM Program Galois
   20 REM Version B1.3
   30 REM Author  Michael Taylor
   40 REM BEEBUG  May 1991
   50 REM Program Subject to Copyright
   60 :
  100 MODE0
  110 VDU19,0,7,0;0;0;
  120 VDU19,1,0,0;0;0;
  130 @%=3
  140 REPEAT
  150 PROCInput
  160 PROCTable
  170 PRINT''" Press Return for another
table."
  180 REPEAT UNTIL GET=&0D
  190 UNTIL FALSE
  200 END
  210 :
 1000 DEF PROCInput
 1010 REPEAT
 1020 CLS
 1030 PRINT'SPC27;"Finite Arithmetic Tab
les"
 1040 INPUT" What is the modulus? "M%:RE
M Maximum of 24 for screen display, bigg
er possible on printers.
 1050 UNTIL M%>1
 1060 REPEAT
 1070 PRINT'" '+', 'x', '^':";:INPUT" Wh
at is the operator? "A$;
 1080 VDU13,11,11:PRINTSPC159:VDU13,11,1
1
 1090 IF A$="^"OR A$="~" A$="^"
 1100 IF A$=";"OR A$="+" A$="+"
 1110 IF A$="x"OR A$="X" OR A$="*" A$="x
"
 1120 UNTIL A$="^" OR A$="+" OR A$="x"
 1130 ENDPROC
 1140 :
 1150 DEF PROCTable
 1160 PRINT'" [";A$;"] ";
 1170 FOR A%=0 TO M%-1
 1180 PRINT A%;
 1190 NEXT
 1200 MOVE 0,844:DRAW 102+M%*48,844
 1210 PRINT
 1220 FOR B%=0 TO M%-1
 1230 PRINT'B%;:
 1240 PRINTTAB(5);
 1250 P%=1
 1260 FOR A%=0 TO M%-1
 1270 IF A$="+"PRINT(A%+B%)MODM%;
 1280 IF A$="x"PRINT(A%*B%)MODM%;
 1290 IF A$="^"PRINTP%;:P%=(P%*B%)MODM%
 1300 NEXT
 1310 NEXT
 1320 MOVE 80,892:DRAW 80,814-32*M%
 1330 ENDPROC
```

Best of BEEBUG Software



Applications II Disc

Crossword Editor - for designing, editing and solving crosswords
Monthly Desk Diary - a month-to-view calendar which can also be printed
3D Landscapes - generates three dimensional landscapes
Real Time Clock - a real time digital alarm clock displayed on the screen
Running Four Temperatures - calibrates and plots up to four temperatures
Julia Sets - fascinating extensions of the Mandelbrot set
Foreign Language Tester - foreign character definer and language tester
Label Processor - for designing and printing labels on Epson compatible printers
Share Investor - assists decision making when buying and selling shares.

General Utilities Disc

Printer Buffer (**Master series only**)
Sprite Editor/Animator
Mode 7 Screen Editor
Epson Character Definer
ROM Filing System Generator
Multi-Column Printing
Multi-Character Printer Driver for View
ROM Controller
BEEBUG MiniWimp (**Requires sideways RAM**)

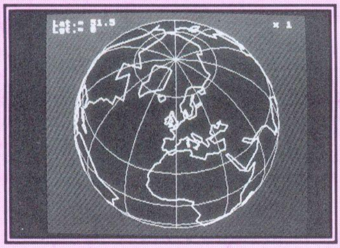
EDIKIT ROM

An indispensible utility ROM for all Basic programmers, containing the following commands:

***FTEXT** (find text) ***FBASIC** (find Basic) ***FPROCFN** (find proc/function)
***LPROC** (list proc) ***LFN** (list function) ***LFROM** (list 8 lines of program)
***RTEXT** (replace text) ***RBASIC** (replace Basic) ***SYSINF** (system info)
***VARLIST** (list program variables) ***FKDEFS** (function key definitions)

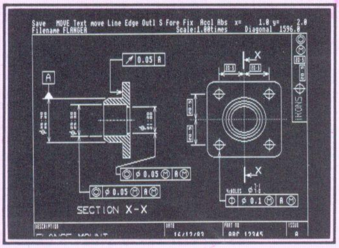
Incorporating the updated Basic Booster utilities:

SUPER SQUEEZE PARTIAL RENUMBER PROGRAM LISTER RESEQUENCER SMART RENUMBER TEXTLOAD AND TEXTSAVE



Applications I Disc

Business Graphics - for producing graphs, charts and diagrams
Video Cataloguer - catalogue and print labels for your video cassettes
Phone Book - an on-screen telephone book which can be easily edited and updated
Personalised Letter-Headings - design a stylish logo for your letter heads
Appointments Diary - a computerised appointments diary
Mapping the British Isles - draw a map of the British Isles at any size
Selective Breeding - a superb graphical display of selective breeding of insects
Personalised Address Book - on-screen address and phone book
The Earth from Space - draw a picture of the Earth as seen from any point in space
Page Designer - a page-making package for Epson compatible printers
World by Night and Day - a display of the world showing night and day for any time and date of the year



Save MOVE Text move Line Edge Outl S Fore Fix Accl Abs x= 1.0 y= 2.0
Filename FLANGER Scale:1.00times Diagonal 1596.0
A
X
SECTION X-X

ASTAAD

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

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

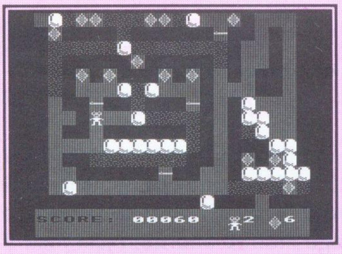
	Stock Code	Price		Stock Code	Price
ASTAAD (80 track DFS)	1407a	£5.95	**ASTAAD** (3.5" ADFS)	1408a	£5.95
EDIKIT (EPROM)	1451a	£7.75			
EDIKIT (40/80T DFS)	1450a	£5.75	**EDIKIT** (3.5" ADFS)	1452a	£5.75
Applications II (80 track DFS)	1411a	£4.00	**Applications II** (3.5" ADFS)	1412a	£4.00
Applications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

Please add p&p

BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS.

Best of BEEBUG Software

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.

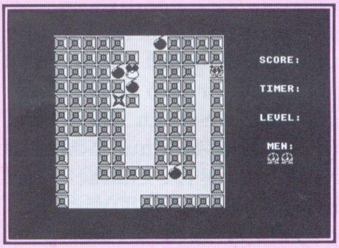
Pitfall Pete - Collect all the diamonds on the screen, but try not to trap yourself when you dislodge the many boulders on your way.

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.



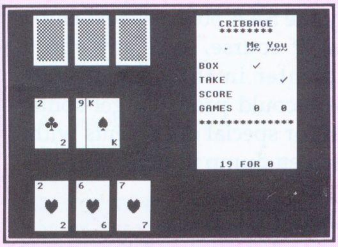
Board Games

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

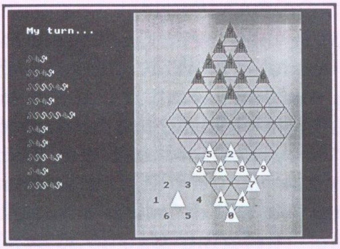
Roll of Honour - Score as many points as possible by throwing the five dice in this on-screen version of 'Yahtze'.

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

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



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



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

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

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

	Stock Code	Price		Stock Code	Price
Arcade Games (40/80 track DFS)	PAG1a	£5.95	**Arcade Games** (3.5" ADFS)	PAG2a	£5.95
Board Games (40/80 track DFS)	PBG1a	£5.95	**Board Games** (3.5" ADFS)	PBG2a	£5.95

Please add p&p:

UK:	*£1.00 for the first item and 50p for every additional item*
Europe and Eire:	*£ 1.60 for the first item and 80p for every additional item*
Elswhere:	*£ 2.60 for the first item and £1.30 for every additional item*

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

Mr Toad's Fontz ROM

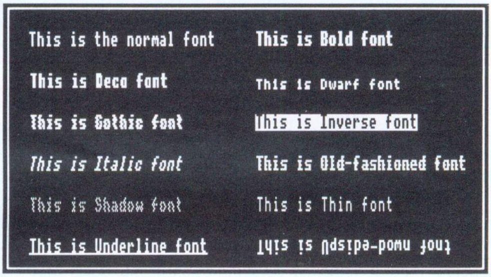
Add some character to your text displays with this utility from David Holton.

This short Sideways ROM image will provide eleven different typefaces on the Master without putting you to the labour of typing in long lists of data. It is intended mainly to extend the visual appeal of your Basic programs by allowing a wide range of styles, but I also find that one or two of the fonts considerably ease eye-strain when word processing in eighty-column modes. It works by redefining the character set held in Private RAM in "ANDY" from &8900 to &8FFF, rather than by altering screen memory, and it will therefore not work in modes 7 and 135, in which the bit-patterns are produced directly by the SAA 5050 Teletext chip. It will not, of course, have any effect on your printer in the normal way (though you could probably get some nice effects for special documents with a printer screen dump utility).

CREATING THE ROM

Type in listing 1, save it and then run the program to assemble the code. Lines 2430 onward will copy the code to the highest-numbered available sideways RAM slot and initialise it, so you don't need to perform Ctrl-Break before using it - an important point if you incorporate it into a Basic program. Note, however, that if a co-processor is active you *will* have to press Ctrl-Break, since line 2470 will poke the co-pro memory whereas the ROM vector table is in the I/O processor. Typing *ROMS and *HELP should show that *Fontz* is present. Issuing any of the commands listed under *HELP*, e.g. *BOL, *ITA or *UND, should give you a demonstration of that character set. Figure 1 shows a sample of all the fonts available.

As an alternative to assembling the code each time it is used, you can save the ROM image on disc and then just load it in when required. Once the image is in place, you can use **SRSave"FontROM" 8000 8400 n*, where n is the number of the sideways RAM slot. To load it back in, use **SRLoad"FontROM" 8000 n*. If you do it this way, however, you must press Ctrl-Break to initialise the ROM before you can use the fonts.



```
This is the normal font        This is Bold font
This is Deco font              This is Dwarf font
This is Gothic font            This is Inverse font
This is Italic font            This is Old-fashioned font
This is Shadow font            This is Thin font
This is Underline font         ⊥ɥıs ıs ∩dsıpǝ-poʍu ɟouʇ
```

The range of fonts available from Mr Toad's Fontz ROM

USING THE FONTS IN YOUR OWN PROGRAMS

The coding has been arranged in separate modules to allow you to omit anything you don't want. If you aim to assemble it as a procedure at the start of a Basic program and use the star commands in your menus, title screens and so on, you will definitely not want the demo - deleting line 1930 will accomplish this, though for permanent use you will probably also wish to delete lines 1950-2090, 2450 and 2480. You may also decide to dispense with the *HELP facility - naughty, but it does save space and assembly time. Missing out line 330 will do the trick, but again, you may just as well delete 360-430 and 2250-2390.

If in any given program you only need a few of the fonts, then delete all reference to the others, including the relevant lines of the command table, or you'll get an assembly error. For my own use, I have included the entire thing (except the demo) as a procedure in a menu program which chains a suite of Basic games; each game then needs only the relevant star commands. Calling any of the fonts automatically cancels the one previously in force, though you can have some fun combining them if you temporarily delete line 1780. You can use as many fonts as you like in one screen, as the effect is not retrospective - earlier parts of the screen are not affected.

It should be understood that fonts produced solely by manipulating the bit-maps of the entire character set can never give perfect results - to get that, it is necessary to provide eight bytes of data per font for each of the 94 characters - eleven fonts would therefore need over 8k of data alone. One or two of the fonts here are frankly only put in for fun; it is unlikely that you would ever really want to use them. *DWArf is only usable with upper-case letters, though it is not bad within these limits. Some fonts give better results in certain modes and colours; *BOLd, for example, is very handy when word processing with dark letters on light "paper"; *THIn is excellent in mode 2, also in modes 0, 4 etc when the "ink" is lighter than the "paper". Give them all a thorough try, therefore, before deleting any.

I would not claim the basic idea here to be in any way new: bold, italic and thin have been around so long that I have no idea who to credit - sorry, whoever you are! The remaining eight fonts are "my own" in the sense that I dreamed them up myself, but it's virtually certain that at least some have been done before. I'm sure, too, that I'm not the first to implement them as a sideways ROM. If I had known of one, however, I'd have missed a lot of fun fiddling with these! It kept me off the streets for quite a few evenings.

```
  10 REM        >Fontz
  20 REM Program Mr Toad's Fontz ROM
  30 REM Version B1.00
  40 REM Author  David Holton
  50 REM BEEBUG  May 1991
  60 REM Program subject to copyright
  70 :
 100 Z%=&7800
 110 osasci=&FFE3
 120 osnewl=&FFE7
 130 oswrch=&FFEE
 140 osword=&FFF1
 150 osbyte=&FFF4
 160 :
 170 FOR N%=4 TO 6 STEP 2
 180 P%=&8000:O%=Z%
 190 [ OPT N%
 200 :
 210 BRK:BRK:BRK
 220 JMP cantYouLeaveMeAlone
 230 EQUB &82
 240 EQUB offset MOD 256
 250 EQUB 144
 260 .title
 270 EQUS "Mr Toad's Fontz Rom"
 280 .offset
 290 BRK:EQUS"(C) BEEBUG May 1991":BRK
 300 :
 310 .cantYouLeaveMeAlone
 320 CMP #4:BEQ SupposeIdBetterCheck
 330 CMP #9:BEQ somebodyWantsHelp
 340 RTS
 350 :
 360 .somebodyWantsHelp
 370 PHA:PHX:PHY:JSR osnewl:LDX #&FF
 380 .helpLoop
 390 INX:LDA helpText,X:PHP:JSR osasci
 400 PLP:BNE helpLoop
 410 JSR osnewl
 420 PLY:PLX:PLA:RTS
 430 :
 440 .SupposeIdBetterCheck
 450 PHA:PHX:PHY:LDX #&FF
 460 .chkLp
 470 LDA (&F2),Y:INY
 480 CMP #&20:BEQ chkLp:AND #&DF
 490 INX:CMP pickYourOwn,X:BEQ chkLp
 500 LDA pickYourOwn,X:BMI haveWeGotOne
 510 .tryNext
 520 INX:LDA pickYourOwn,X:BPL tryNext
 530 INX:PLY:PHY:BRA chkLp
 540 .haveWeGotOne
 550 CMP #&FF:BNE gotOne
 560 PLY:PLX:PLA:RTS
 570 .gotOne
 580 STA &DEFF
 590 INX:LDA pickYourOwn,X:STA &DEFE
 600 JMP (&DEFE)
 610 :
 620 .doBold
 630 LDA #bold DIV &100:STA &DEFF
 640 LDA #bold MOD &100:STA &DEFE
 650 JMP offWeGo
 660 .bold
 670 LDA &DEF0,X:LSR A:ORA &DEF0,X
 680 STA &DEF0,X
 690 INX:CPX #9:BNE bold:RTS
 700 :
 710 .doDeco
 720 LDA #deco DIV &100:STA &DEFF
 730 LDA #deco MOD &100:STA &DEFE
 740 JMP offWeGo
 750 .deco
 760 JSR thin:LDX #1
 770 JSR bold:LDX #1:JSR bold:RTS
 780 :
 790 .doDwarf
 800 LDA #dwarf DIV &100:STA &DEFF
 810 LDA #dwarf MOD &100:STA &DEFE
 820 JMP offWeGo
 830 .dwarf
```

```
 840 LDA &DEF7:STA &DEF8
 850 LDA &DEF5:STA &DEF7
 860 LDA &DEF4:STA &DEF6
 870 LDA &DEF3:STA &DEF5
 880 LDA &DEF1:STA &DEF4
 890 STZ &DEF3:STZ &DEF2
 900 STZ &DEF1:RTS
 910 :
 920 .doGoth
 930 LDA #goth DIV &100:STA &DEFF
 940 LDA #goth MOD &100:STA &DEFE
 950 JMP offWeGo
 960 .goth
 970 JSR shad:LDX #1:JSR bold
 980 LDX #1:JSR bold:RTS
 990 :
1000 .doInv
1010 LDA #inv DIV &100:STA &DEFF
1020 LDA #inv MOD &100:STA &DEFE
1030 JMP offWeGo
1040 .inv
1050 LDA &DEF0,X:EOR #&FF:STA &DEF0,X
1060 INX:CPX #9:BNE inv:RTS
1070 :
1080 .doItal
1090 LDA #ital DIV &100:STA &DEFF
1100 LDA #ital MOD &100:STA &DEFE
1110 JMP offWeGo
1120 .ital
1130 LSR &DEF1:LSR &DEF1
1140 LSR &DEF2:LSR &DEF3
1150 ASL &DEF6:ASL &DEF7
1160 ASL &DEF8:ASL &DEF8:RTS
1170 :
1180 .doNorm
1190 JSR norm:JMP byeToad
1200 .norm
1210 LDA #25:LDY #0:LDX #0
1220 JSR osbyte:RTS
1230 :
1240 .doOld
1250 LDA #old DIV &100:STA &DEFF
1260 LDA #old MOD &100:STA &DEFE
1270 JMP offWeGo
1280 .old
1290 LDA &DEF0,X:LSR A:AND #&3F
1300 ORA &DEF0,X:STA &DEF0,X
```

```
1310 INX:INX:CPX #9:BNE old:RTS
1320 :
1330 .doShad
1340 LDA #shad DIV &100:STA &DEFF
1350 LDA #shad MOD &100:STA &DEFE
1360 JMP offWeGo
1370 .shad
1380 LDA &DEF0,X:AND #&2A:STA &DEF0,X
1390 INX:LDA &DEF0,X
1400 AND #&55:STA &DEF0,X
1410 INX:CPX #9:BNE shad:RTS
1420 :
1430 .doThin
1440 LDA #thin DIV &100:STA &DEFF
1450 LDA #thin MOD &100:STA &DEFE
1460 JMP offWeGo
1470 .thin
1480 LDA &DEF0,X:LSR A:AND &DEF0,X
1490 STA &DEF0,X
1500 INX:CPX #9:BNE thin:RTS
1510 :
1520 .doUnLi
1530 LDA #unLi DIV &100:STA &DEFF
1540 LDA #unLi MOD &100:STA &DEFE
1550 JMP offWeGo
1560 .unLi
1570 LDA #&FF:STA &DEF8:RTS
1580 :
1590 .doUps
1600 LDA #ups DIV &100:STA &DEFF
1610 LDA #ups MOD &100:STA &DEFE
1620 JMP offWeGo
1630 .ups
1640 STZ &DEE0,X:LDY #7
1650 .bkToFr
1660 LDA #0:ROR &DEF0,X:ADC &DEE0,X
1670 ASL A:STA &DEE0,X:DEY:BNE bkToFr
1680 INX:CPX #9:BNE ups
1690 LDX #1:LDY #8
1700 .ups2
1710 LDA &DEE0,Y:LSR A:STA &DEF0,X
1720 INX:DEY:BNE ups2:RTS
1730 :
1740 .jumpOut
1750 JMP (&DEFE)
1760 :
1770 .offWeGo
```

```
1780 JSR norm
1790 LDA #32:STA &DEF0
1800 LDA #23:STA &DEEF
1810 .outerLoop
1820 LDY #&DE:LDX #&F0:LDA #10
1830 JSR osword
1840 LDX #1:JSR jumpOut
1850 :
1860 LDX #0
1870 .setBitz
1880 LDA &DEEF,X:JSR oswrch
1890 INX:CPX #&0B:BNE setBitz
1900 INC &DEF0:BPL outerLoop
1910 :
1920 .byeToad
1930 JSR demo
1940 PLY:PLY:PLY:LDA #0:RTS
1950 :
1960 .demo
1970 JSR osnewl:JSR osnewl
1980 LDA #ASC"0"
1990 .demoLp
2000 JSR osasci:INA
2010 CMP #1+ASC"9":BEQhop1
2020 CMP #1+ASC"Z":BEQhop2
2030 CMP #1+ASC"z":BNEdemoLp
2040 JSR osnewl:JSR osnewl
2050 RTS
2060 .hop1 JSR osnewl:JSR osnewl
2070 LDA #ASC"A":BRA demoLp
2080 .hop2 JSR osnewl:JSR osnewl
2090 LDA #ASC"a":BRA demoLp
2100 :
2110 .pickYourOwn
2120 EQUS"BOL"+CHR$13:EQUBdoBold DIV &1
00:EQUBdoBold MOD &100
2130 EQUS"DEC"+CHR$13:EQUBdoDeco DIV &1
00:EQUBdoDeco MOD &100
2140 EQUS"DWA"+CHR$13:EQUBdoDwarf DIV &
100:EQUBdoDwarf MOD &100
2150 EQUS"GOT"+CHR$13:EQUBdoGoth DIV &1
00:EQUBdoGoth MOD &100
2160 EQUS"INV"+CHR$13:EQUBdoInv DIV &10
0:EQUBdoInv MOD &100
2170 EQUS"ITA"+CHR$13:EQUBdoItal DIV &1
00:EQUBdoItal MOD &100
2180 EQUS"NOR"+CHR$13:EQUBdoNorm DIV &1
```

```
00:EQUBdoNorm MOD &100
 2190 EQUS"OLD"+CHR$13:EQUBdoOld DIV &10
0:EQUBdoOld MOD &100
 2200 EQUS"SHA"+CHR$13:EQUB doShad DIV &
100:EQUB doShad MOD &100
 2210 EQUS"THI"+CHR$13:EQUBdoThin DIV &1
00:EQUBdoThin MOD &100
 2220 EQUS"UND"+CHR$13:EQUBdoUnLi DIV &1
00:EQUBdoUnLi MOD &100
 2230 EQUS"UPS"+CHR$13:EQUBdoUps DIV &10
0:EQUBdoUps MOD &100
 2240 EQUB &FF
 2250 :
 2260 .helpText
 2270 EQUS CHR$13+"Mr Toad's Magic Fontz
 Rom"+CHR$13
 2280 EQUS"  BOL.d"+CHR$13
 2290 EQUS"  DEC.o"+CHR$13
 2300 EQUS"  DWA.rf"+CHR$13
 2310 EQUS"  GOT.hic"+CHR$13
```

```
 2320 EQUS"  INV.erse"+CHR$13
 2330 EQUS"  ITA.lic"+CHR$13
 2340 EQUS"  NOR.mal"+CHR$13
 2350 EQUS"  OLD.fashioned"+CHR$13
 2360 EQUS"  SHA.dow"+CHR$13
 2370 EQUS"  THI.n"+CHR$13
 2380 EQUS"  UND.erline"+CHR$13
 2390 EQUS"  UPS.ide-down"+CHR$13:BRK
 2400 ]
 2410 NEXT
 2420 :
 2430 FOR N%=7 TO 4 STEP-1
 2440 IF ?(&2A1+N%) THEN NEXT
 2450 IF N%=3 PRINT'''"No spare SRAM slot
!":END
 2460 OSCLI "SRWRITE "+STR$~Z%+" "+STR$~
(O%+1)+" 8000 "+STR$ N%
 2470 ?(&2A1+N%)=&82
 2480 PRINT'''"READY IN SLOT ";N%:END
```

+Windows and +Catalogue (continued from page 16)

desirability of a sort function in a database (however simple) ranges from vital to essential. If you were to keep details of your record collection in +Catalogue, for example, you could search for all records by a named artist but you couldn't pull them together in a group although you could print out each entry as you found it.

You could keep names and addresses in it but the same restrictions apply. Although it's not necessary to keep an address file in alphabetical order, it seems somehow unnatural not to do so. Lack of a sort function means anything you want to appear in alphabetical order will have to be entered as such when creating the file.

I always think it's a good idea if developers provide demo data for programs such as databases (no matter how easy-to-use they may be) so the manual can run you through some examples but no demo files were supplied with +Catalogue. It's a good idea to create a few sample files of your own before you get down to serious use, just to make sure you know your way around.

Lest I seem too critical, I am merely pointing out what the program cannot do that a user might reasonably expect of a database. But again, it's worth pointing out that +Catalogue doesn't claim to be a multi-purpose database but a cataloguing system.

Once you've got the operation sussed it's very easy to use and at the price it must be good value for money. Just be sure that it has sufficient facilities for your purpose.

Linear Equations

by Bernard Hill

One of the most fundamental scientific computations is the solution of systems of linear equations. The basic algorithm for doing this, Gaussian Elimination, is relatively simple and has changed little in the past 150 years since it was invented.

My first contact with large systems of linear equations was back in the 1960s when as a student I sent some clothes to a laundry for cleaning. Unfortunately the laundry company did not distribute a tariff. Rather than take the easy route and call in for one in my aberrant maths student way, I preferred to record exactly the clothes cleaned for nine weeks and the resulting bill, thus covering the eight different items I was sending and a possible standing charge. So by the ninth week I had 9 equations in 9 unknowns and set to to solve them by hand. Sadly in the 60s we had no electronic calculators - let alone BBC micros - and I record with humility that apparently the laundry were paying me over £3 to clean a handkerchief but charging over £10 for shirts. Maybe they changed their rates half way through. But on the other hand, perhaps I made a mistake in my algebra.

A SIMPLE EXAMPLE

Suppose we have three variables x, y and z and the following three equations:

$$\begin{aligned} x + 3y - 4z &= 8 \\ x + y - 2z &= 2 \\ -x - 2y + 5z &= -1 \end{aligned}$$

Our goal is to find the values of the three variables which simultaneously satisfy the equations.

In order to make this method easier to follow, and to extend the algorithm to any number of variables, we will deal only with the arrays of coefficients, and write the array as 3 x 4 to include the left-hand side in the now rectangular array:

$$\begin{array}{rrrr} 1 & 3 & -4 & 8 \\ 1 & 1 & -2 & 2 \\ -1 & -2 & 5 & -1 \end{array}$$

There are several things we can do to such a system of equations which will help solve them:

1. Interchange rows.
2. Multiply a row by a constant.
3. Add two rows and replace one of them by this sum.

Rules 2 and 3 can be combined to form:

4. Add a multiple of one row to another and replace the latter with the new value.

Now there are many ways of solving these equations, but Gaussian Elimination uses these four operations as follows:

Step one is to put a 1 in the top left by dividing the first equation by the value in the top left-hand position. We already have a 1 here so we can continue to the second stage immediately: this is to subtract a suitable multiple of the first row from every other row so that we make the first column all zero (except for the top line). We use Rule 4 above:

```
Row 2 = Row 2 + (-1)*Row 1, and
Row 3 = Row 3 +   1 *Row 1, giving:

   1   3  -4      8
   0  -2   2     -6
   0   1   1      7
```

Our zeros in the first column are in place.

Now we repeat the process with the second row but only using the bottom right-hand corner of the array:

```
      -2   2     -6
       1   1      7
```

So divide row 2 by -2:

```
   1   3  -4      8
   0   1  -1      3
   0   1   1      7
```

Now we can subtract the second row from the third:

```
(Row 3 = Row 3+(-1)*Row 2)

   1   3  -4      8
   0   1  -1      3
   0   0   2      4
```

This process can be repeated no matter how many equations (rows) there are, the process is called 'diagonalising' and places 0s in the bottom left-hand half of the main array and 1s down the leading diagonal.

The equations are now easy to solve as follows: The third equation merely reads "2z=4", giving z=2. Having solved for the third variable (z), then the second variable (y) follows from the second equation as this now reads "y-z=3", and since z=2 we have y=5. And of course x follows easily by substituting these two values into the top equation (x+3y-4z=8) to give x=1. This process is called 'back substitution'.

POSSIBLE PROBLEMS

Now while we will always be able to complete this back substitution process it may be that we shall fail to be able to complete the diagonalisation stage. For instance, should we be unlucky enough to have a zero value in the top left-hand position initially then we should not be able to divide through by this as in the very first stage. In fact at any of these division stages if we were to divide by a small quantity then large numbers could result, and subsequently subtracting large numbers from each other can lose accuracy.

So in fact we always interchange equations to make sure that the largest absolute value is in the diagonal position before dividing by it. This process is called 'pivoting', or strictly, 'partial pivoting' as we are not bothering to re-arrange variables (interchanging columns) as this procedure gives no extra accuracy in this case.

If our largest absolute value is in fact still zero this indicates that our equations are thus unsolvable, or 'singular' as the mathematician would say. Depending on the equations, we may have a case where there is no solution (such as if two are mutually contradictory, as in x+y=1, x+y=2) or many solutions (for example if two equations are identical). In either case we would find that our pivot value was zero.

Listing 1 contains a program to solve any set of linear equations in up to about 65 unknowns (depending on memory). It follows very closely the algorithm given here with the coefficient array a(N,N+1) and the solutions array x(N). As always in these articles I have sacrificed speed for clarity of program and leave it to you to introduce resident integer variables for the various counters in the program if speed is a problem. Be aware that equations with 65 variables take about 20 minutes to solve by the program as it stands!

INACCURACIES

It is still possible even with partial pivoting to have large errors coming into our calculations. This could be because the equations are intrinsically 'ill-conditioned', and subject to computer calculation errors. For instance consider the equations:

```
x + 1000*y = 1001
x + 1001*y = 1002
```

with the solution x=1, y=1. Now any small change in the coefficients changes the solution by a large amount: for instance if the first equation were changed to:

```
x + 1000.001*y = 1001
```

then the actual solution to 6 decimal places would be x=-0.002002, y=1.001001: quite a change from the first! Moreover, listing 1 gives the y value correct, but x is given as -0.001929 due to arithmetic rounding errors. Problems of this class are not as rare as we would like, especially with larger numbers of equations, and any commercial equation-solving routine would also give a 'conditioning number' as part of the output to indicate the degree of reliability of the solution. Unfortunately the calculation of this is much longer than that to solve the actual equations and so the routine is not listed here.

In next month's article we shall look at an application of linear equations and meet this concept of ill-conditioning again.

```
  10 REM Gaussian Elimination
  20 REM Version B1.0
  30 REM Author  Bernard Hill
  40 REM Beebug  May 1991
  50 REM Program subject to copyright
  60 :
 100 READ N:REM number of equations
 110 DIM a(N,N+1),x(N)
 120 REM read coefficients
 130 FOR i=1 TO N
 140 FOR j=1 TO N+1
 150 READ a(i,j)
 160 NEXT j
 170 NEXT i
 180 PROCprint(“Equations are:”)
 190 PROCeliminate
 200 PROCprint(“After diagonalising:”)
 210 PROCsubstitute
 220 PRINT “Solutions are:”
 230 FOR i=1 TO N
 240 PRINT TAB(10);x(i)
 250 NEXT
 260 END
 270 :
1000 DEF PROCeliminate
1010 LOCAL j,k,max,t,v
1020 FOR i=1 TO N-1
1030 pivot=FNpivot(i)
1040 FOR j=1 TO N+1
1050 a(i,j)=a(i,j)/pivot
1060 NEXT j
1070 FOR k=i+1 TO N
1080 v=a(k,i)
1090 FOR j=1 TO N+1
```

```
 1100 a(k,j)=a(k,j)-a(i,j)*v
 1110 NEXT j
 1120 NEXT k
 1130 NEXT i
 1140 ENDPROC
 1150 :
 2000 DEF FNpivot(i)
 2010 REM find largest in col i below
 2020 REM the diagonal, interchanging
 2030 REM if necessary
 2040 LOCAL j,k,t,max
 2050 FOR j=i TO N
 2060 IF ABSa(j,i)>max THEN k=j:max=ABSa
(j,i)
 2070 NEXT j
 2080 IF max=0 THEN PRINT"Singular":END
 2090 IF i=k THEN =a(i,i)
 2100 FOR j=1 TO N+1:REM swap rows
 2110 t=a(i,j):a(i,j)=a(k,j):a(k,j)=t
 2120 NEXT j
 2130 =a(i,i):REM return pivot value
 2140 :
 3000 DEF PROCsubstitute
 3010 LOCAL i,k,tot
 3020 x(N)=a(N,N+1)/a(N,N)
```

```
 3030 FOR i=N-1 TO 1 STEP -1
 3040 tot=a(i,N+1)
 3050 FOR k=i+1 TO N
 3060 tot=tot-a(i,k)*x(k)
 3070 NEXT k
 3080 x(i)=tot
 3090 NEXT i
 3100 ENDPROC
 3110 :
 4000 DEF PROCprint(msg$)
 4010 LOCAL i,j,@%:@%=&507
 4020 PRINT'msg$'
 4030 FOR i=1 TO N
 4040 FOR j=1 TO N+1
 4050 PRINTTAB((j-1)*10);a(i,j);
 4060 NEXT j:PRINT
 4070 NEXT i:PRINT
 4080 ENDPROC
 4090 :
 4100 REM data follows here:
 5000 DATA 3:REM number of equations
 5010 DATA 1,3,-4,8
 5020 DATA 1,1,-2,2
 5030 DATA -1,-2,5,-1
```

Mini Wall Planner (continued from page 13)

```
 1570 FOR X=0 TO x STEP 116
 1580 MOVE X,70:DRAW X,980
 1590 NEXT
 1600 FOR Y=70 TO 980 STEP 70
 1610 MOVE 0,Y:DRAW x+116,Y
 1620 NEXT
 1630 ENDPROC
 1640 :
 1650 DEF PROCprint
 1660 LOCAL l%,t%,u%,p
 1670 VDU1,27,1,65,1,8
 1680 FOR l%=0 TO 39
 1690 VDU 1,27,1,76,1,0,1,3
 1700 FOR t%=31 TO 0 STEP -1
 1710 FOR u%=7 TO 0 STEP -1
 1720 p=?(&7C00+t%*320+u%+l%*8)
 1730 VDU1,p,1,p,1,p
```

```
 1740 NEXT
 1750 NEXT
 1760 VDU1,13,1,10
 1770 NEXT
 1780 VDU1,27,1,50
 1790 ENDPROC
 1800 :
 1810 DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,A
UG,SEP,OCT,NOV,DEC
 1820 :
 1830 REM REPLACE THE FOLLOWING DATA STA
TEMENTS WITH YOUR OWN NAMES, DATES ETC.
 1840 :
 1850 DATA A,B,C,D,E,F,G
 1860 DATA 1,2,3,4,5
 1870 DATA a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p
```

512 Forum

by Robin Burton

This month I'll start with a simple query which may benefit some of you without you even realising there was a question. I know that the Forum has a number of readers who are not 512 owners, some of whom may become users in time, so some of these too might benefit.

BUYING A USED 512

I've had a couple of letters from new 512 users who have managed to purchase a machine after considerable effort, searching through classified advertisements and chasing numerous potential buys before finally achieving success (see Members' ads in BEEBUG, for example). It's clear that 512s are getting very difficult to find nowadays, and also that buyers can't always expect to receive everything that should be included with the machine, always assuming of course that they know what should be supplied.

I've been trying to locate three 512s myself, for a school that wants to upgrade its Master 128s to allow them to run DOS, so I can verify that tracking down 512s is no easy job these days. One obvious but unfortunate side effect of an active demand but limited supply is that prices can increase dramatically and this seems to be happening to the 512.

Frequently, even in private sales, prices of £180.00 are being asked, and sadly this now seems to be the standard 'going rate' for a second-hand 512 purchased from a dealer. Those who purchased their 512 after Acorn's final price reduction can therefore feel literally doubly pleased with their machine, since it would today cost about twice as much, even though it would be second-hand.

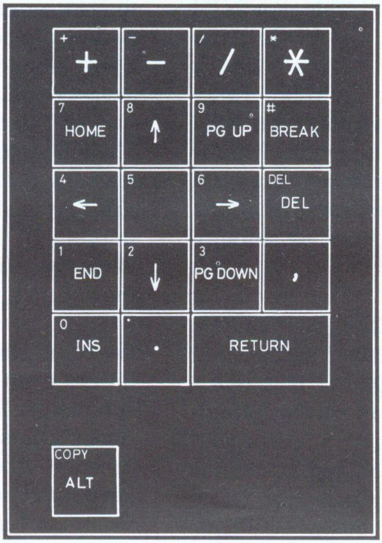
The other fact which often afflicts a used machine purchase is that various bits of the original package can have been mislaid by the previous owner, and that's the real motivation for this part of the Forum this month.

From my discussions with dealers and letters from users it's obviously quite common for the mouse, the manual, the four DOS issue discs (especially with 'proper' Acorn labels) or the numeric keypad layout card, or sometimes all of them, to have gone missing when a machine is sold or traded in. If you are a prospective 512 purchaser or you recently acquired one, the above is the list of items you should receive with the board. If anything is missing the price should be reduced to reflect this, for example by £20.00 or more for the mouse.

Although there's a thriving second-hand market for 512s, it's clear that Acorn's lack of support and especially the lack of publicity for the machine when it was a current product are still affecting buyers today. Although it might seem incredible to most of us, dealers have told me that they have traded in Masters against an upgrade to an Archimedes system only to discover later, when they took the lid off the Master, that there was a 512 inside.

Obviously in these cases the previous owner didn't know about the 512, and equally neither did the dealer at the time of the trade-in. The result for the next owner of the 512 of course is a board with none of the normal items which should be supplied as standard.

I've had a couple of requests to include the keypad layout in the Forum from recent 512 purchasers, which is why the information is reproduced here this month. This, and my own recent search for machines is basically responsible for the ramblings above, but it also leads me to think about the following mystery.



512 keypad layout

WHERE ARE THEY?

According to the best figures I have, Acorn manufactured and sold (eventually) about 13,000 512s in total. Judging by the overall response to my 512BBC BASIC offer in 512 Forum over the last two years, BEEBUG readers account for about 400 or so of these (since the software was free I assume that almost everyone sent for it). As a percentage of BEEBUG membership that's roughly 5%, which is an extremely high proportion when compared to the overall number of 512s versus BBC micros. If I include Essential Software and Dabs Press customers, the total number of 512 users I 'know about' one way or another is still probably less than 750 in all (there's naturally a good deal of overlap between all three sources of information).

I don't suggest that absolutely every 512 user reads 512 Forum, nor that they all must alternatively be either Essential Software or Dabs Press customers, but it's a fair bet that the vast majority of active users fall into at least one of these categories. The question that I keep wondering about therefore, is where are the rest of those 512s?

From the (admittedly unscientific) available evidence it would seem that somewhere there are up to 12,000 512s that are not being used. An even more appalling thought is that, while there are reasonable numbers of BBC micro users who would dearly love to get hold of a 512 to use it properly now they belatedly know about it, it's equally possible, even likely, that thousands of unused machines are owned by people who don't even know they have them!

You might well feel this idea is just too far fetched, but I don't think so! Apart from the numerous stories I've heard from dealers, I recently bought a 512 from a primary school (thanks to the help of Seymour Leeds, a 512 Forum reader) The school concerned most certainly didn't use the 512, in fact they didn't actually know what it was! That 512 is now on its way to a much more active and productive life, while the school's finances have benefited appropriately. In this case there's therefore a happy ending all round, but it does illustrate my point.

By the way, if you bought a used Master 128 and you're now looking for a 512 but you've never taken the lid off your micro, it might be a good idea to do so. You never know!

PROBLEM SOLVER

Long time Forum readers will remember previous encounters with this program (designed to help with installing and running PC software on a 512), a story which in retrospect was more like a television 'soap' than a software review.

Shibumi Soft, the suppliers of *Problem Solver*, seemed to disappear from the scene after a series of regular advertisements over quite some time, leaving those who had yet to purchase or upgrade wondering what happened to them.

A few months ago advertisements again began to appear, though this time under the slightly changed name of Shibumi Software. Most of the more recent advertisements concentrate on their new area of interest, Archimedes software, but thanks are again due to Seymour Leeds and to Steve Clements for providing the following information.

An updated version of Problem Solver is now available (address below) both to new purchasers and existing users. Also, since there is now a U.K. address, telephone orders can be taken if you wish to use Access or Visa for payment.

If you have an original version of Problem Solver you can obtain the upgrade for £5.00 from Shibumi Software quoting the serial number of your original copy, there's no need to return the original disc. The cost to new purchasers is £24.95 plus £1.00 p&p.

I haven't tried the new version myself, so I can't say whether it's a worthwhile upgrade or not if you have the original version. However, according to Shibumi's reply to Seymour, the new version is more reliable than the old one, especially with DOS Plus 2.1, and it has no known bugs. As I understand it, the operation of the new program and the facilities provided by it look just like the old one.

One other fact that emerged is that orders and queries have been taking a very long time to be processed by Shibumi. I know that Seymour wrote several times and waited literally months before finally receiving a reply. In the end however, he did get the software, together with a letter explaining that although they now have a U.K. mailing address and 'phone number, Shibumi Soft still operates from Portugal so sometimes "things take longer than they should".

I've also heard from other members that a response to a letter or an order took weeks if not months, though, so far as I know everyone eventually did receive a reply and Shibumi Soft have promised that their service will improve.

HINTS AND TIPS

I deduce from one or two queries that some of you have forgotten, or perhaps have never heard of the trick of loading a second copy of COMMAND.COM when you have a package that won't work in the 512. A reminder seems to be in order, therefore.

One package I know that this applies to is a genealogy program called *Brother's Keeper*. At first sight it seems that the latest version of this program will not run in the 512 although a previous one did. Initially the new version sometimes hangs the system, sometimes it produces an error message. However, it turns out

Continued on page 56

BEEBUG Function/Procedure Library (3)

by Stefano Spina

This month we present two major routines from the function and procedure library of Stefano Spina. The first provides a generalised input function giving detailed user control over defaults and the checking of the input received. The second provides an immediate mode 7 menu system, which can again be adapted to suit the needs of many programs. Note that both routines require the use (as documented) of functions and/or procedures previously published in this library.

As both of this month's routines are quite lengthy, they will repay detailed and careful study before you try to use them in your own programs.

THE FUNCTION/PROCEDURE LIBRARY (PART 3)

Routine 20: Input

Type: PROCEDURE

Syntax: PROCin(A$,B$,C$,D$,E$,A%,B%,C,D)

Scope: General purpose customised input routine

Parameters:
- A$ Input prompt
- B$ Input prompt (extension)
- C$ Allowable set of characters or ranges
- D$ Allowable strings
- E$ Default input
- A% Input field length
- B% Flag for numeric inputs
  - 0 Input isn't numerical (even if it's composed of digits)
  - 1 Numerical input with low/high limits
  - 2 Numerical input with no limits
- C Low limit \
- D High limit/ these values are checked only if B%=1, but dummy values must otherwise be included

Notes: This is a big procedure which provides detailed user control of input for use in many applications.

The procedure automatically checks for the currently selected screen mode and reserves the last three lines as an input window.

With a few changes it can be modified to accept input in any part of the screen.

The input is checked character by character, first matching with the allowed set or range placed into the C$ parameter.

If C$ is a null string then no control is performed.

A range of characters is specified by a pair of letters; the routine doesn't discriminate between lower/upper case letters.

A specific set of characters can be selected by preceding them with a backslash "\".

For example, "AF15*." allows characters A to F and 1 to 5 plus asterisk and full stop to be accepted. Range(s) (when required) must precede a set.

If the input string exceeds the max length (A% param.) then the string is deleted with an audible signal.

The Delete key deletes the last character entered.

The Tab key sets the global logical variable style% to FALSE, so that a subsequent FNstyle call doesn't change the output's layout.

On entry *style%* is always set to TRUE.

If the Return key is pressed, then the resulting string ischecked against the range of allowable strings.

These are specified in the D$ parameter; if D$ is a null string then no control is performed.

Allowable string(s) must be followed by the tilde "~".

If the resulting string doesn't match any of these then the string is deleted with an audible signal.

If a default input is supplied in the E$ parameter, then the simple Return key press exits from the procedure.

The output is placed into three global variables:

sr$	holds the string
sr%	holds the integer value
sr	holds the real value

If B% is set to 0 then both sr% and sr are 0.

Numerical inputs can be decimal or hexadecimal for integer values, while for real values only decimal input is allowed.

Hexadecimal values must be preceded by an "&".

PROCin1	Clears the input window
FNin1	Checks for valid characters
FNin2	Checks for valid strings
FNin3	Checks for numerical values; if a number has an illegal format, say a real number with two full stops, the input is deleted with an audible signal.

Related: PROCcolour, PROCcurs, FNmd, FNswap

```
  100 PROCin("File","(Max 10 chrs)","AZ09
\ *","","Store",10,0,0,0)
```

This call checks for alphanumeric characters and asterisk, has no string control, the default input is "Store", the max length is 10 characters and no numerical check.

On exit:	sr$	<filename>
	sr%	Zero
	sr	Zero

```
  100 PROCin("Article","(Max 6 digits)","
0 9","","000000",6,2,0,0)
```

This call checks for digits only, has no string control, the default input is "000000", the max field length is 6 digits and numeric output isn't checked for min/max value.

On exit:	sr$	Value in string format
	sr%	Integer value
	sr	Real value

```
  100 PROCin("Noise","DB Dolby B   DC Dol
b y C   DX Dbx   NO None","\DBCNOX","DB~D
C~D X~NO~","DB",2,0,0,0)
```

This call checks for "DBCNOX" characters only, checks for "DB", "DC", "DX", "NO" strings, the default input is "DB", the max length is 2 characters and has no numerical check.

On exit: sr$ Required noise reduction type
sr% Zero
sr Zero

```
   100 tot%=0
   110 REPEAT
   120 PROCin("Quantity","","09","",CHR$13
, 3,1,10,100)
   130 IF sr$<>CHR$13 tot%=tot%+sr%
   140 UNTIL sr$=CHR$13
```

In this case a loop ends when the Return key is pressed without input. With numerical inputs enclosed in a loop the default must be CHR$13.

Routine 21: Menu

Type: FUNCTION
Syntax: FNmenu(H$,O%)
Purpose: General purpose menu screen
Parameters: H$ Header string
O% Number of options
Notes: The function runs in mode 135 (mode 7 on a model B) and self-centres itself on the screen depending on the menu options.
Up/Down cursor keys select the desired option and Return exits from the function.
Line 23070 (as currently numbered) must hold the menu option descriptions.
Related: PROCcurs

```
   10 PROCinit
   20 ON ERROR PROCerr
   30 REPEAT
   40 me%=FNmenu("Working Disc",5)
   50 ON me% PROCcomm,PROCasm,PROCbasic,P
R OCgraphic,PROCexit
   60 UNTIL FALSE
   70 END
22540 DATA Command Page,Assembler Tools,B
a sic Editor,Graphics,EXIT
```

```
22220 REM Input
22230 :
22240 DEF PROCin(A$,B$,C$,D$,E$,A%,B%,C,
D)
22250 LOCAL F$,G$,E%,F%,G%,H%,J%,K%
22260 C$=FNswap(C$,1):D$=FNswap(D$,1)
22270 IF FNmd>=0 AND FNmd<=5 AND FNmd<>3
 Inr%=29 ELSE Inr%=22
22280 IF FNmd=0 OR FNmd=3 Inl%=78 ELSE I
F FNmd=2 OR FNmd=5 Inl%=19 ELSE Inl%=38
22290 H%=LEN(A$)+3:PROCcolor(0)
22300 PRINTTAB(1,Inr%)A$": "STRING$(A%,"
 ")" "B$
22310 REPEAT:REPEAT:VDU7:PROCcolor(1)
22320 PRINTTAB(H%,Inr%)STRING$(A%," ")
22330 F$="":G%=0:sr$="":sr=0:sr%=0
22340 K%=0:J%=0:style%=TRUE:F%=FALSE
22350 REPEAT
22360 E%=FALSE:PROCcurs(1)
22370 VDU31,H%+K%,Inr%:F$=GET$:G%=ASCF$
22380 IF G%=9 style%=FALSE:GOTO 22450
```

```
22390 IF G%=13 THEN 22450
22400 IF G%=127 AND J%=0 THEN 22450
22410 IF G%=127 J%=J%-1:sr$=LEFT$(sr$,J%
):PRINTTAB(H%+K%,Inr%)CHR$127:K%=K%-1:GO
TO 22450
22420 IF NOT FNin1(C$,F$) THEN 22450
22430 sr$=sr$+F$:J%=LEN(sr$)
22440 IF J%>A% E%=TRUE ELSE K%=K%+1:PRIN
TTAB(H%-1+K%,Inr%)F$
22450 UNTIL G%=13 OR E%
22460 IF sr$="" AND E$="" E%=TRUE
22470 IF E% THEN 22510
22480 IF sr$="" AND E$<>"" sr$=E$:style%
=FALSE:F%=TRUE
22490 IF B%>0 IF NOT FNin3 E%=TRUE:GOTO
22510
22500 IF FNin2 ELSE E%=TRUE
22510 UNTIL NOTE%
22520 IF F% PROCcolor(0):PROCin1:GOTO 22
580
22530 PROCcurs(0):PROCcolor(0):VDU7
```

```
22540 PRINTTAB(1,Inr%+1)"Confirm (Y/N)"
22550 REPEAT:G$=GET$
22560 UNTIL FNin1("YN",G$)
22570 IF (ASCG$ AND 95)=89 PROCin1 ELSE
PRINTTAB(1,Inr%+1)STRING$(13," "):E%=TRU
E
22580 UNTIL NOT E%
22590 ENDPROC
22600 :
22610 DEF PROCin1
22620 PRINTTAB(1,Inr%)STRING$(Inl%," ")
22630 PRINTTAB(1,Inr%+1)STRING$(Inl%," "
)
22640 ENDPROC
22650 :
22660 DEF FNin1(C$,F$)
22670 IF C$="" =TRUE
22680 F$=FNswap(F$,1)
22690 IF INSTR(C$,F$)>0 AND F$<>"\" =TRU
E
22700 IF LEFT$(C$,1)="\" C$="01"+C$
22710 LOCAL J%,K%:J%=1
22720 REPEAT
22730 K%=F$>MID$(C$,J%,1) AND F$<MID$(C$
,J%+1,1):J%=J%+2
22740 UNTIL K% OR J%>=LEN(C$) OR MID$(C$
,J%,1)="\"
22750 IF K% OR F$=RIGHT$(C$,1) =TRUE ELS
E =FALSE
22760 :
22770 DEF FNin2
22780 IFD$="" =TRUE
22790 LOCAL Y$,Z$,W%,X%,Y%,Z%
22800 W%=LEND$:X%=1:Z%=0
22810 Y$=FNswap(sr$,1)
22820 REPEAT
22830 Z%=INSTR(D$,"~",X%)
22840 Z$=MID$(D$,X%,(Z%-X%))
22850 IF Z$=Y$ Y%=TRUE ELSE X%=Z%+1:Y%=F
ALSE
22860 UNTIL Y% OR X%>=W%:=Y%
22870 :
22880 DEF FNin3
22890 IF sr$=CHR$13 =TRUE
22900 LOCAL Z$,X%,Y%,Z%:X%=TRUE
22910 IF LEFT$(sr$,1)="-" Z$=MID$(sr$,2)
 ELSE Z$=sr$
22920 IF LEFT$(Z$,1)="&" Z%=TRUE:Z$=MID$
(Z$,2) ELSE Z%=FALSE
22930 IF INSTR(Z$,"&")<>0 OR INSTR(Z$,"-
")<>0:=FALSE
22940 IF LEFT$(Z$,1)="." AND Z%:=FALSE
22950 FOR Y%=65 TO 70
22960 IF INSTR(Z$,CHR$Y%)<>0 AND NOT Z%:
X%=FALSE
22970 NEXT Y%:IF NOT X%:=FALSE
22980 sr=EVALsr$:sr%=EVALsr$:IF B%=2 =TR
UE
22990 IF sr<C OR sr>D =FALSE ELSE=TRUE
23000 :
23010 REM Menu
23020 :
23030 DEF FNmenu(H$,O%)
23040 LOCAL N$,J%,L%,P%,S%,T%:CLS:VDU22,
135
23050 L%=LEN(H$):S%=(31-L%)DIV2-1
23060 T%=(19-O%)DIV2:RESTORE 23070
23070 DATA *****
23080 PROCcurs(0):PRINTTAB(1,T%)CHR$145C
HR$232STRING$(31,CHR$172)CHR$180
23090 FOR J%=T%+1 TO T%+2
23100 PRINTTAB(0,J%)CHR$145CHR$141CHR$23
4CHR$131STRING$(S%," ")H$STRING$(28-(S%+
L%)," ")CHR$141CHR$145CHR$181
23110 NEXT
23120 PRINTTAB(1,T%+3)CHR$145CHR$234STRI
NG$(31,CHR$172)CHR$181
23130 FOR J%=1 TO O%:READN$
23140 PRINTTAB(1,J%+T%+3)CHR$145CHR$234C
HR$131CHR$156CHR$134N$;TAB(31)CHR$145CHR
$156CHR$181
23150 NEXT
23160 PRINTTAB(5,J%+T%+2)CHR$130N$
23170 PRINTTAB(1,T%+O%+4)CHR$145CHR$170S
TRING$(31,CHR$172)CHR$37
23180 P%=1
23190 PRINTTAB(4,P%+T%+3)CHR$157CHR$129
23200 *FX4,1
23210 REPEAT
23220 REPEAT:L%=GET
23230 UNTIL L%=13 OR L%=138 OR L%=139
23240 IF L%=13 THEN 23300
23250 IF P%<O% J%=134 ELSE J%=130
23260 PRINTTAB(4,P%+T%+3)CHR$156CHR$J%
23270 IF L%=138 P%=P%+1:IF P%>O% P%=1
23280 IF L%=139 P%=P%-1:IF P%=0 P%=O%
23290 PRINTTAB(4,P%+T%+3)CHR$157CHR$129
23300 UNTIL L%=13:*FX4,0
23310 =P%
23320 :
```

1st course

VDU and FX Calls (2)

by Mike Williams

We will continue this month with further discussion of VDU calls and their uses. To start with I want to look at a number of VDU calls which are largely concerned with graphics - and remember from last time that where graphics co-ordinates are to be specified each must be followed by a semicolon (;) for correct interpretation.

I have already mentioned that VDU29 can be used to specify a new graphics origin; once such a call has been executed all graphics co-ordinates are taken relative to the new origin (except for window definitions). VDU25 is entirely equivalent to Basic's PLOT instruction, and I refer you to the relevant section of your manual for more information on that. On the whole, there is little advantage to the VDU version, though a sequence of VDU codes, including those for VDU25, could all be included in a single VDU call for economical use of memory.

For example:

```
PLOT 4,0;0;:  PLOT 5,200;0;
PLOT 4,0;200;:PLOT 5,200;200;
```

which draws two parallel straight lines could be written as:

```
VDU25,4,0;0;5,200;0;4,0;200;5,200;200;
```

but this is hardly as easy to understand. Indeed the PLOT instructions I have used could themselves be more understandably written as:

```
MOVE 0,0:  DRAW 200,0
MOVE 0,200:DRAW 200,200
```

MIXING TEXT AND GRAPHICS

Normally, the positioning of text on the screen is related to the rows and columns appropriate to the screen mode you have chosen to use (20, 40 or 80 columns, 25 or 32 rows). If you are mixing text and graphics on the same screen, then more precise positioning of text is called for. This can be achieved by the use of VDU5. Effectively this call joins the text and graphics cursors together, and the position of both is determined by appropriate graphics calls. Thus:

```
VDU5
MOVE 640,512
PRINT"This text starts at the centre"
```

This would have the effect of displaying the text in quotes at graphics co-ordinates 640, 512. However, when you start thinking about it, that statement doesn't really give you enough information. After all, each letter in the string is several pixels high and wide. So which pixel of the first letter is located at the graphics co-ordinates specified?

The answer, perhaps surprisingly, is that it is the top left-hand pixel which is controlled by the graphics cursor. For example, suppose in the same mode, you wanted to position a line of text starting at the bottom left-hand corner of the screen, The sequence would be:

```
VDU5
MOVE 0,28
PRINT"This is at bottom left"
```

Why '28' you might ask. Well, in any graphics mode the screen is divided into

Screen Planning Sheet

To assist in the design and layout of screen displays we are including a free copy of our own Screen Planning Sheet with this issue of BEEBUG. Please feel free to copy this for your own use.

32 lines of text equivalent to 1024 graphics units, that's 32 graphics units per line. Each text character is 8 pixels high, so each pixel corresponds to 4 graphics units. So you might think that we should have written:

```
MOVE 0,32
```

but remember that the bottom-most row of pixels will have graphics co-ordinate 0, the second row of pixels will have a graphics co-ordinate of 4, and so on. So the top row of pixels in the character will have a graphics co-ordinate of four less than 32, i.e. 28. If you feel a little confused, then hopefully the diagram in figure 1 will make things clear.

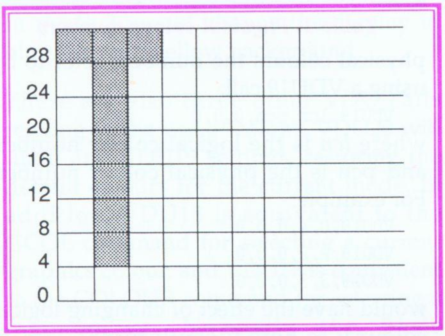


Figure 1. The letter 'T' in graphics position (0,28) in mode 0

In any graphics mode, the number of text rows, and hence the relationship between graphics co-ordinates and character height remains constant. However, the same is not true horizontally, as will be obvious when you reflect that graphics modes can have 20, 40 or 80 columns, yet the width of the screen in graphics co-ordinates is always 1280.

In 20 column modes (modes 2 and 5), each character width corresponds to 64 graphics units, in 40 column modes (modes 1 and 4) each character width is equivalent to 32 graphics units, and in the 80 column mode 0 (illustrated in figure 1), each character corresponds horizontally to just 16 graphics units.

Again, since the first graphic 'x' value is zero, all co-ordinates will be offset (by 2, 4, or 8 depending on mode). In mode 0, a character in the bottom left-hand corner of the screen has its first pixel in position 0, and its last in position 14 (see figure 2). From this you can deduce that each pixel of the character corresponds horizontally with 2 graphics units.

As you can see, this whole business of relating character positions to graphics co-ordinates can be quite confusing, varying as it does with the choice of screen mode. The User Guide for the model B contained a useful planning sheet to assist with this,

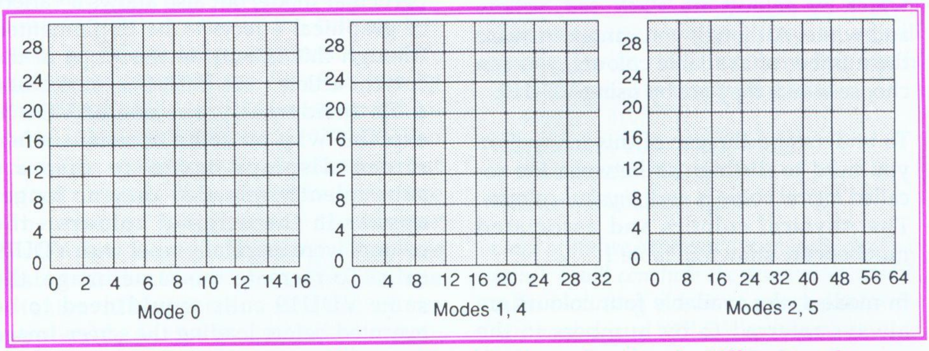


Figure 2. Graphics co-ordinates related to character sizes in different modes

but regrettably this was not included in The Welcome Guides for the Master 128 or Master Compact.

As a final encore on this topic, note that it is not necessary to continue to issue a VDU5 call before every PRINT statement when printing text in graphics mode. Once issued its effect remains in force. What you do need to know as well is that VDU4 separates text and graphics cursors, so that text once again follows the dictates of the text cursor alone.

There is also another feature of writing text at the graphics cursor which can be exploited to achieve interesting effects. Text displayed in this way is controlled just like graphics (i,e. by the settings governed by the GCOL command). By repeating a line of text with a slight offset (and change of colour) you can achieve shadowed text and many other effects, but that's another story.

VDU FOR COLOUR MANIPULATION

One area where the use of VDU codes can be vital is in colour control. As you are probably aware, the number of colours from which you can choose is determined by your choice of screen mode, with the maximum of 16 colours available only in mode 2. In a mode such as mode 5, four colours are possible which by default are black, red, yellow and white. Although you cannot increase the number of available colours, you can choose which they are by using VDU19.

To understand the use of this instruction you need to distinguish between the so-called *logical* colours and *physical* colours. The physical colours and associated numbers are shown in table 1.

Number	Colour
0	black
1	red
2	green
3	yellow
4	blue
5	magenta
6	cyan
7	white
8	flashing black/white
9	flashing red/cyan
10	flashing green/magenta
11	flashing yellow/blue
12	flashing blue/yellow
13	flashing magenta/green
14	flashing cyan/red
15	flashing white/black

Table 1. Physical colour numbers

In mode 4, the available four colours are always referred to by numbers in the range 0 to 3. But you can select which physical colours the numbers refer to by using a VDU19 call:

```
VDU19,lcn,pcn,0,0,0
```

where *lcn* is the logical colour number, and pcn is the physical colour number. For example:

```
VDU19,0,4,0,0,0
VDU19,2,2,0,0,0
VDU19,3,3,0,0,0
```

would have the effect of changing logical colour 0 to blue (default black), logical colour 2 to green (default yellow), and logical colour 3 to yellow (default white). Not only is this useful in merely selecting other than the default colours in a particular mode, but also allows a variety of graphical effects to be implemented (though this is outside the scope of the current article - see BEEBUG Vol.6 Nos.6 & 7). Incidentally, the use of VDU19 explains why on some occasions when screen displays saved to disc are subsequently reloaded they no longer appear in the original colours - the colours were originally reset with VDU19 calls. To recreate the same image, the same VDU19 calls would need to be executed before loading the screen image (but see below).

One further hint which can be useful is a way of specifying a VDU19 call as a control code sequence. VDU19 is equivalent to Ctrl-S. Thus to change the default black background to blue in immediate mode enter:

```
<Ctrl-S>04000
```

(i.e. press Ctrl-S, and then the keys for 04000 but without the commas used in a VDU call proper). This can be used *after* a screen display has been reloaded from disc to change the colours on screen. Likewise, for example, it can be used in View's command mode to change the colours, thus:

```
<Ctrl-S>03000
<Ctrl-S>14000
```

in mode 3 would change the display to blue text on a yellow background.

There are also three other VDU calls relating to the use of colour: VDU20 will undo any VDU19 settings, restoring the default colours for the current mode. In addition, VDU18 is equivalent to the GCOL command for selecting a current graphics colour, and VDU17 is equivalent to the COLOUR command for selecting a text colour. Normally the Basic equivalents are to be preferred for their greater readability.

USING VDU23

The last VDU call which I want to deal with is VDU23. One of the commonest uses of this is in programming user-defined characters which often has its most frequent application in graphics rather than text. However, VDU23 also performs a number of miscellaneous but often useful functions which are documented fully only in the Master Reference Manual Part One. They are listed in the Welcome Guides for the Master 128 and Compact, but those applicable to the model B are largely ignored in its User Guide.

Taking its use in defining characters first, the format of the relevant VDU23 call is:

```
VDU23,code,v0,v1,v2,v3,v4,v5,v6,v7
```

where *code* is the ASCII code of the character to be defined, and *v0* to *v7* are values representing the eight rows of eight pixels which make up any character. ASCII codes in the range 224 to 255 are normally available to the user for his own definitions. All codes in the range 32 to 255 are redefinable, but you need to consult your User Guide before attempting this as it may involve the allocation of extra memory for this purpose, and is one area where the model B is quite different from the Master 128 and Master Compact. It is perhaps also worth pointing out that you cannot redefine mode 7 characters.

You will not infrequently come across programs where the defining values in a VDU23 call are given in hexadecimal (recognised by the '&' preceding each value). In some cases pairs of bytes are combined to save space (like graphics co-ordinates) and recognised by the use of the semicolon rather than comma as separator. Indeed, many programs have both VDU19 and VDU23 calls near the start to initialise colours and additional characters required.

The other uses of the VDU23 call relate to parameters in the range 0 to 31. Most of these are quite complex to use, so we will not deal with them for now, but one is quite essential in many programs:

```
VDU23,1,0;0;0;0;
```

will turn the cursor off, while:

```
VDU23,1,1;0;0;0;
```

will turn it back on again.

Once again we have run out of space, but next month I will conclude the discussion of VDU calls, including a look at some of the more advanced uses of VDU23, and hopefully make a start on FX calls as well.

Crossword Compiler (continued from page 22)

```
 2830 M%=0:FOR N%=1 TO grid:FOR P%=1 TO
grid:IF ?FNspace(N%,P%,1)=ASC"F" M%=M%+1
 2840 NEXT:NEXT:PROCpos:PRINT''"! square
s: ";M%''"# squares: ";grid*grid-M%''" P
ress"'" Space"'" Bar":VDU5:REPEAT:A=GET:
UNTIL A=32:PROChelp
 2850 ENDPROC
 2860 :
 2870 DEFFNspace(x%,y%,z%)
 2880 =TOP+2+x%*60+y%*3+z%
 2890 :
 2900 DEF PROCclr
 2910 FOR M%=grid TO 1 STEP-1:FOR N%=1 T
O grid
 2920 IF (?FNspace(N%,M%,1)=ASC"F" AND Q
%=1) ELSE ?FNspace(N%,M%,0)=32:?FNspace(
N%,M%,1)=ASC"U":PROCunmksq:side=N%:heigh
t=M%:PROCmksq
 2930 NEXT:NEXT:PROCunmksq:side=1:height
=1:PROCmksq
 2940 ENDPROC
 2950 :
 2960 DEF PROCdump
 2970 FOR I%=0 TO 2 STEP2
 2980 P%=&900
 2990 [OPTI%
 3000 LDX#&0
 3010 .n LDY#&0
 3020 .l LDA(&70),Y
 3030 STA&72,Y:INY:CPY#8
 3040 BNEl:LDA#0:STA&84
 3050 .p LDY#0
 3060 .m LDA&72,Y
 3070 ASL A:STA&72,Y:LDA#4
 3080 BIT &83:BEQ e:BCS d
 3090 ROL &85:CLC:JMP e
 3100 .d ROL &85:SEC
 3110 .e ROL &85:INY:CPY&83
 3120 BNE m:LDA#1:JSR&FFEE:LDA&85
 3130 JSR&FFEE:LDA#4:BIT&83:BEQ f
 3140 LDA#1:JSR&FFEE:LDA&85:JSR&FFEE
 3150 .f INC&84:LDA#8:CMP&84:BNE p
 3160 CLC:LDA&70:ADC#&8:STA&70
 3170 LDA&71:ADC#&0:STA&71:INX
 3180 CPX#40:BNE n:RTS
 3190 ]
 3200 NEXT:ENDPROC
 3210 :
```

```
 3220 DEF PROCprt
 3230 PROCpos
 3240 PRINT"Print Size"''"Large/Small"'"
=      ="
 3250 REPEAT:I%=GET:UNTIL I%=76 ORI%=83:
IF I%=76 ?&83=4 ELSE ?&83=8
 3260 DEF PROCsmallprt
 3270 PROCpos:VDU5
 3280 VDU2,1,27,1,64,1,27,1,51,1,24
 3290 !&70=&5800
 3300 FOR I%=&5800 TO &7EC0 STEP320
 3310 IF ?&83=4FOR J%=I% TO I%+4 STEP4:!
&70=J%
 3320 VDU1,27,1,75,1,(2560/?&83)MOD256,1
,(2560/?&83)DIV256
 3330 CALL&900:VDU1,13
 3340 IF ?&83=4 NEXT
 3350 NEXT:VDU1,27,1,50,3:ENDPROC
 3360 :
 3370 DEF PROCpuzzle(c%)
 3380 IF c%<>0PROCacr_dn:IF num%>99 ENDP
ROC
 3390 VDU26,4,7:PRINTTAB(2,31)"MASKING O
UT LETTERS - PLEASE WAIT";:VDU5
 3400 side=1:height=grid
 3410 Q%=2:FOR M%=grid TO 1 STEP-1:FOR N
%=1 TO grid
 3420 IF ?FNspace(N%,M%,1)=ASC"U" PROCun
mksq:side=N%:height=M%:PROCmksq
 3430 NEXT:NEXT:PROCunmksq:Q%=0
 3440 VDU4:PRINTTAB(0,31)SPC39;:VDU5
 3450 PROCprt
 3460 IF c%=0 PROCpartdisplay:ENDPROC
 3470 VDU2
 3480 FOR I%=1 TO (11+(grid-1)/2)*(8/?&8
3)
 3490 VDU1,27,1,106,1,36:NEXT
 3500 VDU1,27,1,50
 3510 VDU2,1,27,1,83,1,0
 3520 VDU1,27,1,108,1,3*(8/?&83)
 3530 FOR h%=grid TO 1 STEP-1
 3540 FOR s%=1 TO grid:VDU1,27,1,80,1,15
 3550 IF space%(s%,h%)=0 VDU1,32,1,32
 3560 IF (space%(s%,h%)<10 AND space%(s%
,h%)>0) VDU1,ASC(LEFT$(STR$(space%(s%,h%
)),1)),1,32
 3570 IF space%(s%,h%)>9 VDU1,ASC(LEFT$(
STR$(space%(s%,h%)),1)),1,ASC(RIGHT$(STR
$(space%(s%,h%)),1))
 3580 IF ?&83=4 VDU1,32,1,32
 3590 VDU1,27,1,77,1,32:IF ?&83=4 VDU1,3
2
 3600 NEXT:VDU1,13:IF ?&83=4 VDU1,13
 3610 NEXT:VDU3:PROCpartdisplay
 3620 ENDPROC
 3630 :
 3640 DEF PROCacr_dn
 3650 VDU26,4,7:PRINTTAB(2,31)"DETERMINI
NG NUMBERING - PLEASE WAIT";
 3660 FOR h%=grid TO 1 STEP-1:FOR s%=1 T
O grid:?FNspace(s%,h%,2)=0:space%(s%,h%)
=0
 3670 IF ?FNspace(s%,h%,1)=ASC"F" NEXT:N
EXT:PROCnumber:ENDPROC
 3680 IF s%=1IF ?FNspace(s%,h%,1)=ASC"U"
 AND ?FNspace(s%+1,h%,1)=ASC"U" ?FNspace
(s%,h%,2)=ASC"A"
 3690 IF (s%>1 AND s%<grid) IF ?FNspace(
s%-1,h%,1)=ASC"F" IF ?FNspace(s%,h%,1)=A
SC"U" AND ?FNspace(s%+1,h%,1)=ASC"U" ?FN
space(s%,h%,2)=ASC"A"
 3700 IF h%=grid IF ?FNspace(s%,h%,1)=AS
C"U" AND ?FNspace(s%,h%-1,1)=ASC"U" ?FNs
pace(s%,h%,2)=?FNspace(s%,h%,2)+ASC"D"
 3710 IF (h%<grid AND h%>1) IF ?FNspace(
s%,h%+1,1)=ASC"F" IF ?FNspace(s%,h%,1)=A
SC"U" AND ?FNspace(s%,h%-1,1)=ASC"U" ?FN
space(s%,h%,2)=?FNspace(s%,h%,2)+ASC"D"
 3720 NEXT:NEXT
 3730 :
 3740 DEF PROCnumber
 3750 num%=1
 3760 FOR h%=grid TO 1 STEP-1:FOR s%=1 T
O grid
 3770 IF ?FNspace(s%,h%,2)<>0 space%(s%,
h%)=num%:num%=num%+1
 3780 NEXT:NEXT
 3790 IF num%>99 PRINTTAB(0,31)CHR$7"TOO
 MANY WORDS Must be <100 - Hit SPACE";:R
EPEAT:UNTIL GET=32
 3800 PRINTTAB(0,31)SPC39;:VDU5
 3810 ENDPROC
 3820 :
 3830 DEF PROCpanswers(T$)
 3840 PROCpos
 3850 PROCacr_dn:IF num%>99 ENDPROC
```

```
 3860 @%=2:VDU4,12:PRINTTAB(5,31)"Press
'Shift' to continue"
 3870 VDU28,10,29,39,1:IF T$="P" VDU2
 3880 PRINT'"ACROSS"':FOR N%=0 TO 1
 3890 IF N%=1 PRINT''"DOWN"'
 3900 FOR h%=grid TO 1 STEP-1:FOR s%=1 T
O grid:A%=0
 3910 IF N%=0 IF ?FNspace(s%,h%,2)=ASC"A
" OR ?FNspace(s%,h%,2)=ASC"A"+ASC"D":PRI
NTspace%(s%,h%);" ";:REPEAT:A%=A%+1:PRIN
TCHR$(?FNspace(s%,h%,0));:s%=s%+1:UNTIL
s%=grid+1 OR ?FNspace(s%,h%,1)=ASC"F":PR
INT" (";A%;")"
 3920 IF N%=1 IF ?FNspace(s%,h%,2)=ASC"D
" OR ?FNspace(s%,h%,2)=ASC"A"+ASC"D":PRI
NTspace%(s%,h%);" ";:h1%=h%:REPEAT:A%=A%
+1:PRINTCHR$(?FNspace(s%,h1%,0));:h1%=h1
%-1:UNTIL h1%=0 OR ?FNspace(s%,h1%,1)=AS
C"F":PRINT" (";A%;")"
 3930 NEXT:NEXT:NEXT:VDU3
 3940 VDU26,7:PRINTTAB(2,30)"PRESS SPACE
 BAR TO RE-DISPLAY GRID";
 3950 *FX15 1
 3960 REPEATUNTIL GET=32:PROCdisplay
 3970 ENDPROC
 3980 :
 3990 DEF PROCprinting
 4000 PROCmksq:PROCpos
 4010 VDU4,14,28,V,31,39,3,12
 4020 PRINT"PRINTING"
 4030 PRINTTAB(0,3)"1-Grid"
 4040 PRINTTAB(0,5)"2-Grid/No"
 4050 PRINTTAB(0,7)"3-As seen"
 4060 PRINTTAB(0,9)"4-Final"
 4070 PRINTTAB(0,11)"5-Answers"
 4080 PRINTTAB(0,13)"6-Menu"
 4090 REPEAT:K%=INKEY(0):UNTIL K%>48 AND
 K%<55
 4100 IF K%=49 PROCpuzzle(0)
 4110 IF K%=50 PROCpuzzle(1)
 4120 IF K%=51 PROCprt:PROChelp:PROCmksq
 4130 IF K%=52 PROCpuzzle(1):VDU2,1,13,1
,13,1,13,3:PROCunmksq:PROCsmallprt:PROCh
elp:PROCmksq
 4140 IF K%=53 PROCpanswers("P")
 4150 IF K%=54 PROChelp
 4160 ENDPROC
```

512 Forum (continued from page 45)

that all that's needed to cure this problem is a second copy of COMMAND.COM.

To try this fix, 'COMMAND' should be loaded a second time before you try to run the package. This is simply achieved by typing 'COMMAND' at the DOS prompt, ensuring that COMMAND.COM is available in a current path. Doing this does of course reduce the amount of free RAM available to the application by about 30K, though in an expanded 512 this is of little concern.

If you have a program that refuses to run normally in the 512 this trick is always worth a try; it could make all the difference. Even if it doesn't, it has the considerable merits that it doesn't take long to try and costs absolutely nothing.

If you do load a second copy of COMMAND.COM and later wish to reclaim the extra memory used by it there's no need to reboot the system (unless it's crashed of course). Simply type 'EXIT' at the DOS prompt, and the second copy of 'COMMAND' will be terminated. This will instantly return you to the normal DOS prompt, so it looks as if nothing happened, but running 'BACKG' both before and after will clearly show that the extra memory has been freed again.

There's one other side effect to loading a second copy of 'COMMAND.COM' which might become obvious if you are in the habit of using the 512's standard function key strings and cursor keys. Of course, if you use a command line editor this won't trouble you.

For example, pressing f5 on the command line produces the word 'COPY' followed by a space. The idea is to save you having to type the command manually: all you need to do is to press f5 and then add the source and destination drives and filenames as appropriate before pressing Return.

However, if a second copy of 'COMMAND.COM' is loaded, all the Acorn function key definitions are lost, being replaced instead by the normal IBM function key actions. This change is temporary though; the Acorn function key definitions return to normal if you later exit back to a single DOS shell.

Address **Shibumi Software,**
3 Parkfield Road, Long Buckby,
Northampton NN6 7QJ.
Tel. (0392) 437756

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

PERPETUAL CALENDAR (Vol.9 No.8)

The following lines need to be amended:

```
  2650 VDU2:VDU28,1,1,1,1,17,0
  2990 VDU2:VDU28,1,1,1,1,17,0
  2790 VDU26,17,3
  3120 VDU26,17,3:VDU3
  3270 VDU1,15:PRINT STRING$(20," ")+STRI
 NG$(5," M  T  W Th  F  S Su ")+" M  T":
 VDU1,17
```

Apart from line 3270, which is required on all machines, these modifications are needed for the model B only, as a result of a quirk in the operating system, which, when the printer is enabled, sends characters preceded by a 1 to the printer twice if a VDU21 command has been issued. Model B owners will also have to alter lines 1370 (in 2 places) and 3470 by replacing "169" with "167".

IMPROVED ROUTE PLANNER (Vol.9 No.9)

There are two small errors in the program as published, both in the magazine and on the disc. Firstly, line 1400 should read:

```
  1400 UNTIL A%=13 OR EOF#X
```

Secondly, lines 3580 and 3590 should be joined together, so that 3580 now reads:

```
  3580 IF ?C%<>32 name$(A%)="":FOR B%=0
 TO 7:name$(A%)=name$(A%)+CHR$((C%?B%)
 AND 127):NEXTB%
```

and 3590 is deleted.

HINTS and tips HINTS and tips HINTS and tips HINTS and tips HINTS and tips

MORE MEMORY FOR THE MODEL B

C.J.Collins

Hard pressed model B programmers may be unaware of a potential extra page of RAM under certain conditions. This occurs when the function keys are programmed to generate ASCII codes with an offset base, e.g. with *FX4,2 and *FX225,129.

Memory page B may then be used for data or machine code without affecting the use of the function keys at all. Naturally, tidying this area to remove 'garbage' is advisable before returning the function keys to their normal programmable state.

AUTOMATED MACRO ENTRY IN VIEW

H.M.Humphries

The Macro function in View is not particularly well documented, but can be one of the more useful features of this word processor. Not only can it be used with form letters, and for creating mailing lists, but it can serve as a simple but extremely flexible database for telephone directories, membership lists and the like; all of which may be interrogated rapidly using the SEARCH function.

If the macro list is lengthy, however, the task of entering the stored commands becomes tedious and time consuming. This problem may be overcome by adopting the following procedure:

1. In command mode, enter the following:

```
*FX228,1
*KEY0 "|!$xx|M|!Z
```

where 'xx' represents any valid macro title.

2. Switch to edit mode, and enter the list in the normal format, but *without* the stored commands.

3. Having completed the entry of data, press f1 to return to the beginning of the text, then hold down the Shift and Ctrl keys and press the function key (f0) defined in step 1, until the chosen stored command has been inserted against each line of input.

Thereafter, the complete macro can be stored to disc in the normal manner.

THOUGHTS ON RANDOM NUMBERS

David Holton

If you want random numbers in machine code, read them from the ROM area. The section from &E000 to &FC00 is a good bet. If writing a program in Basic, why not give the random numbers a 'stir' by generating and discarding some while waiting for input:

```
        REPEAT:r%=RND:i$=INKEY$(1):UNTIL
INSTR("*abc",i$)>1
```

where the '*' is NOT one of the desired inputs.

PROTECTING TIMERS AGAINST BREAK

C.J.Collins

A timer based on event number 4 can be readily combined with an OSWORD 7 (sound) call to make a useful alarm. Such an alarm can also be made proof against both Break and Ctrl-Break by using sideways ROM/RAM. A ROM-based program can be used to set a flag, zero counter locations, set a delay byte, enable event 4, redirect the user vector, and download a short piece of code into user RAM. It must also respond to service call 1 and if the flag is set, jump to the downloadable code to re-enable event 4 and reset the user vector after a Break. I put the code at &380 on my model B, but it could go into any 'safe' location to suit individual needs.

HIDING ROM IMAGES

David Holton

If you give no title and no binary version number for any sideways ROM image you write, it will not appear at all in the list produced by the *ROMS command, yet it will function perfectly in all other respects.

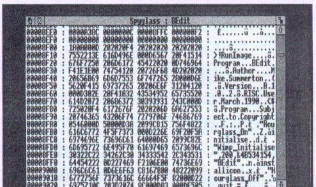
RISC USER

The Archimedes Magazine & Support Group

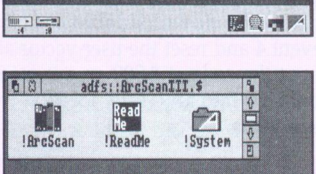
Now in its fourth year of publication, Risc User continues to enjoy the largest circulation of any magazinedevoted solely to the Archimedes range of computers. It provides support for all Archimedes users at work (schools, colleges, universities, industry, government establishments) and home.

Existing Beebug members, interested in the new range of Acorn micros, may either transfer their membership to the new magazine or extend their subscription to include both magazines.

A joint subscription will enable you to keep completely up-to-date with all innovations and the latest information from Acorn and other suppliers on the complete range of BBC micros. RISC User has a massive amount to offer to enthusiasts and professionals at all levels.



```
Spyglass : BEdit
00008EE0 : 0000A3BC 0000000C 00008FFC 000000F2 : £.......ü ..ò...
00008EF0 : 00000000 00000000 00000000 00000000 : ................
00008F00 : 1A0A000D 202020F4 20202020 20202020 : ....ô...........
00008F10 : 7552213E 616D496E 000D6567 20F41514 : >!RunImage....ô.
00008F20 : 676F7250 206D6172 45422020 0D746964 : Program...BEdit.
00008F30 : F41E1E00 74754120 20726F68 4D202020 : ...ô.Author....M
00008F40 : 20656B69 6D6D7553 6F747265 28000D6E : ike.Summerton..(
00008F50 : 5620F415 69737265 20206E6F 31204120 : .ô.Version...A.1
00008F60 : 000D302E 20F41A32 43534952 65735520 : .0..2.ô.RISC.Use
00008F70 : 614D2072 20686372 30393931 243C000D : r.March.1990..<$
00008F80 : 725020F4 6172676F 2020206D 6A627553 : ô.Program...Subj
00008F90 : 20746365 43206F74 7279706F 74686769 : ect.to.Copyright
00008FA0 : 0546000D 50000D3A 2099C815 756F4822 : ..F.:..P.È ."Hou
00008FB0 : 616C6772 4F5F7373 000D226E 69F20F5A : rglass_On"..Z.òi
00008FC0 : 6974696E 73696C61 64000D65 2099C82E : nitialise..d.È .
00008FD0 : 6D695722 6E495F70 61697469 6573696C : "Wimp_Initialise
00008FE0 : 30322C22 34262C30 34333542 2C343531 : ",200,&48534154,
00008FF0 : 64454222 0D227469 F2106E00 74736E69 : "BEdit"..n.òinst
00009000 : 696C6C61 0D6E6F63 C8167800 48222099 : allicon..x.È ."H
00009010 : 6772756F 7373616C 66664F5F 82000D22 : ourglass_Off"..
00009020 : 6975710C 203D2074 8C000DA3 000DF505 : .quit.=.£.. .õ..
:4   :0
```



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

SCREEN FADE ROUTINES
A selection of some professional screen fading routines to use with your displays.

SPYGLASS: A MULTIPLE TASK MEMORY VIEWER
A multi-tasking memory viewer which allows you to select any running application and display in a window the contents of the memory allocated to it.

A GOOD IDEA
A look into the new range of Integrated Drive Electronics disc drives.

STICKY BACKDROP
A very versatile implementation of the sticky backdrop concept, which allows you to stick file and directory icons onto the Desktop for quick access.

QUITE A REVELATION
A review of Revelation, the sophisticated image processing application from Longman Logotron.

MULTI-TASKING WIMP DEBUGGER
A multi-tasking application which allows you to debug Wimp based programs while testing and running them.

WP/DTP
A regular column which offers advice on using different DTP and WP packages.

PROGRAMMER'S WORKSHOP
A major series covering all advanced aspects of the Archimedes and incorporating the Assembler Workshop. This month: an ADFS disc examiner.

USING ANSI C
A new series on programming Desktop applications in C.

MASTERING THE WIMP
A major and very popular series on the Wimp programming environment. The most recent installment is devoted to accessing printer drivers.

INTO THE ARC
A regular series for beginners currently unveiling the basic features of the RISC operating system.

ARCADE
A round-up of the latest games for the Archimedes: Iron Lord, Twin World and Powerband.

SUBSCRIPTION DETAILS

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

Destination	Additional Cost
UK,BFPO &Ch Is	£ 9.10
Rest of Europe and Eire	£14.00
Middle East	£17.00
Americas an d Africa	£19.00
Elsewhere	£20.00

POSTBAG

POSTBAG

MORE THOUGHTS ON THE NEW MASTER MOS

Having been an avid reader of your magazine for a number of years, and reading of the new Master MOS ROM, I purchased one. To my horror I now find that Spellmaster will not give instant spelling corrections; if it does text is lost. Also your own SWR loader program (in the series *Software for Sideways RAM*, BEEBUG Vol.5 Nos.2 to 4) does not now initialise ROM images, and I have to resort to manual initialisation.

On the 'up' side *The Publisher*, *Starbase*, *Advanced Teletext System*, *OverView* and *Dumpout 3* all work under the new MOS as far as I can see.

Other than this the new MOS has a lot going for it, though I dare say that there will be other problems in the future.

D.R.J.Ockelford

Regarding David Holmes' letter in *Postbag* Vol.9 No.9, I have tried the two problems he mentioned using the new Master MOS ROM and the BEEBUG Master ROM. I can confirm that Spellmaster will appear in two ROM positions (6 & 7) when using the *P command with the BEEBUG Master ROM. With regard to the *MERGE command, this still appears to be operative. The problem may be due to another ROM in the machine.

The new MOS ROM appears to be designed mainly for ADFS usage, and the DFS has been relegated to a secondary position. From my point of view, Acorn has cured some problems and made improvements with this ROM, particularly for ADFS use, but caused problems with the DFS.

D.R.Clarke

Some ROMs such as Spellmaster have always had something of a reputation for their ability to interfere with some other aspect of your system. It is also probable, that not all software was written in strict conformity with Acorn's guidelines, and thus 'short cuts' may now fall foul of the new MOS.

Readers need to consider carefully before installing the new Master MOS ROM, though the benefits in improved performance do seem worthwhile for ADFS users (and I do recommend all Master users to convert to ADFS format if they have not done so). However, most BBC software is now well into its old age (though still performing an excellent job), and it is therefore unlikely that many software houses would feel that they can justify devoting effort to such software if problems now arise.

MORE PLUS THAN MINUS

A recent issue of BEEBUG (Vol.8 No.6) carried a program for printing a plus-or-minus sign, and in a later issue (Vol.9 No.9) a reader suggested using the underline facility for the same result.

I have used embedded commands in InterWord to obtain a similar result for Epson or Epson compatible printers. For one occurrence, the sequence looks tedious, but if required several times, the copy facility of InterWord makes life quite pleasant. Proceed as follows:

1. Press f3
2. Print the '+' sign
3. Press f1
4. Press Return
5. Enter 8,27,83,1
6. Press Escape
7. Enter the '-' sign
8. Press f1
9. Press Return
10. Enter 27,84
11. Press Escape
12. Press f3

This uses subscript mode for the minus sign. For subsequent entry of the plus-or-minus sign just press f9.

P.A.Vedamuttu

Personal Ads

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

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

BBC Master 512K co-processor & GEM interface, dual 40/80T switchable drives in plinth, green screen monitor £480 o.n.o. C.C. Interword, Spellmaster, Acornsoft Viewstore £15 each, 1st Word Plus for Archimedes £20. Tel. (0462) 459338 eves.

Micro Users etc. to '88 £5, Excellent Beeb text books £10, many tapes & software (1.2 OS) including Elite, Snowball and some educational £25 or £30 the lot. (0707) 323736.

Master 128, Taxan 625 supervision colour monitor, Watford 5.25" DD 1Mb 3.5" DD, Epson FX80+ printer, joystick, Mini Office, Spellchecker, View reference manuals etc. all for £800 o.n.o. Tel. 031-443 5398 eves.

BBC Master 512 plus manuals & software, twin 40/80T switchable plinth mounted disc drive & PSU, Master 512 User Guide, Problem Solver & two Essential Software discs, ADT & Master Adv. 1770 DFS, BEEBUG mags & discs, some BBC games, Uviprom 16/32 Eprom programmér & software, two Care Master ROM cartridges £400 & carriage, will split. Tel. (0704) 0545590 after 6pm.

BBC B issue 7 computer with DFS twin 40/80T disc drives,Sideways ROM fitted into a Viglen console unit, AMX Super Mouse, Cumana Sketch Pad, View Word Processor, many tapes, discs and books £300 o.n.o. Tel. (0372) 375438 eves.

BBC B issue 4 with DFS £110, Universal 2nd Processor (unused) £23. Tel. 081-858 6086 eves & wk/ends.

Acorn Teletext adaptor, Advanced Teletext System, BBC - book & tape; Hobbit, Linkword German. Any reasonable offers accepted. Write to; Ms G Bobbett, 81 The Slayde, Yarm, Cleveland TS15 9HZ.

M128 with Plinth, includes Spellmaster, Lisp and Basic Booster ROMs and ROM cartridges, manuals etc. £275, 512 co-processor expanded to 1Mb, includes Essential Software utilities £150, two 80T disc drives with power supplies £100, Microline 80 printer with tractors £25, Microvitec monitor £75. Tel. (0276) 32358.

M128 computer £260, dual 40/80T disc drives mounted in plinth £125, Philips CM8833 colour med. res. stereo monitor £150, Morley A-A ROM board for master £30, Datachart 1223 modem with software and lead for BBC £45, Watford co-pro adaptor for external connection of Turbo or 512 board to Master or B £30, ROMs; Spellmaster £30, Exmon II £15, Viewstore £25, Viewplot £12, printer monitor £7, Discs; Dumpmaster £5, Masterfile £8, Quickcalc £7, View & Viewsheet manuals £5 each, Acorn DTP for Arch/A3000 £80. Tel. 051-606 0289.

Viewstore 1.2 £20, Microsoft EXCEL 2.1 (MS-DOS) unused £55. Tel. (0902) 20768.

BBC B and Master 128 software and hardware from £1.50. Please write enclosing s.a.e. for comprehensive list to; Douglas Bell, 28 Adam Crescent, Stenhousemuir, Larbert FK5 4DQ.

BBC Master 128 with 512 board, DOS+2.1, ROMs include BEEBUG Master, WYSWIG, Questmouse, Spellmaster and COMSOFT, 2 disc drives, low res. monitor, modem, 50 capacity disc box, 100 capacity disc box, and 7x10 capacity disc boxes, over 150 floppy discs, 11 BBC manuals for £500 o.n.o. Tel. (0928) 569686.

M128 with Philips green monitor, AMS 3" twin drive, Viewstore ROM, Interword ROM, Wordwise ROM, plus spare discs, cartridges manuals etc. £285. Tel. (0689) 857403.

BEEBUG Printwise disc and manual, Dabs Hyperdriver ROM and manual, Merlin database ROM, disc and manual, Micro Aid Cashbook disc and manual, Acornsoft Viewsheet ROM and manual, no reasonable offer refused. Tel. (0705) 371018.

M128 complete system with Welcome and Reference manuals 1&2, View, Viewsheet, Viewstore, Interword, Spellmaster, 2 Care ROM cartridges, Philips green monitor, Akhter twin 40/80 DS DD, Panasonic KX-P1080 dot matrix printer £500. Tel. (0705) 594845.

M128, 512 co-processor plus Superior Software 512 upgrade including software and mouse as supplied, Teletext adaptor & ROM, dual 40/80 double sided disc drive, Overview complete, Dab's 512 guide & disc, 2 volumes of Dab's shareware discs for the 512, Dab's Viewsheet/Viewstore guide & disc, Shibumi Problem Solver disc, Essential Software's Utilities disc, Essential Software's CDFS ROM & disc (allows you to use all 2nd processor memory as a RAM disc when in normal BBC mode), AMX mouse, Numerous programming books including Dab's Guide to C: Forth etc. Offers accepted for the complete system or for any part thereof (I have upgraded my system). Tel. (0274) 878881 for full details or to make offers.

BBC model B issue 7, 1770 DFS with manuals, quick cash sale £135. Tel. 081-876 5347 eves.

WANTED: Cumana 20Mb hard disc (CAB20) for BBC Master, can collect reasonable distance (Kent). Tel. (0227)454204.

WANTED: Copy of Spellcheck III Utility disc (with the facility to display words on Dictionary Disc). Tel. (0673) 60892.

WANTED: AMX Mouse including artwork and EPROM, The Graduate (G400, G800) (IBM compatible, 8088 processor, MS-DOS), Printmaster, Graph Pad, Sideways RAM (128k, Solidisk), Starstick ROM, Disc Doctor, Lightpen (with software/handbook), Elite (5.25" disc only), Video camera interface, software to run a graphic plotter. Please send a list of hard/software you wish to sell to; Lystad, Geir Morten, Saupstadringen 75a, 7078 Saupstad, NORWAY. Ps. Please send list with prices and remember your address so I can write to you.

M128 £240, 40/80T twin DD bridging unit £140, Wordwise Plus, Spellmaster, Dumpmaster and Master ROMs £20 each, 512 upgrade, mouse & GEM £75, Epson LX-80 printer £75, BEEBUG mags vols.1-8 £5 per vol. Tel. (0634) 668185.

WANTED: Advice on installing, using, experience of Integra-B board in model B. Tel. (0753) 884360.

WANTED: Med. res. colour monitor, also Disc Filing System User Guide (1770 DFS) by Acorn. Tel. 081-550 8294.

BBC B issue 7 with manuals, fitted with Watford 32k Shadow RAM card and manual, ATPL Sideways ROM/RAM expanison board, BBC Master with manuals, Cumana double disc drive 40/80T switchable with manuals, Microvitec Cub 14" colour monitor, Kaga Taxan 810 24" dot matrix printer and manual. No reasonable offer refused. Tel. (0705)371018.

Fractals: Shareware/PD programs for BBC-M or 512 wanted (have all BEEBUG's). Tel. 092-575 4783 eves.

Acorn User Jan '86 - June '90 £25, **Micro User** July '85 - April '90 £30, BEEBUG Vols. 4-9 inc. £40. Tel. (0202) 699500 after 5pm.

M128 as new £250, Micronix twin disc drive 5.25" 40/80T switchable built in PSU £100, Philips green screen monitor £15, works fine but no documentation, Instant Mini Office ROM £25, all post extra or collect. Tel. 091-581 9989.

BEEBUG mags Vol. 1 (1-10) Vol. 2 (1-10) complete each volume £4.50, BEEBUG magazine cassettes Vol. 2 Nos, 7,9, 10. Vol. 3. 1,2,4,5,6,8,9,10. Vol. 4. 3,9. Program Builder, Quickcalc, Masterfile, BBC Welcome cassette all with inst. 60p each. Acorn cassette data recorder complete with leads £7.50, 2x 1.5 parallel printer leads £3 each. Books; Assembly Language Programming for BBC Micro - Ian Bimbaum £3.50, BBC Micro Compendium - Jeremy Ruston £5, Advanced User Guide (well used) £2.50 p&p extra. Tel. (0530) 243949.

BBC Master 512 c/w Microvitec 1431 RGB monitor, combo 5.25"/3.5" DSDD disc drives in Master bridge unit with PSU, mouse and all original 128 and 512 software, plus Viewstore, Interword and Wordwise Plus ROMs, 2 ACP ROM cartridges, all handbooks, inc. Master Reference manuals 1&2, Dab's Viewsheet/Viewstore UG, Dab's 512 UG and disc, GEM Desktop, Write, Paint and Draw manuals plus BEEBUG Vols. 3-8 (in binders) and Vol. 9 (unbound) £350 the lot, also twin 5.25" DS DD disc drive with PSU £80, and twin 3.5" DS DD drive, computer powered £60, everything in perfect order. Tel. (0380) 870938 after 6pm.

Inter Mega 3 ROM, consisting of Interword, Intersheet and Interchart £50, Interbase ROM £35, Apricote Accounts book V3 + Invoice V2 ADFS disc £25, all packaged as new. Tel. 081-508 1737.

Brother HR5 thermal and plain paper printer, serial BBC lead, thermal paper, instruction manual, ribbons and mains adaptor, excellent condition £15. Honeywell L11 tractor-feed printer, parallel 'Centronics' BBC lead unused, spare ribbons & manual, excellent condition, very robust £20. plus postage. Tel. (0536) 68491.

Master 128 £275, for BBC B Watford 128k ROM/RAM card all options installed £75, Watford 32k Shadow RAM/printer buffer £40. Tel. (0474) 362114 after 8pm.

BBC B with Acorn DFS, Microvitec 1413 colour monitor, 40T DS DD, Olivetti sparkjet printer, complete with leads, books and programs. Tel. (0924) 261120 eves only.

Music software, complete music performing/printing package for the BBC B/Master, this hardly used package includes EMR Miditrack Editor, Performer, Scorewriter, Utilities, Midi Interface and all manuals, may split, offers around £230. Tel. (0705) 371018.

M128, Teletext adaptor, 40T disc drive, Microvitec 1431 monitor, Taxan-Kaga printer, AMX Art, Mouse, Hyperdriver, Master ROM, Commstar II, discs, ref. manuals 1&2, View guide, lots of books, all leads etc. just switch on and go, all this for only £600. Tel. 071-328 0707.

Z88 + 32k RAM, PC Link II and parallel printer lead, as new £185, Publisher DTP pack £35, AMX Art and Pagemaker (BBC and Master versions) £25, Arc Wordwise Plus £10, Commstar II £10. Tel. (0736) 63918.

WANTED: Acorn User Calligraphy disc, also Masterfile II 5.25" ADFS, already have DFS version. Tel. (0525) 715013 after 6pm.

FREE to collector 40T double sided disc drive with PSU (0525) 715013 after 6pm.

BBC software to sell, Games, Utilities etc. all originals, some unused, most less than 1/2 price. Send s.a.e. for list to; S Shar, c/o 19 Elvednon Road, Palmers Green, London N13 4SJ.

Swap; Epson LQ800 24 pin printer with parallel and serial interfaces, tractor unit, three new ribbons, all manuals and original packing, in excellent condition **for;** Juki 6200/6300 daisywheel printer with either tractor unit or sheetfeeder, (new square shape) also ESM colour screenprint ROM for Star LC10 printer swap for Computer Concepts Spellmaster. Tel. (0900) 68231 after 6pm.

Computer Concepts Intersheet, ROM version for BBC Master or B, unused and unregistered £20, Acorn Hi View for use with 6502 second processor, new unused £10. Tel. 086 732 8776.

BEEBUG MEMBERSHIP

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

BEEBUG SUBSCRIPTION RATES

		BEEBUG & RISC USER
£18.40	1 year (10 issues) UK, BFPO, Ch.I	£27.50
£27.50	Rest of Europe & Eire	£41.50
£33.50	Middle East	£50.50
£36.50	Americas & Africa	£55.50
£39.50	Elsewhere	£59.50

BACK ISSUE PRICES (per issue)

Volume	Magazine	5"Disc	3.5"Disc
5	£0.50	£2.00	£2.00
6	£0.50	£2.00	£2.00
7	£1.00	£2.00	£2.00
8	£1.20	£3.00	£3.00
9	£1.60	£4.75	£4.75
Binders	£4.20		

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

POST AND PACKING

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

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

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

BEEBUG MAGAZINE is produced by BEEBUG Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Technical Editor: Alan Wrigley
Technical Assistant: Glynn Clements
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Managing Editor: Sheridan Williams

The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

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

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

In all communication, please quote your membership number.

ISSN - 0263 - 7561

Printed by Arlon Printers (0923) 268328

Magazine Disc

May 1991
DISC CONTENTS

PRINTING SCIENTIFIC CHARACTERS WITH WORD PROCESSORS (1) - Two programs which help you design scientific characters: GridPrint creates any matrix up to size 9x12 and Analysis gives you the definition codes of any character resident in the computer.

MINI WALL PLANNER - print out your own personalised wall planner with an index of important dates alongside it.

VIEW TO ASCII - a program which allows you to convert View word processing files into ASCII format, which can than be imported into other applications such as DTP or another Word Processor.

CROSSWORD COMPILER - this program allows you not only to design your own crosswords, but also to enter the answers and print the crosswords out.

CONVERTING ASCII TO BINARY - this program converts ASCII text files into binary files which can be useful for generating printer control code sequences.

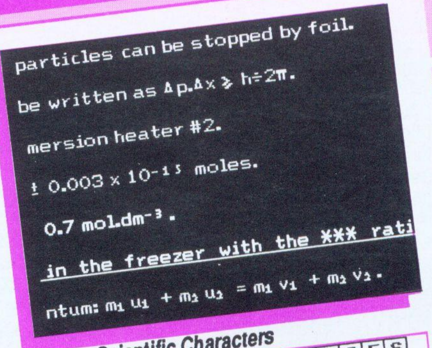
RECREATIONAL MATHEMATICS: EUCLID'S ALGORITHM AND MODULAR ARITHMETIC - two programs: the first displays Euclid's algorithm for finding the greatest common divisor and the lowest common multiple, and the second program Galois demonstrates modular arithmetic.

MR TOAD'S FONTZ ROM - a short Sideways ROM image which provides eleven different typefaces on the Master.

BEEBUG WORKSHOP: LINEAR EQUATIONS - a program using the Gaussian Elimination algorithm for solving systems of linear equations.

BEEBUG FUNCTION/PROCEDURE LIBRARY (PART 3) - a Basic program containing the latest two additions - a generalised input function, and a general purpose menu screen.

MAGSCAN DATA - bibliography for this issue (Vol.10 No.1).



particles can be stopped by foil.

be written as $\Delta p.\Delta x \geqslant h \div 2\pi$.

mersion heater #2.

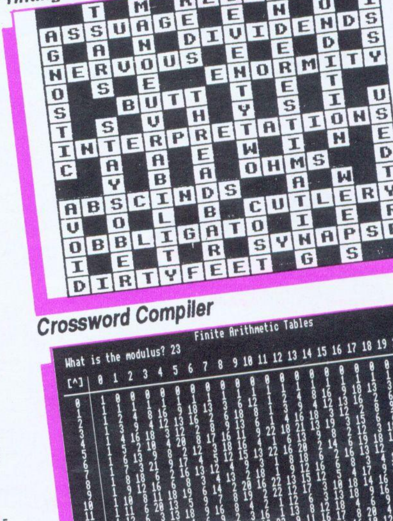
$\pm$ 0.003 x 10^{-15} moles.

0.7 mol.dm^{-3}.

in the freezer with the *** rati

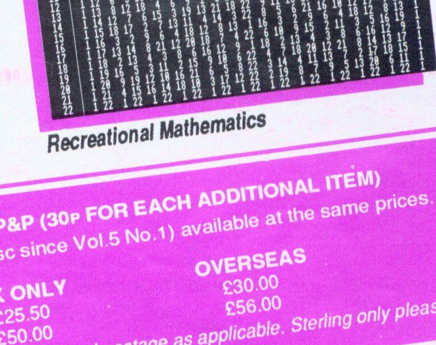
ntum: $m_1 u_1 + m_2 u_2 = m_1 v_1 + m_2 v_2$.

Printing Scientific Characters



```
    T   M   R E S O U R C E S
A S S U A G E   E   N   O   I
G   A   N   D I V I D E N D S
N E R V O U S   E   E   D   S
O   S   E     E N O R M I T Y
S     B U T T   T   E   T
T   S   V   H   Y   S   I   U
I N T E R P R E T A T I O N S
C   A   A   E   W   I   N   E
    Y   B   A   O H M S     D
A B S C I N D S     A   W   T
V   O   L   B   C U T L E R Y
O B B L I G A T O   I   E   R
I   E   T   R   S Y N A P S E
D I R T Y F E E T   G   S
```

Crossword Compiler



```
                      Finite Arithmetic Tables
What is the modulus? 23
[^] |  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
----+---------------------------------------------------------------------
  0 |  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  1 |  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
  2 |  1  2  4  8 16  9 18 13  3  6 12  1  2  4  8 16  9 18 13  3  6 12  1
  3 |  1  3  9  4 12 13 16  2  6 18  8  1  3  9  4 12 13 16  2  6 18  8  1
  4 |  1  4 16 18  3 12  2  8  9 13  6  1  4 16 18  3 12  2  8  9 13  6  1
  5 |  1  5  2 10  4 20  8 17 16 11  9 22 18 21 13 19  3 15  6  7 12 14  1
  6 |  1  6 13  9  8  2 12  3 18 16  4  1  6 13  9  8  2 12  3 18 16  4  1
  7 |  1  7  3 21  9 17  4  5 12 15 13 22 16 20  2 14  6 19 18 11  8 10  1
  8 |  1  8 18  6  2 16 13 12  4  9  3  1  8 18  6  2 16 13 12  4  9  3  1
  9 |  1  9 12 16  6  8  3  4 13  2 18  1  9 12 16  6  8  3  4 13  2 18  1
 10 |  1 10  8 11 18 19  6 14  2 20 16 22 13 15 12  5  4 17  9 21  3  7  1
 11 |  1 11  6 20 13  5  9  7  8 19  2 22 12 17  3 10 18 14 16 15  4 21  1
 12 |  1 12  6  3 13 18  9 16  8  4  2  1 12  6  3 13 18  9 16  8  4  2  1
 13 |  1 13  8 12 18  4  6  9  2  3 16  1 13  8 12 18  4  6  9  2  3 16  1
 14 |  1 14 12  7  6 15  3 19 13 21 18 22  9 11 16 17  8 20  4 10  2  5  1
 15 |  1 15 18 17  2  7 13 11  4 14  3 22  8  5  6 21 16 10 12 19  9 20  1
 16 |  1 16  3  2  9  6  4 18 12  8 13  1 16  3  2  9  6  4 18 12  8 13  1
 17 |  1 17 13 14  8 21 12 20 18  7  4 22  6 10  9 15  2 11  3  5 16 19  1
 18 |  1 18  2 13  4  3  8  6 16 12  9  1 18  2 13  4  3  8  6 16 12  9  1
 19 |  1 19 16  5  3 11  2 15  9 10  6 22  4  7 18 20 12 21  8 14 13 17  1
 20 |  1 20  9 19 12 10 16 21  6  5  8 22  3 14  4 11 13  7  2 17 18 15  1
 21 |  1 21  4 15 16 14 18 10  3 17 12 22  2 19  8  7  9  5 13 20  6 11  1
 22 |  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1 22  1
```

Recreational Mathematics

ALL THIS FOR £4.75 (5.25" & 3.5" DISC) + 60P P&P (30P FOR EACH ADDITIONAL ITEM)

Back issues (5.25" disc since Vol.3 No.1, 3.5" disc since Vol.5 No.1) available at the same prices.

DISC (5.25" or 3.5") SUBSCRIPTION RATES	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.

BEEBUG, 117 Hatfield Road, St.Albans, Herts AL1 4JS

Special Offers to BEEBUG Members May 1991

BEEBUG'S OWN SOFTWARE

Code	Product	Members Price inc Vat
1407a	ASTAAD3 - 5" Disc (DFS)	5.95
1408a	ASTAAD3 - 3.5" Disc (ADFS)	5.95
1404a	Beebug Applics I - 5" Disc	4.00
1409a	Beebug Applics I - 3.5"Disc	4.00
1411a	Beebug Applics II - 5" Disc	4.00
1412a	Beebug Applics II - 3.5" Disc	4.00
1600a	Beebug magazine disc	4.75
1405a	Beebug Utilities - 5" Disc	4.00
1413a	Beebug Utilities - 3.5" Disc	4.00
1450a	EdiKit 40/80 Track	5.75
1451a	EdiKit EPROM	7.75
1452a	EdiKit 3.5"	5.75
0005b	Magscan Vol.1 - 8 40 Track	12.50
0006b	Magscan Vol.1 - 8 80 Track	12.50
0011a	Magscan Update 40 track	4.75
0010a	Magscan Update 80 track	4.75
PAG1a	Arcade Games (5.25" 40/80T)	5.95
PAG2a	Arcade Games (3.5")	5.95
PBG1a	Board Games (5.25" 40/80T)	5.95
PBG2a	Board Games (3.5")	5.95
0077b	C - Stand Alone Generator	14.56
0081b	Masterfile ADFS M128 80 T	16.86
0024b	Masterfile DFS 40 T	16.86
0025b	Masterfile DFS 80 T	16.86
0074b	Beebug C 40 Track	45.21
0075b	Beebug C 80 Track	45.21
0084b	Command	29.88
0073b	Command(Hayes compatible)	29.88
0053b	Dumpmaster II	23.76
0004b	Exmon II	24.52
0087b	Master ROM	29.88
1421b	Beebug Binder	4.20

OTHER MEMBERS' OFFERS

These offers are available for one month from publication. To avoid disappointment please order early, as offers are only available while stocks last and demand is always high. Orders are dispatched on a first come first served basis. To order phone (0727) 40303 or write to BEEBUG, 117 Hatfield Rd, St.Albans, AL1 4JS.

ViewStore

£14.10 (inc VAT) + p&p

Normal Members price £42.49 (inc VAT)

ViewStore is Acorn's database manager program. It offers 40 and 80 column display, multiple indexes, detailed report generation, variable field lengths and much more. Excellent value for money.

Stock code 1019b

ViewSheet

£12.77 (inc VAT) +p&p

Normal Members price £42.49 (inc VAT)

Excellent spreadsheet for the BBC micro and Master, supplied on a 16K ROM. A very powerful and extremely useable product which will produce results ready to print or for direct merging into View text files.

Stock code 1001b

ViewSpell

£7.16 (inc VAT) + p&p

Normal Members price £28.75 (inc VAT)

An automatic spelling checker with a built-in 75 000 word dictionary. Supplied on ROM with full manual, examples disc and reference card. Ideal for View or ASCII text files, and can use updateable user dictionaries.

Stock code 1043b

FORTH

£3.06 (inc VAT) + p&p

Normal Members price £20.00 (inc VAT)

FORTH language on ROM (no manual).

Stock code 1041a

Post & Packing	UK	Europe	Americas, Africa Middle East	Elsewhere
a	£ 1.00	£ 1.60	£ 2.40	£ 2.60
b	£ 2.00	£ 3.00	£ 5.00	£ 5.50
c	£ 3.10	£ 6.50	£10.50	£11.50
d	£ 3.60	£15.50	£24.50	£25.50

LISP

£3.06 (inc VAT) + p&p

Normal Members price £20.00 (inc VAT)

LISP language on ROM (no manual).

Stock code 1003a