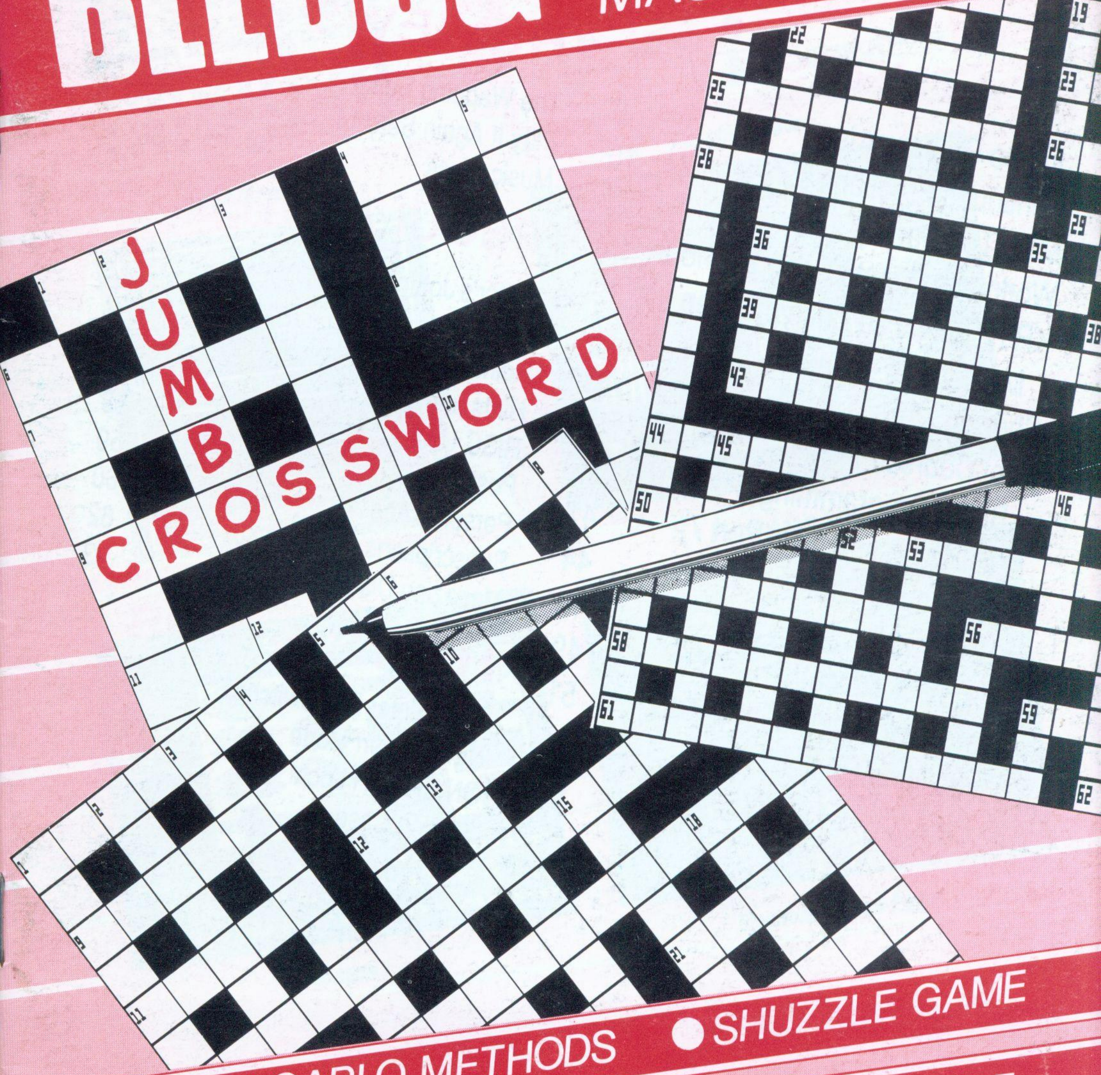
Vol.9 No.7 December 1990

BEEBUG

FOR THE BBC MICRO & MASTER SERIES



- MONTE CARLO METHODS
- SHUZZLE GAME
- WAPPING EDITOR
- MIKROTEL DATABASE

BEEBUG Vol.9 No.7 December 1990

FEATURES

REVIEWS

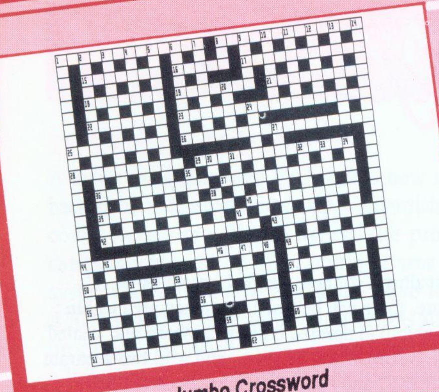
REGULAR ITEMS

HINTS & TIPS

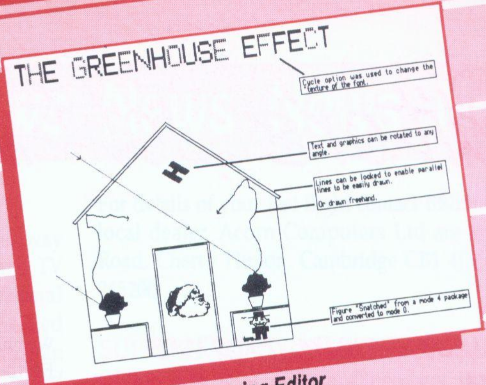
PROGRAM INFORMATION

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



Jumbo Crossword



THE GREENHOUSE EFFECT

Cycle option was used to change the texture of the font.

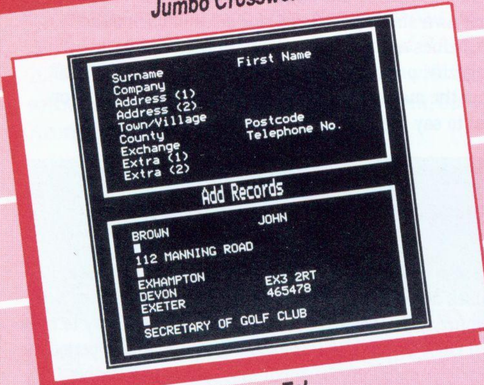
Text and graphics can be rotated to any angle.

Lines can be locked to enable parallel lines to be easily drawn.

Or drawn freehand.

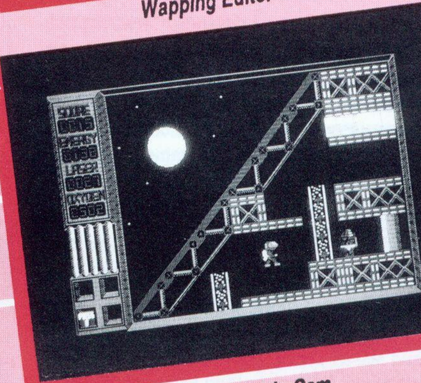
Figure 'Snatched' from a mode 4 package and converted to mode 0.

Wapping Editor

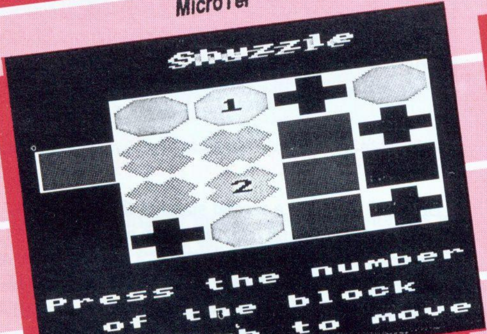


```
Surname            First Name
Company
Address (1)
Address (2)
Town/Village       Postcode
County             Telephone No.
Exchange
Extra (1)
Extra (2)
            Add Records
                   JOHN
BROWN

112 MANNING ROAD

EXHAMPTON          EX3 2RT
DEVON              465478
EXETER

SECRETARY OF GOLF CLUB
```

MicroTel



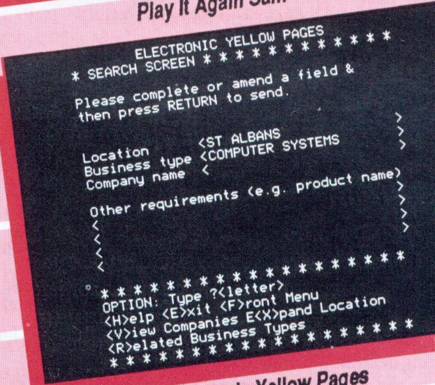
Play It Again Sam



Shuzzle

Press the number of the block you wish to move

Shuzzle



```
          ELECTRONIC YELLOW PAGES
* SEARCH SCREEN * * * * * * * * * * * * *

Please complete or amend a field &
then press RETURN to send.
                                        >
Location      <ST ALBANS                >
Business type <COMPUTER SYSTEMS         >
Company name  <
                                        >
Other requirements (e.g. product name)
                                        >
<                                       >
<                                       >
<
<
* * * * * * * * * * * * * * * * * * * * *
OPTION: Type ?<letter>
<H>elp <E>xit <F>ront Menu
<V>iew Companies E<X>pand Location
<R>elated Business Types
* * * * * * * * * * * * * * * * * * * * *
```

Electronic Yellow Pages

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 will not function on a cassette-based system.

Program needs at least one bank of sideways RAM.

Program is for Master 128 and Compact only.

Editor's Jottings

ENTERTAINMENT FOR THE CHRISTMAS SEASON

As this is the last issue for 1990, and our pre-Christmas issue, we have put more emphasis than usual on entertainment and leisure in this issue of BEEBUG. Even serious computer users need time to relax occasionally. As well as a most puzzling and challenging game (Shuzzle), we have also included an item on recreational mathematics (Continued Fractions), and a versatile program for replicating and designing crosswords.

Of course, you will also find a varied range of more serious and technical items in this issue, with (we hope) something for everyone, and all entirely dedicated to the BBC micro and Master series. The next issue of BEEBUG is a two month issue, for January and February, and we expect to mail this out in early January 1991.

NEW PUBLISHING VENTURE

As a new venture, we are publishing a book with the title *File Handling for All*, based on a series of articles first published in BEEBUG magazine. All the original articles have been thoroughly edited and updated, with much additional material and a supporting disc. Again, more details are given elsewhere in this issue. The book, which runs to over 120 pages, is scheduled for release this December and will retail at £9.95 (see advert on page 9 for ordering details). If file handling has puzzled you until now, or you are seeking ideas on more advanced techniques, then this is the book for you.

COMPUTER SHOPPER SHOW

Last year's *Computer Shopper Show* proved so successful that local radio put out appeals on the final day for the public NOT to attend. This year's event has moved to much larger premises at the Wembley Conference Centre (close by Wembley stadium), so there should be room for all. Furthermore, there will be a self-contained area within the main exhibition devoted entirely to Acorn and related companies. BEEBUG will be there, and with two separate stands. BEEBUG and RISC User magazines will have their own stand (stand J21) for sales of back issue magazines and magazine discs, and other related magazine products. You will also be able to meet staff from the magazine department for help and advice, or just to say 'Hello'.

NEW GAMES DISCS

We also expect to have available not one but two new games discs for the BBC micro, each disc containing a compendium of eight first rate games, an ideal way to relax and enjoy yourself over the Christmas/New Year period, or maybe to entertain the children. You will find further details elsewhere in this issue, and you will, of course, be able to see these new releases at the Computer Shopper Show.

BEEBUG HARDWARE AND SOFTWARE

BEEBUG will have a second stand at the Computer Shopper Show (stand J25/26) which will be devoted to the demonstration and sale of BEEBUG's growing range of own hardware and software products. We will also have a wide range of other products on sale, with staff to advise you on the best purchase for your needs.

REDUCED PRICE ENTRANCE TO SHOW

As further encouragement to members, we are including in this issue of BEEBUG a voucher which will entitle you to 50p off the normal entry price for the Computer Shopper Show. See you there.

News News News News News News

TEST FLY AN ARCHIMEDES

A major sales campaign by Acorn is now under way backed by nationwide newspaper advertising and TV commercials in selected areas. The promotional campaign focuses on *The Learning Curve* packaged systems which are now available in two formats; an A3000 based system with applications discs, word processor (1st Word Plus), PC Emulator (DR DOS), educational software (Genesis multi-format information system), and supporting materials; or the A420 based system, featuring 2MBytes of RAM and 20MByte hard disc as standard, together with all the software and support as for the A3000 system, but with Acorn DTP (Desktop Publisher) as well.

The A420 Learning Curve System

The Test Flight initiative invites potential customers to visit their local dealer to try out the systems with the option of a 14 day home trial (A420 systems only). Customers can also try out their aviation skills on Interdictor, Clares' flight simulator, and there is a competition with major prizes following the aeronautical theme - trip by Concorde, flying lessons for helicopter and fixed wing aircraft, and more.

BEEBUG will be fully participating and supporting this campaign. In addition all systems purchased through BEEBUG will be enhanced with additional memory/hard disc capacity as appropriate at no extra charge.

For details of your test flight contact BEEBUG, or your local dealer. Acorn Computers Ltd are at Fulbourn Road, Cherry Hinton, Cambridge CB1 4JN, tel. (0223) 245200.

SUPPORT FOR WORDWISE PLUS USERS

CORPLAN is a software product intended to complement Computer Concepts' Wordwise Plus word processor (or the Wordwise Plus 2 version supplied by Ifel). It also provides useful support for Interword.

CORPLAN is a document management system written in the Wordwise Plus programming language. Up to 22 user-designed and menu-selectable document layouts can be held in one CORPLAN system. Address files can be maintained and merged with standard letters, and there are many utilities handling a variety of useful and time-saving tasks, including printer control.

CORPLAN is available in 40 or 80 track DFS disc formats, or 5.25" and 3.5" ADFS formats, and costs £19.50 inclusive, post free in the UK. Contact CORPLAN Computer Systems, Three Gables, 7A Talbots Drive, Maidenhead, Berks SL6 4LZ, tel/fax (0628) 24591.

ALL FORMATS COMPUTER FAIR

The next *All Formats Computer* at the New Horticultural Hall, Westminster will take place on 15th December, admission £3. Already the first fair for 1991 has been booked at the same venue for 2nd February, with further fairs through the year to come.

Advance tickets, and further details, are available from Mike Hayes, 8 Midgrove, Delph, Oldham OL3 5EJ, tel. (0457) 875229.

NEW DISTRIBUTOR FOR MIRRORSOFT

Mirrorsoft computer software titles including *Fleet Street Editor* for the BBC micro and Master series will in future be distributed by Software Toolworks. This in turn is the brand name for the business applications products of Mindscape International, P.O.Box 1019, Lewes, East Sussex, tel. (0444) 86545.

Jumbo Crosswords

Use this program by Richard Jones to print out duplicates of published crosswords, or to design your own.

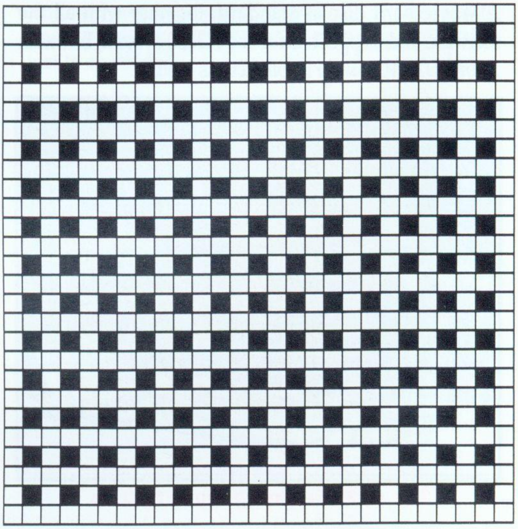
This program arose out of the habit adopted by several newspapers of publishing crossword puzzles with two sets of clues using the one grid. In our family, and no doubt in others, a certain amount of friction is generated, as different members of the family enjoy tackling the different puzzles.

The program listed here will draw a basic crossword pattern (of 27 by 27 squares by default - see Program Notes on how to change this), and then permit the user to add or remove black squares. Crosswords of other sizes can also be produced quite readily (see Program Notes). The cursor keys are used to move to any position in the crossword, and the Space Bar and Delete used to insert or remove a black square respectively. The usual symmetry adopted by crosswords is automatically preserved by the program. Once a correct pattern has been achieved, the program will, on request, write the numbers for the clues into the relevant squares quite automatically.

This process is relatively time consuming. The reason for this is that every square has to be checked to see if it is black or white. Every white square is then cleared in case any previous clue number had already been inserted. Where appropriate a new clue number is then inserted. Thus the crossword pattern can continue to be edited and renumbered as often as is required. Pressing Escape will abort the current operation (useful if you realise you want to terminate the insertion of clue numbers, or the provision of printed copy - see later), ready for the next action. Any clue numbers already inserted remain.

Once the screen display of the outline crossword is correct, it may be dumped to a printer. It is up to the user to insert the correct call for his own printer dump routine (see line 1020), and to ensure that this is loaded prior to running the program if this is necessary.

The routine used by the author is a proportional one, and prints the screen *down* the paper, in order to obtain the largest size possible. This dump is NOT provided. Other screen printer dumps are equally suitable, but the essential point is that the dump should be a proportional one.



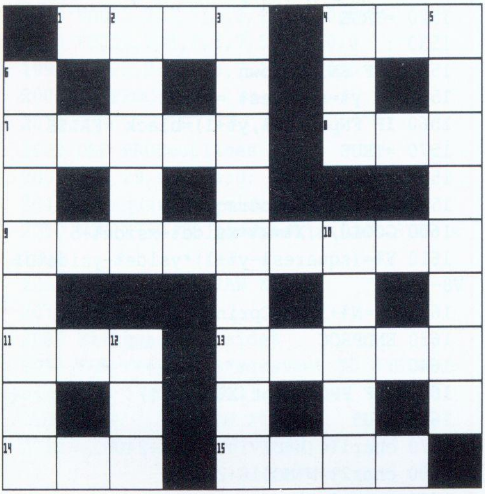
The basic pattern

You can exit from the program at any time by pressing 'Q' for *Quit*.

PROGRAM NOTES

An attempt has been made to make the procedures of the program as clear as possible to other readers by the use of meaningful names and variables. Some specific comments follow:

There are two calls to PROCblock on each of lines 300 and 310 to provide the type of symmetry usual in crossword puzzles. In order to take as little space as possible inside a square, characters 240 to 249 are redefined as small digits 0 to 9 by PROCsmallfigs.



Squares set to nine

The size of the crossword grid can be adjusted by setting a different value for the variable *squares%* (see lines 1760 & 1770). The program will set a suitable value for the size of the individual squares of the grid (see lines 1790 to 1810).

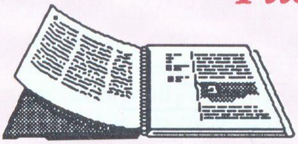
Have fun!

```
   10 REM Program Jumbo
   20 REM Version B1.2
   30 REM Author  R.T.Jones
   40 REM BEEBUG  December 1990
   50 REM Program subject to copyright
   60 :
  100 REM For other sizes alter value of
squares% - see line 1760
  110 MODE 0:ON ERROR GOTO480
  120 PROCinit
  130 PROCoutlines
  140 PROCstandardblocks
  150 ON ERROR GOTO470
  160 REPEAT
  170 VDU23,1,0;0;0;0;
```

```
  180 X%=1:Y%=squares%-1
  190 VDU 29,xside%DIV2;yside%DIV2;
  200 REPEAT
  210 REPEAT
  220 PRINT TAB(0,0)"Use:  Cursor Keys,
 Space Bar or Delete.  N to Number,  D t
o Dump,  Q to Quit";
  230 X%=X%+INKEY(-122)*(X%<squares%+1)-
INKEY(-26)*(X%>0)
  240 X%=X%MODsquares%-squares%*(X%MODsq
uares%=0)
  250 Y%=Y%-INKEY(-42)*(Y%<squares%)+INK
EY(-58)*(Y%>-1)
  260 Y%=Y%MODsquares%-squares%*(Y%MODsq
uares%<0)
  270 PROCmove(X%,Y%)
  280 PROCwait(2)
  290 PROCmove(X%,Y%)
  300 IF INKEY(-99) PROCblock(X%,Y%,1):P
ROCblock(squares%+1-X%,squares%-1-Y%,1)
  310 IF INKEY(-90) PROCblock(X%,Y%,0):P
ROCblock(squares%+1-X%,squares%-1-Y%,0)
  320 UNTIL INKEY(-51) OR INKEY(-17) OR
INKEY(-86)
  330 IF INKEY(-17) M$="Quit" ELSE IF IN
KEY(-51) M$="Dump to printer" ELSE IF IN
KEY(-86) M$="Number squares"
  340 *FX21,0
  350 REPEAT
  360 PRINTTAB(0,0)M$" : Are you sure ?
(Y/N)"SPC(56-LENM$);
  370 G=GET OR&20
  380 UNTIL G=ASC("y") OR G=ASC("n")
  390 PRINT TAB(0,0)SPC(55);
  400 UNTIL G=ASC("y")
  410 C$=LEFT$(M$,1)
  420 IF C$="D" PROCdump ELSE IF C$="N"
PROCnumber
  430 UNTIL C$="Q"
  440 MODE3:PROCquit
  450 END
  460 :
  470 IF ERR=17 THEN GOTO160
  480 MODE3:REPORT:PRINT" at line ";ERL
  490 PROCquit
  500 END
  510 :
 1000 DEF PROCdump
 1010 VDU 26
 1020 REM Insert call to printer dump
 1030 ENDPROC
 1040 :
 1050 DEF PROCquit
 1060 *FX21,0
 1070 *FX4,0
 1080 END
 1090 :
 1100 DEF PROCmove(x%,y%)
 1110 LOCAL X%,Y%
 1120 GCOL 3,1
 1130 X%=x%*xside%-6:Y%=y%*yside%-6
 1140 PROCplotblock(X%,Y%,12,12)
 1150 ENDPROC
 1160 :
 1170 DEF PROCwait(T%)
 1180 t%=TIME
 1190 REPEAT:UNTIL TIME>T%+t%
 1200 ENDPROC
 1210 :
 1220 DEF PROCnumber
 1230 VDU29,-10;0;
 1240 black=FALSE:white=TRUE:N%=0
 1250 FOR y%=1 TO squares%
 1260 FOR sq%=1 TO squares%:x%=sq%+1
 1270 IF FNpeek(x%,y%)<>black THEN VDU29
,xside%DIV2;yside%DIV2;:PROCblock(sq%,sq
uares%-y%,0):VDU29,-10;0;:IF (FNlookleft
=black AND FNlookright=white) OR (FNlook
up=black AND FNlookdown=white) THEN PROC
numbersquare
 1280 NEXT sq%
 1290 NEXT y%
 1300 ENDPROC
 1310 :
 1320 DEF FNpeek(x%,y%)
 1330 LOCAL X%,Y%
 1340 X%=x%*xside%
 1350 Y%=(squares%-y%+1)*yside%-yside%DI
V2
 1360 IF POINT(X%,Y%)=1 =black
 1370 =white
 1380 :
 1390 DEF FNlookleft
 1400 IF sq%=1 =FALSE
 1410 IF FNpeek(x%-1,y%)=black =FALSE
 1420 =TRUE
 1430 :
 1440 DEF FNlookright
 1450 IF sq%=squares% =FALSE
 1460 IF FNpeek(x%+1,y%)=black =FALSE
```

```
 1470 =TRUE
 1480 :
 1490 DEF FNlookup
 1500 IF y%=1 =FALSE
 1510 IF FNpeek(x%,y%-1)=black =FALSE
 1520 =TRUE
 1530 :
 1540 DEF FNlookdown
 1550 IF y%=squares% =FALSE
 1560 IF FNpeek(x%,y%+1)=black =FALSE
 1570 =TRUE
 1580 :
 1590 DEF PROCnumbersquare
 1600 GCOL0,1:X%=x%*xside%-xside%+5
 1610 Y%=(squares%-y%+1)*yside%-yside%DI
V8-1
 1620 N%=N%+1:PROCprint(X%,Y%,N%)
 1630 ENDPROC
 1640 :
 1650 DEF PROCprint(X%,Y%,N%)
 1660 VDU5
 1670 char1%=(N%DIV10)MOD10+240
 1680 char2%=N%MOD10+240
 1690 MOVEX%,Y%
 1700 IF char1%>240 VDUchar1%:PLOT0,-8,0
 1710 VDUchar2%
 1720 VDU4,23,1,0;0;0;0;
 1730 ENDPROC
 1740 :
 1750 DEF PROCinit
 1760 squares%=27:REM 27 for a "TIMES" J
umbo
 1770 IF squares%>33 OR squares%<5 CLS:P
RINTTAB(12,12)"Error!"''"Possible range
for squares%= 5 to 33"''"See line 1800":
PROCquit
 1780 maxht%=972
 1790 xside%=maxht%DIVsquares%
 1800 xside%=(xside%DIV4*4)
 1810 yside%=xside%
 1820 VDU23,1,0;0;0;0;
 1830 VDU19,128,135,0,0,0
 1840 VDU19,1,0,0,0,0
 1850 *FX4,1
 1860 PROCsmallfigs
 1870 ENDPROC
 1880 :
 1890 DEF PROCsmallfigs
 1900 VDU23,240,7,5,5,5,7,0,0,0
 1910 VDU23,241,6,2,2,2,7,0,0,0
 1920 VDU23,242,7,1,7,4,7,0,0,0
 1930 VDU23,243,7,1,7,1,7,0,0,0
 1940 VDU23,244,5,5,7,1,1,0,0,0
 1950 VDU23,245,7,4,7,1,7,0,0,0
 1960 VDU23,246,7,4,7,5,7,0,0,0
 1970 VDU23,247,7,1,1,1,1,0,0,0
 1980 VDU23,248,7,5,7,5,7,0,0,0
 1990 VDU23,249,7,5,7,1,7,0,0,0
 2000 ENDPROC
 2010 :
 2020 DEF PROCoutlines
 2030 VDU 29,xside%;0;
 2040 Y%=squares%*yside%
 2050 FOR X%=0 TO squares%*xside% STEP x
side%
 2060 MOVE X%,Y%:DRAW X%,0
 2070 NEXT
 2080 X%=squares%*xside%
 2090 FOR Y%=yside%*squares% TO 0 STEP -
yside%
 2100 MOVE 0,Y%:DRAW X%,Y%
 2110 NEXT
```

```
 2120 ENDPROC
 2130 :
 2140 DEF PROCstandardblocks
 2150 VDU 29,xside%DIV2;yside%DIV2;
 2160 FOR Y%=squares%-2 TO 0 STEP -2
 2170 FOR X%=2 TO squares% STEP 2
 2180 PROCblock(X%,Y%,1)
 2190 NEXT:NEXT
 2200 ENDPROC
 2210 :
 2220 DEF PROCblock(x%,y%,colour%)
 2230 GCOL 0,colour%
 2240 x%=x%*xside%-xside%DIV2:y%=(y%-.5)
*yside%
 2250 PROCplotblock(x%+2,y%+4,xside%-4,y
side%-8)
 2260 ENDPROC
 2270 :
 2280 DEF PROCplotblock(x1,y1,w,h)
 2290 MOVE x1,y1:MOVE x1+w,y1
 2300 PLOT85,x1,y1+h:PLOT85,x1+w,y1+h
 2310 ENDPROC
```



Continued Fractions for Pleasure

Enter the world of recreational mathematics as Michael Taylor explores the patterns of continued fractions and their convergents.

Many readers of BEEBUG will know that 355/113 is an amazingly good approximation to PI, that 1/(1+1/(1+1/(1+1/(1+....)))) approaches the 'golden ratio' of (1+SQR5)/2 and that this golden ratio is also approached by the ratio of consecutive terms in the Fibonacci series.

These 'facts' of the mathematical world are bound up with *Continued Fractions* and the *Convergents* associated with them. With the program listed here, it is possible to enjoy playing with continued fractions and their convergents, and to investigate their fascinating patterns.

This article is not the place for a treatise on continued fractions - indeed the program is to some extent a substitute for that. Here is a brief explanation, but after playing with the program, the reader may feel urged to explore further. One classic in this field is *The Higher Arithmetic*, by H.Davenport (1952), Cambridge University Press (see Chapter 4).

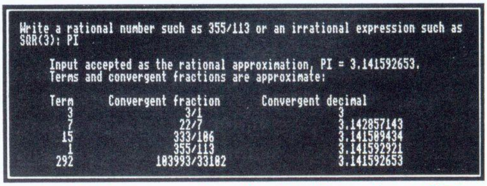
Any fraction, like 67/24, can be written as a decimal, in this case, 2.791666.... Of course, with a base other than 10, the expansion would be different and so there is nothing very special about 2.791666.... However, even the remotest aliens in the Galaxy would agree on the one unique way of writing 67/24 as:

$$2 + \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{4}}}}$$

Rational numbers may be written *exactly* - and in exactly *one way* - as continued fractions. We can also approximate irrational numbers like SQR3 or PI as closely as we like. Each of them has its own unique continued fraction that goes on for ever (that's the way to put it if you're a realist - an intuitionist will want to say that you can construct it as far as you choose).

It is the patterns of some of these continued fractions which are fascinating, and also the way in which by chopping them short, we can find approximations to well known numbers. As an introduction to the program let's find the first few approximations to the value of PI.

The program is short and in Basic, and so it is easy to type in, save, and run. When asked for a number, key in PI. The program will show that it has accepted it as the rational approximation 3.141592653. It then prints a table with three columns (see table 1).



```
Write a rational number such as 355/113 or an irrational expression such as
SQR(3): PI

    Input accepted as the rational approximation, PI = 3.141592653.
    Terms and convergent fractions are approximate:

    Term      Convergent fraction        Convergent decimal
       3               3/1                    3
       7              22/7                    3.142857143
      15             333/106                  3.141509434
       1             355/113                  3.141592921
     292          103993/33102                3.141592653
```

Table 1. Approximation to the value of PI

The left column says that PI can be written as:

$$3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \ldots}}}}$$

The middle column gives a list of approximations to PI. We see the first approximation of 3 and then the well-known one of 22/7. Next but one comes 355/113, the one discovered long ago by the Chinese. These 'convergents' in the second column are written again in the third column as decimal approximations. We can see how they approach the value of PI and what a good approximation 355/113 is.

Now try a fraction like 377/233. This time the program detects that, by writing the fraction as two integers separated by '/', we have offered it a precise rational number and it responds that it has accepted it as such. The reason why there are two different procedures in the program, one for rationals and another for irrationals, is to do with the precision of the BBC micro's arithmetic (see the Program Outline below). We see that the left-hand column just consists of a string of '1's: it was not any old fraction that we chose! Readers who know about the Fibonacci series can explore the ratios of higher consecutive Fibonacci numbers. The biggest ones that can be accepted precisely are 1134903170 and 1836311903. Do try 1836311903/1134903170, and you will be rewarded with all the lower Fibonacci numbers in the convergents.

Finally, let's try the square root of an integer, SQR7 for example. Fascinating patterns arise and the reader can explore further - indeed while playing with the ratios of the Fibonacci numbers, he may already have already keyed in (1+SQR5)/2. Another continued fraction with a pattern is that of 'e' - and also certain functions of it.

PROGRAM OUTLINE

There are two main procedures, PROCRational and PROCIrrational. The only reason that there are two alternative procedures is that precision is only possible for rational numbers with numerators and denominators up to the maximum accepted as integers by the computer, that is up to 2^31-1. Thus, for a fraction, PROCRational gives greater possible precision than if the fraction had been entered as a decimal, but with a decimal approximation to an irrational calculated from EVAL there is more precision with PROCIrrational which does not use integers.

The program treats any input as a rational fraction until it finds that there is not exactly one '/' character in the expression keyed in, or it then finds that what are on either side of the '/' are not integers. If for these reasons it cannot treat the input as an exact rational, the program leaves PROCRational and enters PROCIrrational. Left to itself PROCIrrational would run indefinitely, soon producing chaos when it 'used up' the information in the irrational number. Line 1240 prevents this by terminating the procedure before the terms on the left lose accuracy - the test has been set by trial and error, and the accuracy of the last term cannot be guaranteed.

PROCCalc uses recurrence relations (see Davenport p.85) to calculate the convergents for PROCRational and PROCIrrational. Error trapping is used to detect if EVAL cannot evaluate its argument, and the main program loop is then restarted.

```
   10 REM Program FracFun
   20 REM Version B1.0
   30 REM Author  Michael Taylor
   40 REM BEEBUG  December 1990
   50 REM Program subject to copyright
   60 :
  100 MODE 0:ON ERROR GOTO250
  110 DIM T(50),A(50),B(50),C(50)
  120 VDU19,0,7;0;19,1,0;0;
  130 REPEAT
  140 S=0:@%=10:Rational%=TRUE
  150 REPEAT
  160 VDU10,10,10,10,11,11
  170 INPUT" Write a rational number suc
h as 355/113 or an irrational expression
 such as      SQR(3): "X$
  180 UNTIL X$<>""
  190 Test=EVAL(X$)
  200 PROCRational
  210 IF NOT(Rational% OR error%) PROCIr
rational
  220 UNTIL FALSE
  230 END
  240 :
  250 IF ERR=26 AND ERL<200 GOTO130 ELSE
 IF ERR=26 PROCIrrational:GOTO130
  260 MODE 3:REPORT:PRINT" at line ";ERL
  270 END
  280 :
 1000 DEF PROCRational:error%=FALSE
 1010 IF INT(EVAL(X$))=EVAL(X$) PRINT'"
There is no point in entering an integer
.":error%=TRUE:ENDPROC
 1020  S=0:exit%=FALSE
 1030 LOCAL N,M,X,Y,X%,Y%,XN%,YN%
 1040 N=INSTR(X$,"/"):IF N=0 Rational%=F
ALSE:ENDPROC
 1050 Num$=LEFT$(X$,N-1):Den$=RIGHT$(X$,
LEN(X$)-N)
 1060 M=INSTR(Den$,"/"):IF M<>0 Rational
```

```
%=FALSE:ENDPROC
 1070 Y=EVAL(Num$):X=EVAL(Den$):IF (INT(
Y)<>Y OR INT(X)<>X) Rational%=FALSE:ENDP
ROC
 1080 PRINT'"        Input accepted as the
 rational number, ";Num$;"/"Den$"."'"
  Terms and convergent fractions are ex
act:":PROCPrint
 1090 Y%=Y:X%=X
 1100 REPEAT
 1110 S=S+1
 1120 IF X%<>0 T(S)=Y%DIVX%:PROCCalc:PRI
NTT(S)TAB(20)A(S);"/";B(S);TAB(55);C(S)
ELSE exit%=TRUE:GOTO1150
 1130 YN%=X%:IFX%<>0 XN%=Y%MODX%
 1140 Y%=YN%:X%=XN%
 1150 UNTIL exit%
 1160 ENDPROC
 1170 :
 1180 DEF PROCIrrational
 1190 exit%=FALSE
 1200 X=EVAL(X$)
 1210 PRINT'"        Input accepted as the
 rational approximation, ";X$;" = ";X;".
"'"      Terms and convergent fractions
are approximate:":PROCPrint
 1220 REPEAT:S=S+1:Y=Y*X
 1230 T(S)=INT(X):PROCCalc
 1240 IF ABS(C(S))<>0 THEN IF S>2 AND (A
BS(C(S)-C(S-1))/ABS(C(S)))<5E-9 exit%=TR
UE:GOTO1280
 1250 PRINT T(S)TAB(20)A(S);"/";B(S);TAB
(55);C(S)
 1260 X=1/(X-INT(X))
 1270 IF INT(X)=X exit%=TRUE
 1280 UNTIL exit%
 1290 ENDPROC
 1300 :
 1310 DEF PROCCalc
 1320 IF S=1 A(1)=T(1):B(1)=1
 1330 IF S=2 A(2)=T(1)*T(2)+1:B(2)=T(2)
 1340 IF S>2 A(S)=T(S)*A(S-1)+A(S-2)
 1350 IF S>2 B(S)=T(S)*B(S-1)+B(S-2)
 1360 C(S)=A(S)/B(S)
 1370 ENDPROC
 1380 :
 1390 DEF PROCPrint
 1400 PRINT'TAB(6)"Term"TAB(16)"Converge
nt fraction"TAB(42)"Convergent decimal"
 1410 ENDPROC
```

The Wapping Editor

Jim Hudson investigates Wapping Editor, Watford Electronics' desktop publisher for the BBC micro and Master series.

Product	**Wapping Editor**
Supplier	**Watford Electronics** **Jessa House,** **250 Lower High Street,** **Watford WD1 2AN.** **Tel. (0923) 37774**
Price	**£79.35 inc. VAT** **Art Disc £17.25 inc. VAT** **Font Disc £13.80 inc. VAT**

The increasing emphasis placed by software houses on their Archimedes ranges of products means that the Watford Electronics' *Wapping Editor* will probably be one of the last purpose-built desktop publishers produced for Acorn's 8-bit machines.

The package itself consists of a 64K ROM, a utility disc and a 100 page manual. A *Font Disc* and an *Art Disc* are available as optional extras. The Wapping Editor is compatible with machines from a Model B to a Master, though some minor features are lost with a basic 32K Model B. It is of advantage to have both shadow and sideways RAM available, since the Editor is able to access both, and this speeds up a variety of options. Wapping Editor is almost entirely "mouse controlled", so a rodent (Watford supply a suitable one) or compatible device is essential.

GETTING STARTED

The Editor is compatible with DFS, ADFS and Network filing systems. The first stage in using this package is to format a disc and copy three programs from the utility disc onto it. These ensure that pages can be previewed and printed, (a procedure that seems to contradict the advertisements that state "a 64K ROM containing ALL the software needed to get into print fast").

A preferable alternative to copying files is to use the "Support ROM" contained on the utility disc. This can be loaded into a sideways RAM slot or blown into an EPROM. This ROM image contains the three extra programs, and also a number of other features used during the drawing process (see later). I now load the Support ROM, initialise it (see BEEBUG Vol.6 No.1) and call the Editor as part of a !Boot File. Once called you are faced with the "Front Page". In common with the rest of the package this is a mode 0 screen.

USING THE EDITOR

The Front Page is the entry point to the various sections of the Editor, and this screen clearly demonstrates the modular nature of the package. There are four main areas that can be used:

Area 1. "Modules" - containing:

a) the Graphics Module, allowing drawings to be created and "blended with text" or in Wapping Editor language "typeset",

b) the Word Processor module, and

c) the Font Editor Module, for the alteration and production of fonts.

Area 2. "Pages" - containing some of the software to load, save and preview pages.

Area 3. "Print" - allowing pages to be printed using the default Epson printer driver, or allowing another to be loaded.

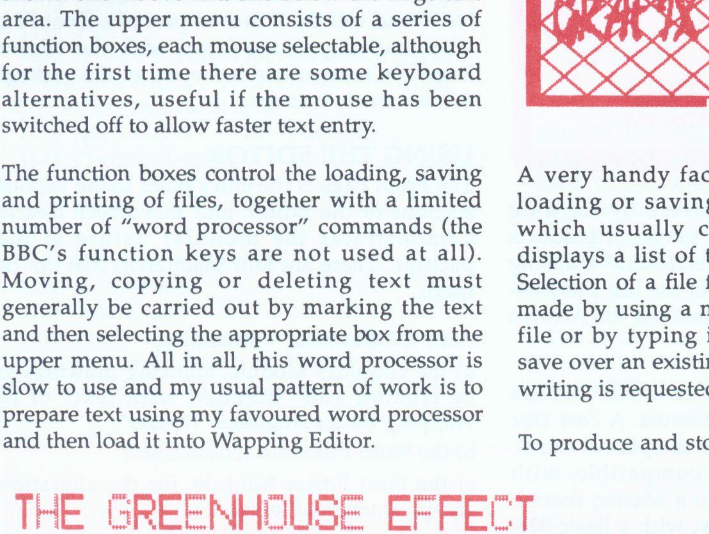
Area 4. "Utilities" - enables pages to be created, or allows conversion of non-mode 0 screens for use within the Editor.

When starting a page from 'scratch', my first selection from the Front Page is to "create a page". The Wapping Editor scores very highly in this regard; there are 8 default pages provided, and a facility to create any size of page. The default pages range from A5 to A3, both vertical and landscape formats being allowed. My usual choice of format is the "A4 Hi-res"; on printing, the pixels are squashed vertically and horizontally, ensuring that 1 pixel = 1 printed dot.The result is a high quality printout for a 9-pin dot-matrix printer.

After careful planning of the page, it is a matter of personal taste whether to start by entering text or diagrams; my own preference is to produce text. Wapping Editor's own word processor is entered by selecting the appropriate box from the Front Page.

There are two menus for the word processor, shown one above and one below the large text area. The upper menu consists of a series of function boxes, each mouse selectable, although for the first time there are some keyboard alternatives, useful if the mouse has been switched off to allow faster text entry.

The function boxes control the loading, saving and printing of files, together with a limited number of "word processor" commands (the BBC's function keys are not used at all). Moving, copying or deleting text must generally be carried out by marking the text and then selecting the appropriate box from the upper menu. All in all, this word processor is slow to use and my usual pattern of work is to prepare text using my favoured word processor and then load it into Wapping Editor.

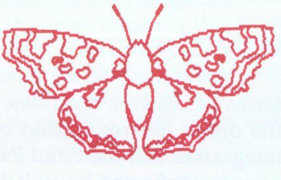

THE GREENHOUSE EFFECT
Cycle option was used to change the "texture of the font.
Text and graphics can be rotated to any angle.
Lines can be locked to enable parallel lines to be easily drawn.
Or drawn freehand.
Figure "Snatched" from a mode 4 package and converted to mode 0.

Control codes from external word processors are stripped on loading, and minor editing is all that is needed to ensure that line and paragraph breaks occur where required. Wapping Editor's word processor produces its document in simple ASCII code, all highlighting and formatting occurring when the page is "typeset."

A very handy facility becomes apparent on loading or saving files. The bottom menu, which usually contains the word count, displays a list of text files contained on disc. Selection of a file for loading or saving can be made by using a mouse to point at the correct file or by typing its name. If you attempt to save over an existing file confirmation for over-writing is requested.

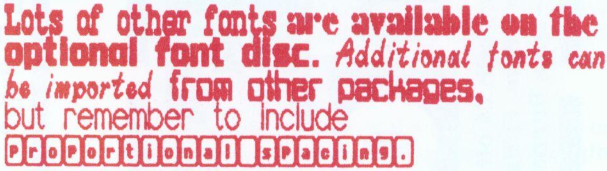
To produce and store graphics a page is loaded and the Graphics Module entered, once again through the Front Page. For such an important and powerful module the absolutely blank screen comes as something of a surprise. There are menus available, but these must first be called. They appear at the top and bottom of the screen, almost like pull down/pull up menus. Any items underneath the screen are stored in normal RAM while the menu is on screen (sideways RAM is needed if both menus are to appear on screen simultaneously). The top menu, entitled the "command menu" contains most of the drawing functions plus entry to the "I/O" and "typesetting" sections.

The I/O menu allows star commands to be issued, discs to be catalogued, and items such as cut-outs and screens to be loaded or saved. These can be home-made, taken from the "Wapping Editor Art Disc", or loaded from similar products, such as AMX Stop Press or 4Mation's "Snatch" (reviewed in Beebug Vol.7 No.6, essentially a program to "grab" screens from other products).



I have been very impressed with the drawing capability of such a reasonably priced product as Wapping Editor. Drawing functions are selected from the upper menu and depending on the particular function chosen the menu may change, offering more choices. For example, if the ellipse drawing function is called, new boxes dealing with the starting position and whether or not the ellipse should be circular are presented.

The options available within the graphics module are extensive, and it is necessary to follow the manual page by page in order to discover the full potential. All of the common drawing functions are present, although to have access to circular arcs and chords and to be able to rotate or distort cut-outs, both Acorn's GXR ROM and Wapping Editor's Support ROM must be present.

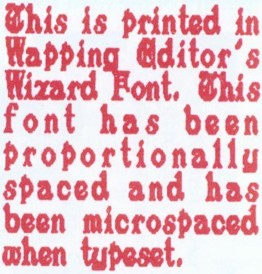


Lots of other fonts are available on the optional font disc. Additional fonts can be imported from other packages, but remember to include Proportional spacing.

No icons appear in the upper menu; all functions are listed using plain English. Personally I prefer this; it saves me having to read the icon designer's mind. Features which I find particularly helpful are the "Undo" and the "Protect" facilities. *Undo* will remove the last drawing action whilst *Protect* will prevent over-writing other work. I feel that the lack of a scale on any of the screens is a problem, so much so that I start my graphics work by first cutting and pasting a home-made scale onto my page.

The Graphics Module's bottom menu is called the "palette menu". As its name suggests it contains a library of patterns for filling areas, and shapes for changing brushes. It allows cycles of patterns to be designed, so that areas will be filled with different patterns automatically, rather than by having to select a pattern for each item. New patterns and brush shapes can be designed within the menu.

A third menu, the "Toolbox" can be called at any stage by pressing the central button on the mouse. This Toolbox offers an eraser, a pixel editor and a screen scrolling facility. The manual correctly argues that these facilities are frequently needed, so should be readily available. The pixel editor is particularly good; it provides two magnifications, and allows easy movement around a diagram, without having to re-position the editor. The speed of page scrolling depends on several factors. A Hi-Res page is stored on disc, so several seconds are needed to move a significant distance. However, a smaller page is stored in sideways RAM (if available) and scrolling is then almost instantaneous.



This is printed in Wapping Editor's Wizard Font. This font has been proportionally spaced and has been microspaced when typeset.

After preparing text and artwork, the two are blended together using the command menu's "typeset" option. This involves defining areas around the artwork into which text will be read. After calling the necessary text files and choosing appropriate fonts, the text fills the defined areas. By default this text is microspaced. There is no problem if the text file is too big, as the page can be panned and loading continued. A variety of fonts are available on disc, but the Editor can also use "Stop Press" fonts, so there is a very large stock of potential fonts available. For the best results, it is vital that any imported fonts are proportionally spaced; this ensures that a thin letter, such as an "i" takes up less space than a fat "w" for example. The font editor makes short work of such corrective action.

I have attempted to use a font editor/designer several times and usually given up. The one contained here is just about the best I have met. The font under study is displayed at all times, all functions are mouse selected, and it is quite quick to modify and produce new characters.

PRINTING

For me, this is the factor that dominates my opinion of such a product. I only have access to a 9-pin dot-matrix printer, but still want a good quality printout. A choice of draft, standard or high quality printouts is offered, but the overall quality of the hard copy is largely controlled by the type of page loaded. A standard A4 page gives the obvious stepping of angled lines, whereas the Hi-Res page gives a more than acceptable result. One problem with the Hi-Res printout is that the compression of pixels at the printing stage results in a "distortion" on the hard copy, circles have flat tops for example. The optional art disc has screens available to keep this problem to a minimum. For this reason alone I would suggest that this optional disc is purchased.

CONCLUSION

It is convenient to think of this package as a series of discrete modules, each isolated from the other. However, that belies the degree of integration that the Front Page allows. A loaded page is unaffected by switching between any of the available modules. It is a simple matter to leave a graphics screen, compose text in the word processor and then return to "typeset" the text. There is no need to have even saved the page. It is this degree of flexibility and speed, coupled with the quality of output that has allowed Wapping Editor to become my first choice desktop publisher. B

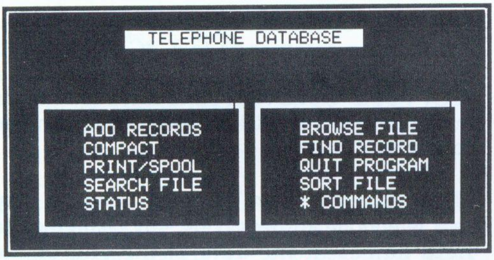
MikroTel (Part 1)

To complement the recent Phone Call Costing program Mike Bryant presents a versatile database for storing names, addresses, phone numbers, and more besides.

MikroTel is a versatile program for storing telephone numbers, addresses, and other such information. It is simple enough to be operated by 'non-computer' users, but powerful enough to process large quantities of data.

MikroTel is, by necessity, a long program, to be split over this and the next issue. In this issue you will find the core of the main program, together with a few of the basic facilities: you can read, write, and delete records, and with the short assembler routine, sort them into alphabetical order.

Once you have typed in the main listing (Listing 1), save the program immediately - DO NOT RENUMBER as it has been carefully planned to accommodate part two next month. You will also need to create the two blank files DATA and HASH, ready for writing your own records. The DATA file should be of a round 64K of blank space, and the HASH file 8K. This can be done by running the following program:



The main MicroTel menu

```
 10 MODE7:HIMEM=&5C00
 20 FOR I%=&5C00 TO &7BFD STEP 4
 30 !I%=0
 40 NEXT
 50 *SAVE HASH 5C00 7BFF
 60 X=OPENOUT("DATA")
 70 FOR I%=0 TO &FFFF
 80 BPUT#X,0
 90 NEXT
100 CLOSE#X
```

You will also need to type in Listing 2, saving this as Sort, and then running it to create the machine code sort routine, MCSORT used by MikroTel. You are then ready to start using MikroTel.

MIKROTEL IN ACTION

For the technically minded, MikroTel stores each record in two parts: the surname from each record goes into a hash table in memory from &5C00 to &7C00. The format of this is shown in table 1.

&5C00	no. of records in the file (LSB)
&5C01	no. of records in the file (MSB)
.....	
.....	
&5C10 to &5C1D	Surname of first record (in ASCII, followed by &0D)
&5C1E	pointer to location of rest of record data on disc (LSB)
&5C1F	pointer to location of rest of record data on disc (MSB)
&5C20 to &5C2F	next record etc.

Table 1. Format of hash table for surname

The rest of the record is variable length, and stored on disc in the familiar internal format used by the PRINT# command. Each record begins with a '<', and ends with a '>', so that the compacting procedure is simplified (see next issue for compaction routine).

The reason for splitting the data is that it makes sorting (by surname) very much faster, since the only things that are compared and swapped are the surnames and pointers themselves, and in memory - it is very much faster to read data from memory than from disc.

Another point of note is the technique used to delete records. When using variable length records, it is always a problem to delete them when required, and what I have done is as follows: first, find the location of the record data on disc, and put a CHR$(0) where the '<' used to be. That takes care of the data on disc! Next, take the LAST surname in the hash table in memory, and put that where the surname is that you are deleting. Reduce the value for the number of records that there are, and voila! a record is deleted. However, the records are not in order any more, so, after a quick sort (1 sec or so) the records are back in their proper order, and the hash table can be saved to disc (2 secs.). This is a faster, if more indirect, method of deletion using a hash table.

USING MIKROTEL

MikroTel operates from a central menu from which each of the functions is selected by moving the cursor (with the cursor keys), and pressing Return to select any option. In this issue, the Add, Quit, Oscli options, and part of the Browse option are implemented, and are summarised below. The remaining parts of Browse, as well as Find, Print/Spool, Search, Sort and Status will be included in the next issue.

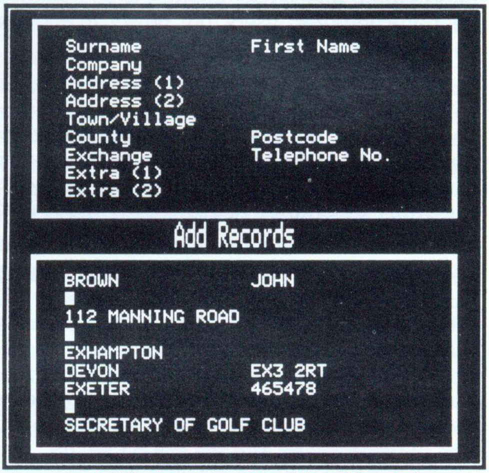
The ADD option allows you to write your name and address data onto disc. You can enter the individual fields in any order, moving about with the cursor keys. You can also change the data you've just entered, if it's wrong, by simply going back and overwriting it. When you've finished writing the record, just keep pressing the cursor-down key until you 'fall off' the bottom of the screen. The screen will clear, the record saved to disc already, and you can enter the next record. When you've finished, press Escape.

You can use any characters you like in the fields other than '<' and '>', apart from in the telephone number field, where only digits and '-' are allowed.

The BROWSE option is the main editing option, allowing you to perform a number of different operations. In part one, however, only a few options are implemented, and the rest will follow in the next issue.

With the present listing, you can flick through the records on disc by pressing the cursor-up key to see the next record, or cursor-down to see the previous record. You can also delete the record that is currently displayed in the box at the top by pressing 'D'. Pressing any of the other keys, whose functions are described in the lower box on the screen, will result in errors, until the next part of the listing is added.

The QUIT option should always be used when finishing with MikroTel, instead of just pressing Break, etc., since it ensures that all data is saved properly, by closing all open files, and generally 'tidying up'. This is vital if you have been entering data, but not so important if you have only been reading the data. Failure to use the Quit option may result in a corrupted data file.



```
Surname              First Name
Company
Address (1)
Address (2)
Town/Village
County               Postcode
Exchange             Telephone No.
Extra (1)
Extra (2)

              Add Records

BROWN                JOHN
■
112 MANNING ROAD
■
EXHAMPTON
DEVON                EX3 2RT
EXETER               465478
■
SECRETARY OF GOLF CLUB
```

Adding a new record in MikroTel

The OSCLI option allows you to perform most normal star commands (e.g. *CAT, *DRIVE, etc), but will prevent you from performing a *COPY or a *COMPACT since these commands may destroy the MikroTel program in the computer's memory.

Please note that MikroTel can be used not only as a telephone address book - by changing the field names in the data at the end of the program you can call the fields whatever you like, to make the database more suited to other needs. For 'personalisation' purposes, the format of the field data is as follows: name of field, max. allowed length of field (including Return), co-ordinates of where the field should be printed in the display box at the top of the screen, the number of the field that is accessed when you press the cursor-up key (i.e. when adding or modifying - the surname field is counted as number 0, first name as 1, etc.), ditto for cursor down. The only reason for limiting the maximum length of all strings, other than the surname, is to keep the screen display looking correct.

Listing 1

```
   10 REM Program MikroTell
   20 REM Version B1.0
   30 REM Author  Mike Bryant
   40 REM BEEBUG  December 1990
   50 REM Program subject to copyright
   60 :
  100 MODE7
  110 HIMEM=&5A00
  120 PROCinit
  130 :
  140 REM Main menu loop...
  150 REPEAT
  160 ON ERROR GOTO5870
  170 PROChead("TELEPHONE DATABASE")
  180 PROCno_cur
  190 Z%=0:*FX202 32
  200 RESTORE5690
  210 PROCbox(20,10,39,4)
  220 PROCbox(0,10,19,4)
  230 FOR I%=5TO9
  240 READ $T%,$U%
  250 PRINTTAB(3,I%)SPC2$T%TAB(17,I%)CHR
$156TAB(23,I%)SPC2$U%TAB(37,I%)CHR$156
  260 NEXT
  270 REPEAT
  280 PRINTTAB(H%,V%)CHR$157CHR$132;
  290 A%=GET
  300 PRINTTAB(H%,V%)CHR$156" ";
  310 IF A%=136 H%=3
  320 IF A%=137 H%=23
  330 IF A%=139 V%=V%-1:IF V%<5V%=5
  340 IF A%=138 V%=V%+1:IF V%>9V%=9
  350 UNTILA%=13
  360 B%=(V%-5)*2+ABS(H%=23)
  370 IF NR%=0THEN390
  380 IF B%=1 PROCbrowse ELSEIF B%=2PROC
compact ELSEIF B%=3PROCfind(0) ELSEIF B%
=4PROCprint ELSEIF B%=6PROCsearch(0) ELS
EIF B%=7PROCsort
  390 IF B%=0PROCadd ELSEIF B%=5PROCquit
 ELSEIF B%=8PROCstatus ELSEIF B%=9PROCos
cli
  400 UNTIL FALSE
  410 END
  420 :
 1000 DEFPROCadd
 1010 LOCAL I%,M%:PROCtopf
 1020 REPEAT
 1030 IF NR%>510 PRINT"No room for more
records":IF INKEY500:ENDPROC
 1040 PROCbotf("Add Records")
 1050 FOR I%=0TO11
 1060 PRINTTAB(pos%(I%,0)-1,pos%(I%,1))C
HR$137field$(I%);
 1070 PRINTTAB(pos%(I%,0),pos%(I%,1)+13)
CHR$255;
 1080 orig$(I%)=""
 1090 NEXT
 1100 PROCinputrecord
 1110 UNTIL M%
 1120 ENDPROC
 1130 :
 1140 DEFPROCbrowse
 1150 LOCAL R%,G%,J%,Z%
 1160 PROCtopf:PROCbrowsef
 1170 REPEAT
 1180 PROCshow(R%):G%=GET
 1190 IF G%=70 PROCfind(-1):R%=Z%:PROCbr
owsef
 1200 IF G%=68ANDR%>0 PROCdelete(R%):PRO
Csavehash
 1210 IF G%=77ANDR%>0 PROCdelete(R%):PRO
Cbotf("Modify"):PROCinputrecord:PROCbrow
sef
 1220 IF G%=83 PROCsearch(-1):R%=Z%:PROC
browsef
 1230 IF G%=136ORG%=138 R%=R%-1:IFR%<1TH
ENR%=1
 1240 IF G%=137ORG%=139 R%=R%+1
 1250 IF R%>NR% R%=NR%
 1260 UNTIL G%=13
 1270 ENDPROC
 1280 :
 1500 DEFPROCoscli
 1510 PROChead("O.S. COMMANDS")
 1520 INPUT"*"$T%:IF $T%>""ANDLEFT$($T%,
2)<>"CO" THENOSCLI($T%):GOTO1520 ELSE EN
DPROC
 1530 :
 1840 DEFPROCquit
 1850 PROChead("QUIT PROGRAM")
 1860 PRINT''"Sure (Y/N)? ";
 1870 IF FNyn="N" ENDPROC
 1880 *CLOSE
 1890 PROCsavehash
 1900 CLS
 1910 PRINT"Thank you for using MikroTel
"
 1920 END
 1930 :
 2750 REM Miscellaneous sub-routines...
 2760 :
 2770 DEFPROCno_cur:?&FE00=10:?&FE01=32:
ENDPROC
 2780 :
 2790 DEFPROCbig_cur:?&FE00=10:?&FE01=96
:ENDPROC
 2800 :
```

```
 2810 DEFPROCtopf
 2820 PRINTTAB(0,0)SPC220SPC220
 2830 PROCbox(0,10,39,0)
 2840 ENDPROC
 2850 :
 2860 DEFPROCbotf($U%)
 2870 LOCAL I%
 2880 PRINTTAB(0,11)SPC220SPC220SPC79
 2890 FOR I%=11TO12
 2900 PRINTTAB(INT((37-LEN($U%))/2),I%)C
HR$141$U%;
 2910 NEXT
 2920 PROCbox(0,23,39,13)
 2930 ENDPROC
 2940 :
 2950 DEFPROCbrowsef
 2960 LOCAL I%
 2970 PROCbotf("Browse")
 2980 PRINTTAB(3,14)"Press ] for next re
cord"
 2990 PRINTTAB(9,15)"[ for previous reco
rd"
 3000 PRINTTAB(9,16)"D to delete this re
cord"
 3010 PRINTTAB(9,17)"F to find a specifi
c record"
 3020 PRINTTAB(9,18)"M to modify this re
cord"
 3030 PRINTTAB(9,19)"S to search file"
 3040 PRINTTAB(9,22)"Return to return to
 main menu"
 3050 ENDPROC
 3060 :
 3170 DEFPROCbox(L%,B%,R%,W%)
 3180 LOCAL I%
 3190 PRINTTAB(L%,W%)CHR$146CHR$55STRING
$(R%-L%-2,CHR$96)CHR$107
 3200 FOR I%=1TOB%-W%-1
 3210 VDU31,L%,W%+I%,146,53,135,31,R%-2,
W%+I%,137,146,106
 3220 NEXT
 3230 PRINTTAB(L%,B%)CHR$146CHR$117STRIN
G$(R%-L%-2,CHR$112)CHR$122;
 3240 ENDPROC
 3250 :
 3260 DEFPROCdelete(R%)
 3270 REM Put flag byte in data file
 3280 PTR#X=!(FNh(R%)+14)AND&FFFF
 3290 *FX200 1
 3300 BPUT#X,0
 3310 REM Put last in list into hole
 3320 $(FNh(R%))=$(FNh(NR%))
 3330 REM ..and put addresses there too!
 3340 ?(FNh(R%)+14)=?(FNh(NR%)+14)
 3350 ?(FNh(R%)+15)=?(FNh(NR%)+15)
```

```
 3360 REM Shuffle along the address of t
he next available space on disk
 3370 ?(FNh(NR%)+14)=?(FNh(NR%+1)+14)
 3380 ?(FNh(NR%)+15)=?(FNh(NR%+1)+15)
 3390 $FNh(NR%)="Pointer":$FNh(NR%+1)=ST
RING$(15," ")
 3400 REM Decrement the # of records
 3410 NR%=NR%-1
 3420 ?&5C00=NR% MOD256
 3430 ?&5C01=NR% DIV256
 3440 *FX200 0
 3450 PROCsorthash
 3460 Z%=0
 3470 ENDPROC
 3480 :
 3560 DEFFNfilesize:=INT((!(FNh(NR%+1)+1
4)AND&FFFF)/1024)
 3570 :
 3580 DEFFNget($U%):LOCALA$:REPEAT:A$=GE
T$:UNTILINSTR($U%,A$)<>0:=A$
 3590 :
 3600 DEFFNyn:LOCALA$:REPEAT:A$=GET$:UNT
ILINSTR("YN",A$)<>0:PRINTA$;:=A$
 3610 :
 3620 DEFPROCanytab:PRINTTAB(3,20)"Press
 any key for next record,"'TAB(3,21)"or
TAB to stop...":ENDPROC
 3630 :
 3640 DEFPROCnomatch:PRINTTAB(7,20)"No m
atch could be found.":ENDPROC
 3650 :
 3660 DEFFNh(R%):=&5C00+R%*16
 3670 :
 3680 DEFPROChead(A$):CLS:VDU31,INT((35-
LEN(A$))/2),0,131,157,132:PRINTA$"  "CHR
$156:ENDPROC
 3690 :
 3700 DEFFNinput(F%,original$)
 3710 LOCAL I%,X%,Y%
 3720 $T%=""
 3730 IF original$="" original$=CHR$255
 3740 X%=pos%(F%,0):Y%=pos%(F%,1)+13
 3750 PRINTTAB(X%-1,Y%-13)CHR$136;
 3760 I%=I%+1
 3770 IF I%=2 PRINTTAB(X%,Y%)SPC(len%(F%
));
 3780 PRINTTAB(X%,Y%)$T%;
 3790 IF I%=1 PRINTTAB(X%,Y%)original$;T
AB(X%,Y%);
 3800 PROCbig_cur:last%=GET
 3810 PROCno_cur
 3820 IF last%=13OR(last%>135ANDlast%<14
0) GOTO3900
 3830 IF last%=127ANDI%>1 $T%=LEFT$($T%,
LEN($T%)-1):I%=I%-2:VDU127:GOTO3760
```

```
 3840 IF last%=127VDU255:I%=I%-1:GOTO376
0
 3850 IF F%=9ANDNOT((last%>47ANDlast%<58
)ORlast%=45ORlast%=32) I%=I%-1:GOTO3760:
REM This deals with the Telephone No. ca
se, where digits, '-', or ' ' only are a
llowed.
 3860 IF I%=len%(F%) I%=I%-1:GOTO3760
 3870 IF last%=60ORlast%=62 GOTO3760
 3880 $T%=$T%+CHR$(last%)
 3890 GOTO3760
 3900 PRINTTAB(X%-1,Y%-13)CHR$137;
 3910 IF original$=CHR$255original$=""
 3920 IF $T%=" "$T%=""
 3930 IF I%=1 =original$ ELSE=$T%
 3940 :
 3960 DEFPROCinit
 3970 LOCAL I%
 3980 *FX200 1
 3990 ON ERROR GOTO5870
 4000 CLOSE#0
 4010 *FX4 1
 4020 PROCno_cur
 4030 DIM field$(11),len%(11),pos%(11,1)
,up%(11),down%(11),new$(11),orig$(11),fo
%(11)
 4040 RESTORE5730
 4050 FOR I%=0TO11
 4060 READ field$(I%),len%(I%),pos%(I%,0
),pos%(I%,1),up%(I%),down%(I%)
 4070 NEXT
 4080 *LOAD HASH
 4090 *LOAD MCSORT
 4100 REM *LOAD MCCMPCT
 4110 *FX200 0
 4120 NR%=(!&5C00 AND&FFFF)
 4130 X=OPENUP("DATA")
 4140 D%=FALSE:H%=3:V%=5
 4150 T%=&5A00:U%=&5B00
 4160 $T%=STRING$(255," ")
 4170 $U%=STRING$(255," ")
 4180 B$=STRING$(5," ")
 4190 C$=STRING$(11," ")
 4200 ENDPROC
 4210 :
 4220 DEFPROCinputrecord
 4230 LOCAL I%,last%
 4240 IF FNdiskfree<300ORFNfilesize>62 V
DU7:M%=TRUE:PRINTTAB(0,11)"WARNING! Insu
fficient disc space left"'"to add any mo
re records.":OSCLI("FX15 1"):IF INKEY500
:ENDPROC
 4250 REPEAT
 4260 new$(I%)=FNinput(I%,orig$(I%))
 4270 orig$(I%)=new$(I%)
```

```
 4280 IF I%=0ANDnew$(0)>""ANDD% THEN IF
FNfindsurname(new$(0))>0 THENPROCexists
 4290 IF last%=13ORlast%=137I%=I%+1
 4300 IF last%=136I%=I%-1:IFI%<0I%=0
 4310 IF last%=139I%=up%(I%)
 4320 IF last%=138I%=down%(I%)
 4330 UNTILI%=12
 4340 *FX200 1
 4350 NR%=NR%+1
 4360 ?&5C00=NR%MOD256
 4370 ?&5C01=NR%DIV256
 4380 $FNh(NR%)=new$(0)
 4390 PTR#X=!(FNh(NR%)+14)AND&FFFF
 4400 BPUT#X,60
 4410 FOR I%=1TO11
 4420 PRINT#X,orig$(I%)
 4430 NEXT
 4440 BPUT#X,62
 4450 ?(FNh(NR%+1)+14)=PTR#X MOD256
 4460 ?(FNh(NR%+1)+15)=PTR#X DIV256
 4470 *FX200 0
 4480 PROCsorthash:PROCsavehash
 4490 ENDPROC
 4500 :
 5040 DEFFNdiskfree:F%=TRUE:=EXT#X-!(FNh
(NR%+1)+14)AND&FFFF
 5050 :
 5060 DEFPROCsavehash:OSCLI("SAVE HASH 5
C00 7BFF"):ENDPROC
 5070 :
 5440 DEFPROCshow(R%)
 5450 IF R%=Z% ENDPROC
 5460 LOCAL I%
 5470 PROCload(R%):PROCtopf
 5480 FOR I%=0TO11
 5490 PRINTTAB(pos%(I%,0)-1,pos%(I%,1))C
HR$137orig$(I%);
 5500 NEXT:Z%=R%
 5510 ENDPROC
 5520 :
 5530 DEFPROCload(R%)
 5540 LOCAL I%
 5550 orig$(0)=LEFT$($(FNh(R%)),14)
 5560 PTR#X=1+!(FNh(R%)+14)AND&FFFF
 5570 FORI%=1TO11
 5580 INPUT#X,orig$(I%)
 5590 NEXT
 5600 ENDPROC
 5610 :
 5620 DEFPROCsorthash
 5630 *FX200 1
 5640 CALL &C00
 5650 *FX200 0
 5660 ENDPROC
 5670 :
```

```
 5680 REM Main menu data...
 5690 DATA ADD RECORDS,BROWSE FILE,COMPA
CT,FIND RECORD,PRINT/SPOOL,QUIT PROGRAM,
SEARCH FILE,SORT FILE,STATUS,* COMMANDS
 5700 :
 5710 REM Field data...
 5720 REM Field,length,X,Y,up,down
 5730 DATASurname,14,4,1,0,2
 5740 DATAFirst Name,17,21,1,0,7
 5750 DATACompany,34,4,2,0,3
 5760 DATAAddress (1),34,4,3,2,4
 5770 DATAAddress (2),34,4,4,3,5
 5780 DATATown/Village,34,4,5,4,6
 5790 DATACounty,14,4,6,5,8
 5800 DATAPostcode,9,21,6,1,9
 5810 DATAExchange,14,4,7,6,10
 5820 DATATelephone No.,13,21,7,7,10
 5830 DATAExtra (1),34,4,8,8,11
 5840 DATAExtra (2),34,4,9,10,12
 5850 :
 5860 REM Error routine...
 5870 VDU3
 5880 IF ERR=17ANDNOTINKEY-1THENGOTO150
 5890 PROCsavehash
 5900 VDU14,7
 5910 PRINT''CHR$129"ERROR:";
 5920 REPORT
 5930 IF GET
 5940 IF INKEY-1PRINT" at line ";ERL:OSC
LI("FX4 0"):END ELSEGOTO150
```

Listing 2

```
   10 REM Hash table sort
   20 REM This routine, for MikroTel, wi
ll sort into alphabetical order the surn
ame strings (and corresponding addresses
 of data on disc).
   30 :
  100 nrL=&70:nrH=&71:rnL=&72
  110 rnH=&73:tableL=&74:tableH=&75
  120 table2L=&76:table2H=&77
  130 counterL=&78:counterH=&79
  140 temp=&7A:flag=&7B
  150 :
  160 FORpass%=0TO2STEP2:P%=&C00
  170 [OPTpass%
  180 .one_sort_pass
  190 LDA #1   :STA flag
  200 LDA &5C01:STA nrH
  210 LDA &5C00:CLC:ADC #1:STA nrL
  220 BCC next_full_loop
  230 INC nrH
  240 .next_full_loop
  250 LDA flag:BEQ end
  260 LDA #0   :STA rnH:STA rnL:STA flag
  270 LDA #&5C :STA table2H
  280 LDA #&10 :STA table2L
  290 LDA nrL:SEC:SBC #1
  300 STA nrL:BCS continue0
  310 DEC nrH:LDA nrL
  320 .continue0
  330 BNE next_surname
  340 LDA nrH
  350 BNE next_surname
  360 .end
  370 RTS
  380 :
  390 .next_surname
  400 INC rnL:LDA rnL:BNE continue1
  410 INC rnH:LDA rnL
  420 .continue1
  430 CMP nrL:BNE not_yet_finished
  440 LDA rnH:CMP nrH:BNE not_yet_finish
ed
  450 JMP next_full_loop
  460 :
  470 .not_yet_finished
  480 LDA table2H:STA tableH
  490 LDA table2L:STA tableL
  500 CLC:ADC #&10:STA table2L
  510 BCC continue2
  520 INC table2H
  530 .continue2
  540 LDY #&FF
  550 .compare
  560 INY:CPY #13:BEQ next_surname
  570 LDA (tableL),Y:CMP (table2L),Y
  580 BEQ compare
  590 BMI next_surname
  600 .swap
  610 STA flag:LDY #&10
  620 .loop
  630 DEY:BMI next_surname
  640 LDA (tableL),Y:STA temp
  650 LDA (table2L),Y:STA (tableL),Y
  660 LDA temp:STA (table2L),Y
  670 JMP loop
  680 ]NEXT
  690 *SAVE MCSORT C00 CFF
  700 END
```

Monte Carlo Methods

by Bernard Hill

We are going to take a look this month at a new subject which is common in larger computer systems, but little mentioned on the smaller species: simulation - the notable exception to this is Jan Stuurman's articles in BEEBUG June 1986 Vol.5 Nos.2 & 3. The principle involved is that we attempt to 'model' or copy a real world situation on the computer, and so find the answer to some problem, or an optimum strategy for operating some process, by running a number of simulations, usually making extensive use of the random number generator in the computer.

First, however, a quick word about random numbers. The seemingly simple function RND in BBC Basic is actually the fruit of a great deal of work, and in reality, of course, it does not produce true random numbers, but so-called *pseudo-random* numbers. That is, the sequence is apparently random (satisfying all tests for randomness) but quite predictable if you know the algorithm by which successive numbers are obtained. This has an important consequence for us in that we can happily repeat a previously obtained sequence of random numbers: quite vital for debugging purposes.

The Beeb's random-number generator has been subjected to a number of statistical tests, including a test for SEQUENCE REPETITION and FLATNESS. It is important in a simulation that the sequence of numbers generated should not repeat systematically: millions of numbers may be generated in our largest simulations and any duplication would render the results meaningless. Furthermore, the frequency of occurrence of any section of numbers should be statistically the same no matter whether the numbers are large or small or somewhere in between.

Let's remind ourselves of the different forms of the RND function available in BBC Basic:

x=RND	Returns a random 4-byte integer (-2147483648 to +2147483647)
x=RND(1)	Returns a random decimal in the range 0 to 0.999999...
x=RND(0)	Returns the last value of RND(1)
x=RND(N)	Returns a random integer in the range 1 to N
x=RND(-N)	Returns -N and selects a new sequence based on N for the next RND call.

The latter format is very useful in our debugging because if we place a call of (say) x=RND(-2) at the beginning of the program, then we will always generate the same sequence every time we run the program. Thus any program errors will be repeatable, which is essential for debugging purposes. Values other than -2 can of course be tried and when we are happy that the program is performing correctly then the call can be changed to (say) x=RND(-TIME) to guarantee that the sequence is not repeated on each run.

EVALUATION OF PI

As an introduction to this idea of simulation we are going to evaluate the constant PI so that we get an idea of the accuracy available in simulations.

Selecting two random co-ordinates (x,y) between 0 and 1 gives us a random point in the square box shown in figure 1. The probability that the point lies inside the quadrant shown will be equal to the area of the quadrant, or PI/4, i.e. about 0.7854. We can find out if the point is inside the quadrant by evaluating x*x+y*y, and if this is less than 1 then it will be (see figure 1). Our estimate of PI is thus four times the proportion of points counted inside the quadrant.

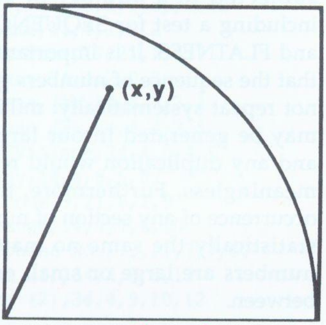


Figure 1

The following program illustrates this:

```
10 in=0:tot=0:PRINT "PI estimate:"
20 REPEAT
30 x=RND(1):y=RND(1)
40 IF x*x+y*y<1 THEN in=in+1
50 tot=tot+1
60 PRINT tot;" tries ";4*in/tot:VDU11
70 VDU 11
80 UNTIL INKEY(-99)
90 PRINT
```

Simply press the space bar at any time to terminate the process.

The volume of a sphere could be used in the estimate by adding a third random number z to line 30 and summing its square in line 40:

```
40 IF x*x+y*y+z*z<1 THEN in=in+1
```

You should get PI/6 as the volume of this octant (four dimensional spheres are similarly obtained by adding a fourth random number: you should get a value close to PI^2/32).

It may come as a surprise to those of you doing advanced mathematics to know that difficult integrals (areas under curves) are often calculated by precisely this type of method (particularly two- and three-dimensional integrals).

SIMULATING A GAME OF TABLE-TENNIS

Suppose you are playing someone slightly better than you at table-tennis. What sort of chance do you stand of beating him or her? This is the subject of our next simulation.

We will refer to the players as A and B. Let us assume that A has a probability of (say) 'p' of winning any given point. According to the rules of the game, one player has won when (s)he either reaches 21 points when the other has 19 or less, or is two points ahead should the weaker player score 20 or more (the 'deuce' concept). We can restate this as "the game is completed when either player has over 20 points and the difference in their scores is more than 1".

Let the Basic variables *scrA* and *scrB* be the two player's scores, so that the following function simulates one game and has the value TRUE if A wins. As a parameter, it takes the value *p* which is the probability of A winning any single point:

```
1000 DEF FNgame(p)
1010 LOCAL scrA,scrB:REM also sets to 0
1020 REPEAT
1030 IF RND(1)<p THEN scrA=scrA+1
          ELSE scrB=scrB+1
1040 UNTIL (scrA>20 OR scrB>20) AND
          ABS(scrA-scrB)>1
1050 =scrA>scrB
```

Now one game called with FNgame(0.4) tells us who won that one particular encounter, but if we call it a large number of times it will give us an estimate of the probability of A winning:

```
2000 DEF FNprobAwins(p)
2010 LOCAL a,n:REM a= # wins by player A
```

```
2020 FOR n=1 TO 1000
2030 IF FNgame(p) THEN a=a+1
2040 NEXT
2050 =a/1000
```

Again, while we're calculating the probability of winning with 0.4, we might as well calculate the probability of winning for a range of probabilities from 0 to 1 and plot a graph - and here is the program to do just this, but you need to add the functions above as well:

```
 10 MODE4:dummy=RND(-TIME)
 20 pinc=0.02
 30 MOVE 0,500:DRAW 1200,500
 40 FOR x=0 TO 1200 STEP 120
 50 MOVE x,490:DRAW x,510
 60 NEXT
 70 MOVE 0,0
 80 FOR p=0 TO 1 STEP pinc
 90 prob=FNprobAwins(p)
100 DRAW p*1200,prob*1000
110 NEXT
120 END
```

Inevitably this program runs a long time, involving over 2 million calls to RND, so be prepared to wait for an hour or two for the full graph, unless you cut down the game count (1000) or narrow the range of interest of p to, say, 0.3 to 0.7 in line 80.

EXTENSIONS

Writing this program in this modular (bottom-up) fashion means that we now have a tool for simulating any similar game: how about lawn tennis with its complicated rules about games, sets and match, advantage and deuce?

I leave you to evaluate the outcome, but it's fairly straightforward if you use the same idea as before, for instance:

```
3000 DEF FNset(p)
3010 LOCAL scrA,scrB
3040 REPEAT
3050 IF FNgame(p) THEN scrA=scrA+1
       ELSE scrB=scrB+1
3060 UNTIL (scrA>5 OR scrB>5) AND
       ABS(scrA-scrB)>1
3070 =scrA>scrB
```

DEFICIENCIES

This particular model is very weak in a number of areas: most glaring is that it ignores the value to a player of having the serve. There are a couple of ways around this:

1. We could use two probability variables to represent a player's skill: the probability of winning a point when he is serving and of winning a point when he is not. It would be interesting to simulate the play of a good-service player and one who is better in rallies.

2. We could move down to the ball-by-ball level giving different probabilities of returning the ball over the net on first and subsequent returns.

Either of these strategies is not too difficult to implement, and I find it very interesting that the graph produced is very steep near the centre of the screen: it tells us that both tennis and table-tennis are very reliable games in that (within the limits of these models) the slightly better player is very much more likely to win. Anyone care to do an analysis of football?

AND NOW TO THE CASINO

Monte Carlo methods are merely methods depending on the simulation of a model needing random calls. But let's get back to the gambling origins of the method, and to the roulette table.

There is an old 'infallible' method of winning at roulette. Simply bet consistently on red (or any event which returns at least even money, i.e. £2 for £1 staked). If you win, pocket your winnings and bet next time at £1. When you lose, bet next time at double the previous stakes. If you win, you have £4 to cover your £2+£1 stakes, if not then bet again, staking twice again (winning £8 to cover £4+£2+£1). Infallible! And the surprise is that it should work AT ANY ODDS - even 100-1 against!

Unfortunately,every Casino has its house betting limit, and doubling every time after a run of losses can soon bring you to that limit: at which point, of course you lose a great deal. But how risky is it?

Let's simulate a house limit of £12,000 per bet, and a minimum of £10. The probability of winning on the red on a wheel with 18 reds and 19 non-reds (there's at least one no-win zero on the wheel) is 18/37. Let's define a function FNprofit in which we bet until we win or hit the house limit:

```
1000 DEF FNprofit(stake,limit)
1010 LOCAL loss
1010 REPEAT
1020 loss=loss+stake
1030 stake=stake*2
1040 UNTIL stake>limit OR RND(1)<18/37
1050 IF RND(0)<18/37 THEN =1 ELSE =-loss
```

(note the use of RND(0) to bring back the last RND(1) which was calculated).

Now to embed this in an evening's entertainment, say 50 betting sequences (statistically, each on average will last 2 bets):

```
10 profit=0
20 INPUT "Stake : "stake
30 INPUT "Limit : "limit
40 FOR n=1 TO 50
50 profit=profit+FNprofit(stake,limit)
60 NEXT
70 PRINT "Net profit ";profit
80 END
```

Be sure to run the model for several evenings at the fireside though, before you risk your cash at the Casino!

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

POSTBAG (Vol.9 No.6)

In editing the formula for retrieving an element from a triangular array store in a single one dimensional array, a small but significant error crept in. The formula should read:

```
J+I*(I-1)/2
```

In the case of Mr. Holt's original problem, all the diagonal elements of the array are zero and can therefore be omitted, and thus the formula becomes:

```
J+(I-1)*(I-2)/2
```

where J runs from 2 to N, and I from 1 to N-1, and N is the order (size) of the matrix.

PHONE CALL COSTING (Vol.9 No.6)

In publishing the details of the changes to be made to the original Phone Call Costing program to extend it to cover international and some other call types we are sorry to say that some minor bugs crept in during editing.

First of all, in Vol.9 No.6, the third of the additional lines should read:

```
IF dist$="Error" GOTO290
```

and not finish with GOTO250 as printed. Remember that the GOTO line number is relevant to the original listing and may need adjusting if you renumber the program in order to incorporate the additional lines. To incorporate this modification in the programs on the magazine disc, line 210 in the program Phone3-M should be changed to read:

```
210 IF dist$="Error" GOTO320
```

and in Phone3-B the equivalent line to change is line 180 which should become:

```
180 IF dist$="Error" GOTO280
```

In addition, in Phone3-B only, line 2700 should read:

```
2700 DATA 0123,0000
```

(i.e. the last value consists of four zeros, not three). We are well aware of the difficulties involved in updating programs, particularly when renumbering is essential to accommodate the additional lines, and are sorry that we failed to get every single detail correct.

Play It Again Sam 14

Reviewed by Mike Williams

We have become familiar with Superior Software releasing collections of games, often former best sellers, under the general title of *Play It Again Sam*. These collections generally offer value for money and the latest, number 14 in the series, offers just as good value for money as before.

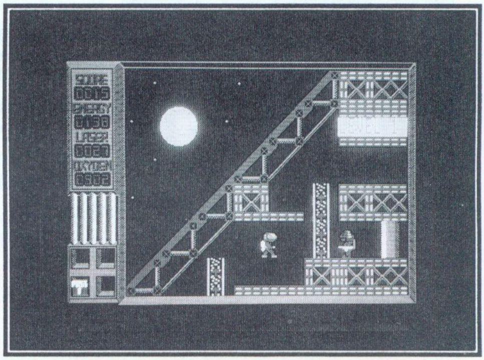
This time we even have one brand new game, Starport included, along with Superior's previous smash hit, *Superior Soccer*. The other games that make up this collection are *Predator*, based on the Schwarzenegger film, and *Ballistix*, a fast action ball game for one or two players.

Starport is essentially a ladders and platforms game. Your task is to explore the abandoned Star Port Beta, now under attack by hostile alien forces and patrolled by decidedly unfriendly robots, in order to recover five cases of medical supplies before making good your escape. Various objects are located within the Star Port, including supplies of oxygen and keys to open various doors, as well as additional ammunition. Superficially, like many such games, life seems comparatively easy, but much thought is required to make significant progress. Graphics and animation are about par for this type of game, which provides an interesting challenge to aspiring games players.

Predator is another game in which you have to combat an alien environment, this time a jungle, but progress here is more dependent upon the correct choice of weapon and your prowess at using it as you fight your enemies. Your ultimate aim is to outwit the alien creature which is stalking you all the time for that final, and hopefully successful, confrontation. Not my cup of tea this one, but ideal for the gamester seeking arcade style action.

Ballistix has a decidedly déjà vu feel about it, and in my view is perhaps the weakest of the group. You can drive this as a one (against the computer) or two player game. It is crudely based on the bat and ball style of game which originated many years ago, though here you fire balls from a moving source to persuade a large 'puck' to enter your opponent's goal. If you can stand it, the challenge (in the one-player game) is to beat the computer over 29 levels for ultimate victory. Unfortunately, I for one do not have the tenacity to bother.

Lastly, but by no means least, this disc also contains *Superior Soccer*, a best seller in its own right last Christmas. For those unfamiliar with it, this features both arcade style action in one or two-player formats, where you can pass, dribble, and head the ball as well as tackling other players, taking corners and free kicks etc.



Starport

You can also manage your team through a season of 11 matches, initially starting in Division 4. If you come first or second in the league you are promoted to the next higher division, with the ultimate aim of becoming Division 1 league champions.

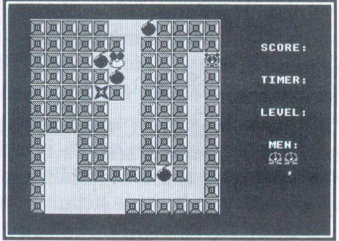
Play It Again Sam 14 is worth the cost for Superior Soccer alone, if that is the type of game which appeals to you, and the new release Starport is well up to standard. The other two games included seem decidedly make-weights, and although others may disagree, I cannot see either of these adding significantly to the attractions of this latest games release.

Product	**Play It Again Sam 14**
Price	**£9.95 (cassette)**
	£11.95 (5.25" DFS disc)
	£14.95 (3.5" ADFS disc)

This game, like all other Superior releases, is available through BEEBUG.

NEW for Christmas!

Arcade Games

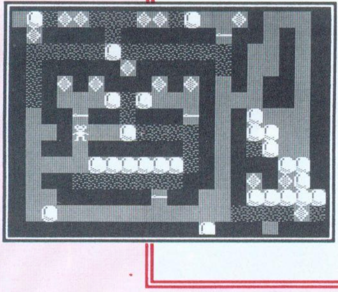


George and the Dragon - Rescue 'Hideous Hilda' from the flames of the dragon, but beware the flying arrows and the moving holes on the floor.

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?

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

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.



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.

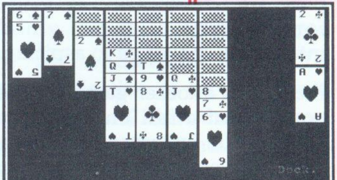
Quad - You will have hours of entertainment trying to get all these different shapes to fit.

Beebug Arcade Games Disc **£5.95** **+ 60p p&p**
Stock Codes **PAG1** (5.25" DFS 40/80T disc) **PAG2** (3.5" ADFS disc)

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'.

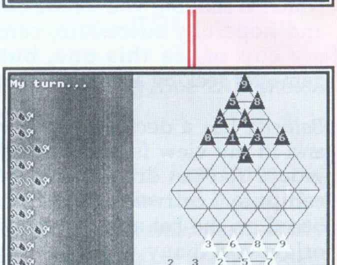


Patience - a very addictive version of one of the oldest and most popular games of Patience.

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.

Beebug Board Games Disc **£5.95** **+ 60p p&p**
Stock Codes **PBG1** (5.25" DFS 40/80T disc) **PBG2** (3.5" ADFS disc)

KRYPTON - File Encryption on the Master 128

David Holton describes a file encryption utility designed to run from sideways RAM and configured for use with Acorn's View word processor.

There are a number of encryption utilities around which will "code" your files, usually on the principle of performing a logical EOR on each byte of the file with a "random" number, or a letter of a "codeword". Having used some, I found them cumbersome and slow - the fastest I could manage in Basic when trying to improve one was about 3 secs/kilobyte, or nearly 90 seconds for the biggest file that View, for example, can handle. It was also a nuisance having to load and use such a program separately from the word processor. What was needed, I thought, was a sideways ROM version.

Many toilsome hours later, the result was *Krypton*. It is called from within View and performs a fairly secure encryption of a maximum-length file (28K or so) in somewhere around 0.4 seconds.

Type in the program from Listing 1, save it and then run it. Try typing *HELP and *ROMS. If all seems well, test it further by going into View, loading a file or typing in some text, and then typing *KRY (no need for a dot) - from the command screen of course. The screen will change to mode 7, and a title appear, followed by the filename, an invitation to enter a key (i.e. "codeword"), and a small window showing a sample of the file (so that you can see if you've got code or plain text - it should be the latter at this stage, of course).

Enter any word, number or string of characters up to 26 in total. They will appear in light blue on a white background, so as to make it hard for anyone to read your key from over your shoulder. Press Return, and the white bar will disappear, the window will instantly show gibberish, and the first menu will appear. This will invite you to (S)ave - i.e. using the existing filename - (R)ename, which saves automatically when you enter the new filename, (U)ndo or (P)ass. Pressing 'U' should instantly show the plain text again, and 'P' should bring up the second menu: (A)gain, (S)creen text or (Q)uit.

'Q' gets you back to View, of course, restoring whatever screen mode you were in. 'S' displays the entire text in a larger window - use Shift to scroll through it, as the program sets page mode (VDU 14). The program shows plain text in rather an odd way; capital letters in lower case, for example. This is because encrypted text contains a lot of 'unprintable' bytes - those with the top bit set or below 32 - so that the print-routine has to mask them off to 0x1xxxxx, with unfortunate results on plain text. What you see is definitely not what you get! Go into View with a decrypted file, and you will see that all is quite normal. Incidentally, pressing Return alone, instead of entering a key, gets you straight to the first menu.

Play around for a while, encrypting a file, doing 'Pass', screening the result, going back to the start by 'Again' and decrypting with the same key. If this works, try 'Save' using an UNIMPORTANT file. Note that the program makes no distinction between encrypting and decrypting - if you keep using the same key, the text will keep switching from one state to the other - this is where the small display window comes in handy.

View will load any file, which is why Krypton has no facilities for loading, but it will not screen or save a file whose last byte is not a Carriage Return (CHR$ &0D); it just gives "No text". This is why Krypton has its own save and display facilities. It would have been very easy to preserve the final Carriage Return of a text, rather than let it be encrypted; but if you try to screen such a file in View, the result is always a crash. Krypton therefore ensures that if you "Quit" back to View with an encrypted file, the text is deleted - so save it first. There's nothing you could do with it, anyway. If the file is plain text, you can almost always get away with quitting via the Break key, though this is not recommended.

The 'Undo' facility is handy in the event that you load an encrypted file and enter the wrong key. The window will show you that something is amiss; if you then enter the correct key immediately you will just double-encrypt it, and will then need both the right and the wrong keys to decrypt the text! To avoid this, if you see that the text is not decrypted, press 'U' for Undo; once you are back where you started, entering the correct key should do the trick. If you value a file and intend to save it encrypted under the old filename, make very sure you know the key!

The method used to print all the prompts (using Y holding an offset from the start of the data) is excellent when working, but a beast to debug, so type in the last part (from line 1680) with scrupulous care, and don't change ANYTHING! A single byte more or less in the data, and every prompt from that point on will be incorrect. Note that the menus have two spaces between each option.

When all is working, delete line 2140 and save the entire listing to a disc with a !BOOT file to CHAIN it - use View to create a suitable file of two lines:

```
*BASIC
CHAIN "Krypton"
```

and save it as !BOOT. Type *OPT4,3 so that the disc will auto-boot. To use the program, just put in the disc and press Shift-Break as usual - Basic will assemble the code, find a vacant slot (if there is one), initialise it and tell you what it has done. Now use View as normal, and the program is at your command. It's not really worth saving the object-code separately; assembly only takes about 1.2 seconds each time.

How does it operate? The real business is done by the two short subroutines *makekeypage* and *main*; all the rest is just there to serve these. Encryption using just the symbols of a key is not very secure - better would be a series of the so-called random numbers generated by Basic, and better still, both.

Random numbers are not easy to generate in machine code, but the sequence of bytes in the computer's ROM is fairly random, taken just as numbers. What Krypton does is to choose a sequence of 256 bytes in the MOS ROM, the starting point being determined by the value of *romkey* (set permanently in line 170) plus an offset calculated from the key by adding up the ASCII values of all the characters, discarding any carry. This sequence from the ROM is EORed with the key (used over and over again) and stored in a page of memory at &DD00. The main loop then uses *keypage* to encrypt the file; this keeps it very simple and therefore very fast.

Security-wise, the effect is the same as using a 256-character, very random key. To see the repeating pattern, fill a page of View with about 30 lines of the same character, and encrypt with a one character key. Screen it within Krypton, and look hard; it repeats every 256 characters - but try to find a pattern with normal text! With a codeword only system, a near miss at the key would reveal a lot of the text, and the correct key would soon be found. Try deleting EOR(r_k_lo) in line 890. Run the program again, and then encrypt a file using PIGGY, and decrypt with POGGY. See what I mean? Now restore line 890 and do the same again - no pattern.

More secure still would be to allow the user to specify the offset separately, but this was discarded in the interest of simplicity. As it stands, it is rather more secure than a simple keyword system against anyone who has a copy of this program, and very much more secure otherwise. Also, when typing in line 170 you can set the value of *romkey* to any value between &E000 and &FB00 - even a change of one byte makes the encryption secure against anyone who has both the program and your keyword, unless their copy of the program has *romkey* set to exactly the same value as yours.

NOTE: We would be pleased to give modifications to permit this program to work with other word processors on the Master 128 if any readers are able to submit relevant details.

```
   10 REM Program Krypton
   20 REM Version B1.3
   30 REM Author  David Holton
   40 REM BEEBUG  December 1990
   50 REM Program subject to copyright
   60 :
  100 adlo=&60:adhi=&61
```

```
 110 ctr_lo=&62:ctr_hi=&63
 120 pveclo=&64:pvechi=&65
 130 r_k_lo=&66:r_k_hi=&67:comvec=&F2
 140 key=&DD00:p_blk=&DE00
 150 k_ofst=&DE1D:str_len=&DE1E
 160 mode=&DE1F:k_str=&DE20
 170 f_name=&07EC:romkey=&F000
 180 osfile=&FFDD:osnewl=&FFE7
 190 oswrch=&FFEE:osbyte=&FFF4
 200 FORN%=4TO6STEP2:P%=&8000:O%=&6000
 210 [:OPT N%
 220 BRK:BRK:BRK:JMPcomcheck
 230 EQUB&82:EQUBoffset MOD256
 240 EQUB2:EQUS"KRYPTON":EQUB0
 250 .offset EQUB0
 260 EQUS"(C) David Holton MCMXC":EQUB0
 270 .comcheck PHA:PHX:PHY
 280 CMP#4:BEQis_it_ours
 290 CMP#9:BNEnot_ours
 300 LDA(comvec),Y:CMP#ASC".":BEQhelp
 310 CMP#&0D:BNEnot_ours
 320 .help JSRsetup:JSRosnewl:LDY#&0F
 330 JSRp_prm2
 340 .not_ours PLY:PLX:PLA:RTS
 350 .is_it_ours LDX#2
 360 .i_i_o_loop LDA(comvec),Y:AND#&DF
 370 CMPhelp_string,X:BNEnot_ours
 380 INY:INX:CPX#5:BNEi_i_o_loop
 390 .its_for_us JSRsetup
 400 DEY:LDA#&FC:JSRosbyte
 410 CPX#&0E:BEQany_text:JSRosnewl
 420 LDY#&39:JSRp_prm2:JSRosnewl
 430 BRAerr_xit
 440 .any_text LDActr_hi:BNEany_f_name
 450 LDActr_lo:CMP#1:BNEany_f_name
 460 LDY#&53:JSRp_prm2:BRAerr_xit
 470 .any_f_name LDA&3C:BNEgo_ahead
 480 LDY#&68:JSRp_prm2
 490 .err_xit PLY:PLY:PLY:LDA#0:RTS
 500 .go_ahead PLY:PLY:PLY
 510 LDA&0355:ORA#&80:STAmode
 520 LDA#&10:LDX#0:JSRosbyte
 530 LDA#&D5:LDY#0:LDX#&CD:JSRosbyte
 540 LDA#&D6:LDY#0:LDX#2:JSRosbyte
 550 JSRsetup:JSRp_prm
 560 .re_entry LDY#&5A:JSRp_prm
 570 JSRpfnme:JSRsample
 580 .keyinput LDY#&6F:JSRp_prm
 590 LDA#&19:STAstr_len:JSRgetstring
 600 JSRmakekeypage:LDY#&81:JSRp_prm
 610 .choices1:JSRgetkey:ORA#&20
 620 CMP#ASC"s":BEQsaveit
 630 CMP#ASC"r":BEQrenameit
 640 CMP#ASC"p":BEQpass
 650 CMP#ASC"u":BNEchoices1
```

```
 660 JSRmain:JMPkeyinput
 670 .renameit LDY#&A6:JSRp_prm
 680 LDA#6:STAstr_len:JSRgetstring
 690 LDY#&5A:JSRp_prm:JSRpfnme
 700 .saveit LDY#&C3:JSRp_prm
 710 JSRpfnme:JSRsave
 720 .pass LDY#&D2:JSRp_prm
 730 .choices2 JSRgetkey:ORA#&20
 740 CMP#ASC"q":BEQquit
 750 CMP#ASC "a":BEQrestart
 760 CMP#ASC "s":BNEchoices2
 770 JSRdisplaytext:BRApass
 780 .restart JSRbigwindow:JMPre_entry
 790 .quit LDA#&16:JSRoswrch
 800 LDAmode:JSRoswrch
 810 LDA#&8E:LDX#&0E:JSRosbyte:LDA#0:RT
S
 820 .makekeypage LDAk_str
 830 CMP#&0D:BEQend
 840 LDA#romkey MOD&100:CLC:ADCk_ofst
 850 STAr_k_lo:LDA#romkey DIV&100
 860 ADC#0:STAr_k_hi:LDY#0
 870 .od:LDX#0
 880 .mkp_loop LDAk_str,X:CMP#&0D:BEQod
 890 EOR(r_k_lo):STAkey,Y:INX
 900 INCr_k_lo:BNErkhi_ok:INCr_k_hi
 910 .rkhi_ok INY:BNEmkp_loop
 920 .main JSRsetup:LDXctr_lo
 930 .codeloop LDA(adlo),Y:EORkey,Y
 940 STA(adlo),Y:INY:BNEcount:INCadhi
 950 .count DEX:BNEcodeloop
 960 DECctr_hi:BPLcodeloop
 970 .sample JSRbigwindow
 980 LDY#0:JSRp_prm2:JSRsetup
 990 LDActr_hi:BEQsampleloop
1000 LDY#&81:STZctr_lo
1010 .sampleloop JSRp_ch
1020 CPYctr_lo:BNEsampleloop
1030 LDA#&1A:JSRoswrch
1040 .end RTS
1050 .displaytext JSRbigwindow
1060 LDA#&0E:JSRoswrch
1070 JSRsetup:LDXctr_lo
1080 .dsp_lp JSRp_ch:BNEctr:INCadhi
1090 .ctr DEX:BNEdsp_lp
1100 DECctr_hi:BPLdsp_lp
1110 LDA#&0F:JSRoswrch:JSRosnewl:RTS
1120 .setup
1130 LDA#prompts DIV&100:STApvechi
1140 LDA#prompts MOD&100:STApveclo
1150 LDA&0B:STAadlo:LDA&0C:STAadhi
1160 LDA&0D:SEC:SBC&0B:STActr_lo
1170 LDA&0E:SBC&0C:STActr_hi
1180 LDX#0:LDY#0:RTS
1190 .getkey LDA#&0E:LDX#0:JSRosbyte
```

```
1200 .inky CLC:LDA#&91:LDX#0:JSRosbyte
1210 BCSinky:LDA#7:JSRoswrch:TYA:RTS
1220 .getstring STZk_ofst
1230 LDA#&CA:LDX#&60:JSRosbyte
1240 LDA#&76:JSRosbyte:LDY#&FF
1250 .getstrloop INY:PHY:JSRgetkey
1260 PLY:CMP#&7F:BEQdelete
1270 LDXstr_len:CPX#&19:BEQnot_f_n
1280 STAf_name,Y
1290 .not_f_n STAk_str,Y:JSRoswrch
1300 CMP#&0D:BEQcalc_k_ofst
1310 CPY str_len:BNEgetstrloop:INY
1320 LDA#&0D:STAf_name,Y:STAk_str,Y
1330 .calc_k_ofst CLC:LDAk_str,Y
1340 ADCk_ofst:STAk_ofst
1350 DEY:BPLcalc_k_ofst
1360 LDY#7:JSRp_prm2:RTS
1370 .delete DEY:BMIgetstring
1380 JSRoswrch:BRAgetstrloop
1390 .save LDA#0:LDY#&FF
1400 .clr_pb_loop INY:STAp_blk,Y
1410 CPY#&15:BNEclr_pb_loop
1420 LDA&0B:STAp_blk+&0A
1430 STAp_blk+6:STAp_blk+2
1440 LDA&0D:STAp_blk+&0E
1450 LDA&0C:STAp_blk+&0B
1460 STAp_blk+7:STAp_blk+3
1470 LDA&0E:STAp_blk+&0F
1480 LDA#f_name MOD&100:STAp_blk
1490 LDA#f_name DIV&100:STAp_blk+1
1500 LDX#p_blk MOD &100
1510 LDY#p_blk DIV &100
1520 LDA#0:JSRosfile:RTS
1530 .bigwindow LDY#&F5
1540 .p_prm LDA(pveclo),Y:JSRoswrch
1550 INY:CMP#&1B:BNEp_prm:RTS
1560 .p_prm2 INCpvechi:JSRp_prm
1570 DECpvechi:RTS
1580 .p_ch LDA(adlo),Y:AND#&7F:ORA#&20
1590 CMP#ASC":":BNEprintit:LDA#&20
1600 .printit JSRoswrch:INY:RTS
1610 .pfnme LDY#0
1620 .pfnloop LDAf_name,Y:CMP#&0D:BEQbo
1630 JSRoswrch:INY:BRApfnloop
1640 .bo LDA#&20
1650 .b_o_loop INY:JSRoswrch:CPY#&15
1660 BNEb_o_loop:RTS
1670 .prompts
1680 EQUB7:EQUB22:EQUB135
1690 EQUB23:EQUB1:EQUD0:EQUD0
1700 EQUB31:EQUB5:EQUB2:EQUB141
1710 EQUB135:EQUB157:EQUB129
1720 EQUS"FILE ENCRYPTION"
1730 EQUB135:EQUB32:EQUB156
1740 EQUB31:EQUB5:EQUB3:EQUB141
```

```
 1750 EQUB135:EQUB157:EQUB129
 1760 EQUS"FILE ENCRYPTION"
 1770 EQUB135:EQUB32:EQUB156
 1780 EQUB31:EQUB4:EQUB5:EQUB131
 1790 EQUS"     By David Holton     "
 1800 EQUB27
 1810 EQUB31:EQUB0:EQUB7:EQUB134
 1820 EQUS"File in memory:"
 1830 EQUB135:EQUB27
 1840 EQUB31:EQUB0:EQUB9:EQUB129
 1850 EQUS"Enter Key:"
 1860 EQUB135:EQUB157:EQUB134:EQUB27
 1870 EQUB31:EQUB0:EQUB16:EQUB131
 1880 EQUS"(S)ave  (R)ename  (U)ndo  (P)
ass"
 1890 EQUB27
 1900 EQUB31:EQUB0:EQUB18:EQUB129
 1910 EQUS"Enter filename to save:"
 1920 EQUB135:EQUB27
 1930 EQUB31:EQUB0:EQUB18:EQUB134
 1940 EQUS"Saving as"
 1950 EQUB135:EQUB27
 1960 EQUB13:EQUB10:EQUB10:EQUB131
 1970 EQUS"(A)gain  (S)creen text  (Q)ui
t"
 1980 EQUB27:EQUB28:EQUB0:EQUB24
 1990 EQUB39:EQUB11:EQUB12:EQUB27:EQUD0
 2000 EQUB28:EQUB4:EQUB14:EQUB35:EQUB11
 2010 EQUB12:EQUB27:EQUB28:EQUB12:EQUB9
 2020 EQUB39:EQUB9:EQUB12:EQUB26:EQUB27
 2030 EQUS"Krypton - the amazing View co
der"
 2040 EQUB13:EQUB10
 2050 .help_string
 2060 EQUS"  KRY - Only from inside View
."
 2070 EQUB13:EQUB10:EQUB27:EQUB13:EQUB10
 2080 EQUS"No Text - Sorry!"
 2090 EQUB13:EQUB10:EQUB10:EQUB27
 2100 EQUB13:EQUB10
 2110 EQUS"No filename - SORRY!"
 2120 EQUB13:EQUB10:EQUB10:EQUB27
 2130 ]:NEXT
 2140 ?677=0:REM ***** REMOVE WHEN DEBUG
GING FINISHED! ******
 2150 FORn%=4TO7
 2160 IF?(n%+673)=0 GOTO2190
 2170 NEXT
 2180 PRINT'''"Sorry - all slots full.":
END
 2190 OSCLI"SRWRITE 6000 "+STR$~(1+O%)+"
 8000 "+STR$n%
 2200 ?(673+n%)=130
 2210 PRINT'''"Krypton installed and rea
dy in slot ";n%'''
```

Shuzzle

A devious and taxing game by Ian Palmer for the approaching festive season.

Shuzzle is a puzzle which is simple in concept, but difficult to solve. Essentially it comprises a 4x4 grid, with one extra box on the left hand side, in which there are 16 coloured shapes, four in each of four colours, red, green, magenta and blue. Each shape/colour has a characteristic movement selected at random at the start from those detailed below.

There are three movements that are possible:

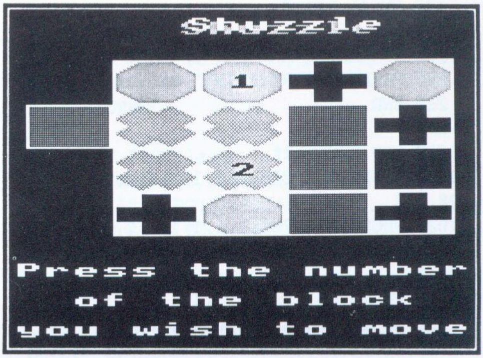
1. **Knight** - this movement is like that of a knight in chess, i.e. it moves forward two squares and then sideways one square (or sideways two, forward/backward one).
2. **Transversal** - this movement is a simple one. A piece can move diagonally any number of squares.
3. **Two step** - this movement is simply a step of two in any direction horizontally or vertically, but not diagonally.

At each stage of the puzzle you will be shown which squares can move into the empty section of the grid. All you need to do is to press the number of the square you wish to move. The object of the puzzle is to arrange the grid such that each row, column and main diagonal has only one of each shape/colour in it, and that no shape is in the extra segment on the left hand side. What could be easier?

Just type in the relatively short program listed here, save it and debug it. When you run it you will be given instructions similar to those above. There will be a message asking you to wait as the computer sets up the puzzle for you; this should take about 20 seconds. Then the message will change to 'Select level 1-3'. There are three levels of play, the first being the easiest (well everything is relative...), the third the hardest. On level one you can always complete the puzzle by heading for the following grid structure:

Red	**Green**	**Magenta**	**Blue**
Magenta	**Blue**	**Red**	**Green**
Blue	**Magenta**	**Green**	**Red**
Green	**Red**	**Blue**	**Magenta**

Escape returns you to the instruction page, and starts a new puzzle.



```
Shuzzle

Press the number
of the block
you wish to move
```

Shuzzle level 1 screen

Shift-Escape exits from the program.

On level two, only three colours will be displayed - four shapes of each of two colours, eight shapes of the remaining colour. Again you must get one of each 'colour' in each row, etc.

This may seem impossible at first, but the colour which appears eight times is in fact two subsets of four shapes which are just displayed as a single colour to confuse you. The only way you can tell the two apart is that the two subsets have differing movement characteristics.

On level three, only two colours will be displayed, in two sets of eight. Again, like level two, each set is made up of two subsets, and each subset has a different movement characteristic from its partner to distinguish one from the other. In this puzzle you need to keep track of what all 16 squares' real identities are.

You may find a piece of paper and a pen helpful when you first try this level (and indeed for level two), but to solve Shuzzle properly you really should do so without any other aids!

```
   10 REM Program Shuzzle
   20 REM Version B2.00
   30 REM Author  Ian Palmer
   40 REM BEEBUG  December 1990
   50 REM Program subject to copyright
   60 :
  100 PROCinit:ON ERROR GOTO190
  110 REPEAT:MODE7
  120 PROCinst
  130 PROCshuffle
  140 MODE2:PROCgame
  150 PROCflash
  160 PROCdisint
  170 UNTIL FALSE
  180 :
  190 IF ERR=17 AND NOT INKEY-1 GOTO 110
  200 MODE7:REPORT:PRINT" at line ";ERL
  210 END
  220 :
 1000 DEF PROCinit
 1010 VDU23,224,255,128,128,128,128,128,
128,128
 1020 VDU23,225,255,1,1,1,1,1,1,1
 1030 VDU23,226,128,128,128,128,128,128,
128,255
 1040 VDU23,227,1,1,1,1,1,1,1,255
 1050 VDU23,228,255,0,0,0,0,0,0,0
 1060 VDU23,229,128,128,128,128,128,128,
128,128
 1070 VDU23,230,1,1,1,1,1,1,1,1
 1080 VDU23,231,0,0,0,0,0,0,0,255
 1090 VDU23,232,255,254,252,248,240,224,
192,128
 1100 VDU23,233,255,127,63,31,15,7,3,1
 1110 VDU23,234,128,192,224,240,248,252,
254,255
 1120 VDU23,235,1,3,7,15,31,63,127,255
 1130 VDU23,236,255,126,60,24,0,0,0,0
 1140 VDU23,237,128,192,224,240,240,224,
192,128
 1150 VDU23,238,1,3,7,15,15,7,3,1
 1160 VDU23,239,0,0,0,0,24,60,126,255
 1170 VDU23,240,255,255,255,255,255,255,
255,255
 1180 DIM S%(5,5),R%(4),P%(18,1),c%(5),P
$(4),bx%(1),by%(1),bbx%(1),bby%(1)
 1190 RESTORE 1250
 1200 FOR A%=1 TO 4:P$(A%)=""
 1210 FOR B%=1 TO 17
 1220 READ C%:P$(A%)=P$(A%)+CHR$(C%)
 1230 NEXT:NEXT:P$(0)=P$(1)
 1240 ENDPROC
 1250 DATA 224,228,225,10,8,8,8,229,32,2
30,10,8,8,8,226,231,227
 1260 DATA 232,228,233,10,8,8,8,229,32,2
30,10,8,8,8,234,231,235
 1270 DATA 240,228,240,10,8,8,8,229,32,2
30,10,8,8,8,240,231,240
 1280 DATA 232,236,233,10,8,8,8,237,32,2
38,10,8,8,8,234,239,235
 1290 :
 1300 DEF PROCgame
 1310 VDU23;11,0;0;0;0;19,5,4;0;19,3,5;0
;
```

```
 1320 GCOL0,5:MOVE400,944:VDU5
 1330 PRINT"Shuzzle":GCOL0,7:MOVE416,952
:PRINT"Shuzzle":VDU4
 1340 bx%(0)=8:by%(0)=-4:bbx%(0)=8
 1350 bby%(0)=4:bx%(1)=1280
 1360 by%(1)=1024:bbx%(1)=-8
 1370 bby%(1)=-4:bb%=0
 1380 COLOUR7:PRINTTAB(2,23);"Press the
number";TAB(4,25);"of the block";TAB(2,2
7);"you wish to move"
 1390 REPEAT:PROCdisplay
 1400 P%=FNpos(TRUE):REPEAT:T%=VAL(GET$)
:UNTIL (T%>0 AND T%<=P%):PROCdo(T%)
 1410 UNTIL FNcheck:PROCdisplay
 1420 ENDPROC
 1430 :
 1440 DEF FNok(x%,y%,X%,Y%)
 1450 h%=ABS(x%-X%):v%=ABS(y%-Y%)
 1460 r%=S%(x%,y%):IF r%=0 OR r%>4 =FALS
E
 1470 IF S%(X%,Y%)<>0 =FALSE
 1480 r%=R%(r%)
 1490 IF r%=1 AND h%=1 AND v%=2 =TRUE
 1500 IF r%=1 AND h%=2 AND v%=1 =TRUE
 1510 IF r%=2 AND h%=v% =TRUE
 1520 IF r%=3 AND h%=2 AND v%=0 =TRUE
 1530 IF r%=3 AND h%=0 AND v%=2 =TRUE
 1540 =FALSE
 1550 :
 1560 DEF PROCswap(K%,x%,y%,X%,Y%)
 1570 S%(X%,Y%)=S%(x%,y%):S%(x%,y%)=0
 1580 ENDPROC
 1590 :
 1600 DEF PROCdisplay
 1610 FOR A%=1 TO 5:FOR B%=2 TO 5
 1620 C%=0:BC%=7:COLOUR128+c%(S%(A%,B%))
:COLOUR7
 1630 IF A%>1 OR B%=3 PRINTTAB(A%*3-2,B%
*3-1);P$(c%(S%(A%,B%))):COLOUR128
 1640 NEXT:NEXT:COLOUR7
 1650 ENDPROC
 1660 :
 1670 DEF PROCshuffle
 1680 LOCAL Q%,x%,y%,q%,d%
 1690 d%=0:REPEAT
 1700 RESTORE 1880
 1710 FOR A%=2 TO 5:FOR B%=2 TO 5
 1720 READ S%(A%,B%)
 1730 NEXT:NEXT
 1740 FOR A%=2 TO 5:S%(1,A%)=5:NEXT:S%(1
,3)=0
 1750 d%=RND(6):FOR q%=1 TO d%:FOR A%=1
TO 4:READ R%(A%):NEXT:NEXT
 1760 UNTIL FNpos(FALSE)>0
 1770 x%=1:y%=3
 1780 FOR Q%=1 TO 100
 1790 q%=0:REPEAT:q%=q%+1:X%=RND(5):Y%=R
ND(4)+1:UNTIL FNok(X%,Y%,x%,y%) OR q%=40
 1800 IF q%<40 PROCswap(TRUE,X%,Y%,x%,y%
):x%=X%:y%=Y%
 1810 NEXT
 1820 FOR A%=0 TO 4:c%(A%)=A%:NEXT
 1830 PRINTSPC(11);"Select level 1-3";SP
C(15)
 1840 REPEAT:L$=GET$:UNTIL INSTR("123",L
$)
 1850 IF L$<>"1" FOR A%=1 TO 15:B%=RND(4
):C%=RND(4):D%=c%(B%):c%(B%)=c%(C%):c%(C
%)=D%:P$=P$(C%):P$(C%)=P$(D%):P$(D%)=P$:
NEXT
 1860 IF L$<>"1" c%(2)=c%(1):IF L$="3" c
%(3)=c%(4)
 1870 ENDPROC
 1880 DATA 1,3,4,2,2,4,3,1,3,1,2,4,4,2,1
,3
 1890 DATA 1,2,2,3,1,2,1,2,1,2,1,3,1,3,1
,3,1,3,1,2,1,3,2,3
 1900 :
 1910 DEF FNpos(p%)
 1920 LOCAL A%,B%,x%,y%,P%
 1930 FOR A%=1 TO 5:FOR B%=2 TO 5:IF S%(
A%,B%)=0 x%=A%:y%=B%:A%=6:B%=6
 1940 NEXT:NEXT
 1950 P%=0:FOR A%=1 TO 5:FOR B%=2 TO 5
 1960 IF FNok(A%,B%,x%,y%) P%=P%+1:IF p%
 P%(P%,0)=A%:P%(P%,1)=B%:COLOUR0:COLOUR1
28+c%(S%(A%,B%)):PRINTTAB(A%*3-1,B%*3);P
%:COLOUR128:COLOUR7
 1970 NEXT:NEXT
 1980 =P%
 1990 :
 2000 DEF PROCdo(T%)
 2010 FOR A%=1 TO 5:FOR B%=2 TO 5:IF S%(
A%,B%)=0 x%=A%:y%=B%:A%=6:B%=6
 2020 NEXT:NEXT
 2030 X%=P%(T%,0):Y%=P%(T%,1)
 2040 PROCswap(FALSE,X%,Y%,x%,y%):ENDPRO
C
 2050 :
 2060 DEF FNcheck
 2070 out%=TRUE
 2080 FOR A%=2 TO 5:PROCcl
 2090 FOR B%=2 TO 5:S%(0,S%(A%,B%))=TRUE
 2100 S%(S%(B%,A%),0)=TRUE:NEXT
 2110 FOR B%=1 TO 4:IF S%(0,B%)=0 OR S%(
B%,0)=0 out%=FALSE:A%=5
 2120 NEXT:NEXT:IF out%=FALSE =FALSE
 2130 PROCcl:FOR A%=2 TO 5
 2140 S%(0,S%(A%,A%))=TRUE:S%(S%(A%,7-A%
),0)=TRUE
```

```
 2150 NEXT:FORB%=1 TO 4:IF S%(0,B%)=0 OR
S%(B%,0)=0 out%=FALSE:A%=5
 2160 NEXT:=out%
 2170 :
 2180 DEF PROCcl
 2190 LOCAL A%
 2200 FOR A%=1 TO 4:S%(0,A%)=0:S%(A%,0)=
0:NEXT
 2210 ENDPROC
 2220 :
 2230 DEF PROCflash
 2240 PRINTTAB(1,8);SPC(3);TAB(1,9);SPC(
3);TAB(1,10);SPC(3);TAB(0,23);
 2250 FOR A%=1 TO 4:c%(A%)=A%:NEXT
 2260 FOR A%=1 TO 150:SOUND&10,-A%/15,4,
4:IF (A% MOD 2)=1 PRINT" ";:IF POS=0 PRI
NT
 2270 FOR B%=1 TO 4:SOUND&11,-10,B%*50,1
:c%(B%)=c%(B%)MOD4+1:VDU19,B%,c%(B%);0;:
NEXT
 2280 NEXT
 2290 ENDPROC
 2300 :
 2310 DEF PROCdisint
 2320 FOR A%=1 TO 4:VDU19,A%,0;0;:NEXT:F
OR A%=0 TO 14:SOUND0,-15+A%,6,(20-A%)/2:
NEXT:PRINTTAB(5,20);"Well  done"
 2330 TIME=0:REPEAT:UNTIL TIME>=700:ENDP
ROC
 2340 :
 2350 DEF PROCinst
 2360 VDU23;11,0;0;0;0;
 2370 PRINTTAB(14,1);CHR$(141);CHR$(130)
;"Shuzzle";TAB(14,2);CHR$(141);CHR$(130)
;"Shuzzle"
 2380 PRINTTAB(0,5);" This puzzle is pla
yed on a 4x4 grid     with one extra box
on the left side.     Each box, except on
e, has a coloured     shape in it."
 2390 PRINT" Each colour can move in one
 of three    ways : like a knight in ches
s, in any    direction two steps, or diag
onally any   number of steps."
 2400 PRINT" You must get the grid so th
at there is  only one of each colour in e
ach line     horizontally, vertically and
 on the two main diagonals."
 2410 PRINT" At any stage you will be sh
own which    blocks can move into the emp
ty slot,     just press the number of the
 block you   wish to move."
 2420 PRINT'"Please wait, I am setting u
p the grid...";CHR$(11);
 2430 ENDPROC
```

Music Discs

Alan Wrigley looks at some of the latest offerings for the Music 5000.

Product	**Music Discs**
Supplier	**Hybrid Technology Ltd. 273 The Science Park, Cambridge CB4 4WE. Tel. (0223) 420360**
Price	**£4.95 each inc. VAT.**

Product	**Music Discs**
Supplier	**JBSoftware 20 Crawley Avenue, Wellingborough, Nothants NN8 3YH. Tel. (0933) 675392**
Price	**£3.95 each inclusive.**

Product	**Music Discs**
Supplier	**Panda Discs Four Seasons, Tinkers Lane, Brewood, Stafford ST19 9DE.**
Price	**£6.00 each inclusive.**

In BEEBUG Vol.8 No.10 I reviewed a number of music discs for the Hybrid Music 5000 synthesiser. Since then, several more have been issued, and in this review I shall be looking at some new releases from Panda Discs, one from Hybrid, and several from JBSoftware.

Ashes, from Hybrid, is by Michael Harbour and follows his *Shivering Again* which I reviewed last time. Whereas the theme of the earlier disc was all things cold, the opposite is the case here. The style was immediately recognisable, being the same distinctive blend of catchy tunes and slick instrumentation. Unfortunately this tended to produce a sense of déjà vu in places. In terms of musical content (melodies, chord structures etc.) the disc is not particularly adventurous, but if you like music to listen to rather than to marvel at, you may find this disc appealing.

From Panda comes *Monster Computations 4*. Whereas its predecessor covered in my earlier review was aimed at "little monsters", this is perhaps more for small-to-medium-sized monsters. The disc contains a collection of pieces excellently programmed, including classical favourites, Irish jigs, traditional tunes, original compositions and much more. Some of the screens feature imaginative, and in some cases animated, graphics. I particularly liked the conductor waving his baton on a piece by Vivaldi. This disc will probably not be quite so appealing to younger children as the nursery rhymes on the earlier disc, but I am sure older children, and parents and teachers too, will love it.

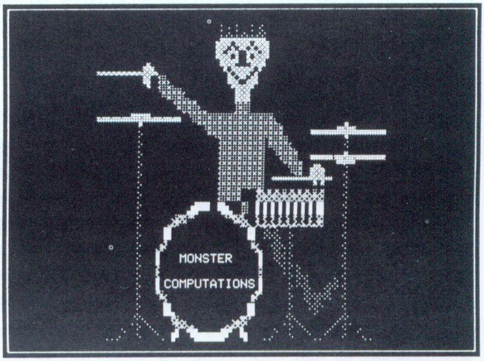
Also recently released by Panda is *Children in Need 2*. Unfortunately, I have not been able to listen to this disc, but it is currently available, and contains the same mix of classical and pop music as its predecessor. Some of the tunes have been specially composed with an environmental theme, and the disc even includes a couple of games. Once again, all profits will be donated to the Children in Need charity, making the disc well worth buying for that reason alone.

The remaining discs are all from JBSoftware, which is run by John Bartlett. All the music on these discs was composed and/or arranged by him. The first I listened to were *Jazz Discs 1 & 2*. Purists may argue that "jazz" is impossible on a computer music system since true jazz relies heavily on improvisation and interaction between musicians for its unique qualities. Nevertheless, jazz does have a distinctive feel to it, and John Bartlett deserves credit for attempting to capture this in the computer environment. What is more, to a large extent he succeeds. He clearly has a great sympathy with jazz, and if this is your kind of music you will enjoy these discs very much. The music is varied, sometimes bright, sometimes moody, never dull. In fact, after listening to these two discs, I felt that the Hybrid system is an ideal medium for jazz, and captures perfectly the inherent intimacy of a jazz performance, which is dependent on a small number of virtuoso instruments coming together under the creative genius of an imaginative composer.

Guitars, also from JBSoftware, is a collection of transcriptions of guitar works, and original compositions in guitar style. The music ranges from classical through Spanish style to blues. Unfortunately, the Hybrid 5000 struggles a little to imitate a guitar accurately, and despite the best efforts of the programmer it still sounds like a synthesiser. Nevertheless, the feel of guitar music is there, particularly in John Bartlett's tributes to some of the great blues guitarists, right down to the steely sound and the raps on the sound board. This is a very courageous undertaking, bearing in mind the limitations of the system, but one that has resulted in another enjoyable disc.

The next offering from JBSoftware is *Impressions*. This is described as a collection of musical images and personal impressions gathered over a number of years, and relates mostly to places around the world. Each piece attempts to capture the essence of the place, from the dreamy nature of the southern seas, with steel guitars, seagulls and waves crashing in the distance, to the hectic bustle and chic sophistication of New York. Many of the tracks have a jazzy feel to them, again indicating John Bartlett's fascination with that medium. This is another excellent disc. Each track is different, and each successfully captures the spirit of the place depicted (the Hybrid 5000 even copes admirably with bagpipes!).

Finally, I was privileged to hear a disc which was not yet completed, but will be on sale by the time you read this review. This is *Symphony*, certainly the most ambitious project for the 5000 I have yet encountered. As well as works by Vivaldi and Mozart, it contains sections from three movements from Beethoven's Pastoral Symphony, and a three-movement concerto written by John Bartlett himself. Transcribing classical music is nowhere near as easy as it might seem; even if you can successfully imitate all the instruments and include all the notes, you must capture the expression, and the way in which each instrument is actually played, or the music will sound flat and lifeless. John has clearly put a lot of effort into this aspect, with the result that the Beethoven in particular sounds fresh and full of life. When you consider that the eight channels of the 5000 are having to cope with a full symphony orchestra, this is no mean achievement. It is a great shame that the memory limitations of the BBC do not allow a complete Beethoven movement to be played on the system.



A screen display from Monster Computations 4

Quite apart from the difficulties of making existing music come to life, the original concerto on the disc is a major undertaking in itself. The scale of the work is not vast, but the music is potent and full of invention. Needless to say, the inevitable jazz influence creeps into the musical phrasing in one or two places, but the overall impression is of lush orchestral texture.

For all the expensive and sophisticated technology employed nowadays in the commercial music business, in the end it all comes down to the quality of the music itself. Every one of these discs from JBSoftware is a great advertisement for the Music 5000 system, and shows how far it can be pushed when imagination is coupled with a real and instinctive feeling for music.

This month's magazine disc contains samples from two of the discs reviewed here. Morris the Cat *comes from* Jazz Disc 1, *and* Purcell's Rondo *is from* Monster Computations 4.

Better Programming (2)

By Mike Williams

I have decided to devote one further instalment of First Course to the subject which I introduced last time, and that is the subject of good programming. When setting out to develop a program to perform a particular function there is a lot of satisfaction to be gained by not only achieving the desired objective, but by knowing that good programming methods have been used as well.

Last month I gave some examples of what I considered to be bad programming practice, together with ways in which I thought the coding could be improved, and I want to give one further example of bad practice which has come my way recently. I then propose to devote the remainder of this article to looking at more positive and constructive ways of achieving that goal of good programming.

First of all, consider the following piece of code (where 'n' represents a line number), presented here just in outline:

```
  MODE 3
  PROCinit
n REPEAT
  PROCtitle
  .................
  PROCcalculate
  .................
  END

  DEF PROCcalculate
  IF X<0 OR Y<0 THEN n
  .................
  .................
  ENDPROC
```

The detailed nature of the instructions given above is unimportant; it is the structure which counts. The procedure definition contains a GOTO to a point in the program which is totally outside of the procedure. I can see that; you can see that; but to Basic, once it has entered the procedure, it will believe that it remains within the procedure unless it encounters an ENDPROC.

Now I have little doubt that when the author tested this piece of code it appeared to work as required, and some of you may argue that if that is so then what more is there to say. However, if the program in question had been tested exhaustively, then I am sure the error would have materialised, and it would have done so with the error message "No room at line ....".

Let's see why. As I said, Basic believes it has entered a procedure, and so it will save in memory sufficient information to allow it to return to the point in the program from which it was called (when it encounters the corresponding ENDPROC). The memory used for this purpose is organised as a *stack*. When any information is stored in a stack it is always the last item entered which is the first to be retrieved. So if from within one procedure, another is called, the *return address* goes onto the stack. On exit from that procedure it is *that* return address which is retrieved, leaving the original return address now exposed. This simple mechanism means that within the limits of available memory, procedures can be nested as deeply as you like. The same mechanism is used for nested loops, either using FOR-NEXT or REPEAT-UNTIL.

To return to our example, if the GOTO is followed back to the main program and events dictate that the same route is followed repeatedly (i.e. a call to the procedure followed by a GOTO back to the main program), the stack will rapidly fill up with return addresses, and eventually the "No room" error will appear.

Now it may be that in practice it is highly unlikely that the same path through the program will be repeated sufficient times to cause the error state, but like so many examples of bad programming, this flawed structure and any other examples in the same program will ultimately make debugging so much more difficult. Further, a variety of subsequent ad hoc modifications on the principle of "if I stick this bit of code in here it now seems to work, even though I don't know why" will lead to a program which is probably much bigger than it needs to be, and which is very difficult to debug or modify subsequently.

A much better solution to the problem is to set a flag (as described last month) before executing a normal exit from the procedure. This flag can then be tested in the main program to determine the next step. Thus the first instruction in the procedure definition might be rewritten:

```
flag=FALSE
IF X<0 OR Y<0 THEN flag=TRUE:ENDPROC
```

In the main program, the call to PROCcalculate would be followed by:

```
IF flag THEN ......
```

GOOD PROGRAMMING PRACTICE

What I want to do now is to concentrate on how to write good programs, and this is achieved by design rather than by accident. First of all let us draw one conclusion from the example above: it is always bad programming to jump from inside a FOR-NEXT or REPEAT-UNTIL loop, or from inside a procedure or function definition, to a point outside. If you approach the task of writing a program in the right kind of way the temptation should not arise. If you feel tempted, then recognise what you are about to do, *and find a different way*. As I said before, take a design approach to programming and the problem won't arise. I wonder how many of the bad examples I have seen, some of which I have shown here, arose as quick fixes to a problem, rather than as part of the original plan.

So how should you go about writing a program? Well clearly one approach is to map out a clear outline of the functions you want your program to perform, indeed write these down on paper in some way, before you actually start to write any code. A purist would probably say you should have your entire program planned out in this way before writing a single instruction. Certainly it is prudent to think about a program as a whole, and to identify what are likely to be its major features before plunging in.

Another approach, which is complementary, is to be aware of the features which are common to many programs. Often, the user has to choose from a series of options, and this gives an immediate shape to the program, thus:

```
Select mode etc
Call initialisation procedure
Repeat
Call menu procedure
IF choice 1 THEN call procedure_one
IF choice 2 THEN call procedure_two
IF choice 3 THEN call procedure_three
.........................
.........................
UNTIL end
Procedure/Function definitions
```

I have deliberately written this in a mixture of Basic and English - a kind of pseudo language. At this stage we are not really interested in the small print. The other point, which is really fundamental to good programming is the immediate use of procedures (or functions as may be appropriate) even in this simple outline. Good planning and the free use of procedures will do a lot to ensure your programs are well written.

Once you have an outline as above, think about it and refine it a bit further. Maybe some of the procedures you postulate would be better subdivided into two or more simpler procedures. Have all the functions which you can think of been included? Once you have a rough outline you can begin to turn it into real code. On a practical note it is often advisable to number your program in large steps, say 100, at this stage so that there is plenty of room to insert more code later as things develop.

The beauty of this approach, too, is that it allows you to concentrate on just a small part of the program at a time. For starters, the outline above can be readily converted into a proper program:

```
100 MODE 3
200 PROCinit
300 PROCtitle
400 PROCmenu
500 REPEAT
600 IF choice=1 THEN PROCload
```

```
700 IF choice=2 THEN PROCsave
800 IF choice=3 THEN PROCedit
900 UNTIL FALSE
1000 END

1500 DEF PROCinit
1900 ENDPROC

2000 DEF PROCtitle
2400 ENDPROC

2500 DEF PROCmenu
2900 ENDPROC

3000 DEF PROCload
3400 ENDPROC

3500 DEF PROCsave
3900 ENDPROC

4000 DEF PROCedit
4400 ENDPROC
```

If you enter and run such a program it will work perfectly - and do nothing. But now you can go back and begin to fill in the detail for each procedure definition, or add additional lines to the main program. Because the program 'works' correctly from the word 'go', you don't have to write the complete program before starting to test it.

This is particularly true with regard to the menu choices. In each case the corresponding procedure can be developed and tested on its own.

It is also important to keep things simple. You may often find a situation developing where the tasks to be undertaken in an IF statement, for example, get longer and longer. As soon as this starts to happen put all the following code as a new procedure definition so that the IF statement becomes:

```
IF <condition> THEN PROCwhatever
```

and at a suitable point:

```
DEF PROCwhatever
................
ENDPROC
```

A program which is developed in this way is not only likely to end up a better program, but is easier to test and to get working correctly. This approach, starting from a general description which is gradually developed with more and more detail is called a *top-down* approach to programming.

Of course, nothing ever runs smoothly, and you are likely to find that you need to modify your ideas as you progress, and indeed even throw away on occasion some of your coding. I would advise that you go back to your original paper design if severe problems begin to develop. Think carefully about what it is you are trying to achieve, and clarify your thoughts as much as possible before continuing, rather than hastily attempting quick fixes, which are as likely to create problems as they are to solve them.

Let's finish now with a little more practical advice. I have advocated the extensive use of functions and procedures. One problem which arises is keeping track of which variables are in use at any one time. As far as possible, make full use of the parameter passing capabilities of BBC Basic, and declare as LOCAL any variables which are genuinely local (i.e. used only within the procedure definition). Keep to a minimum the references within a procedure to any variables used outside or within other procedures, and keep a written list of all the variables you do use in this way.

It can also be helpful to designate certain variables as being *temporary* - i.e. they are only ever used for temporary storage such that their use by other parts of the program never matters. For example, you often see the message "Press any key to continue", or similar. This is usually implemented by writing:

```
PRINT"Press any key to continue":G=GET
```

So reserve the variable 'G' to be used in all situations of this kind. Documenting what you are doing, by keeping lists and writing things down, may seem tedious, but it does help you to get yourself properly organised.

Well, that's it for this month. I hope you find these ideas of help in writing your own programs. I certainly do.

Practical Assembler (Part 7)

by Bernard Hill

STAR COMMANDS ON DISC AND FX CALLS

For the last three issues we have been involved with the complexities of interfacing to Basic via the CALL statement with its parameters, and have got very involved with the way in which Basic stores its variables. This month we're returning to what might be called 'pure' assembler, i.e. routines which are intended to be used on a stand-alone basis, and possibly not even from the Basic environment.

Back in Part 1 of this series (BEEBUG Vol.9 No.1) we mentioned the different ways of calling up and executing an assembler routine, and in this article we are going to return to the very common method of running the program direct from disc (or tape). In the article mentioned above we gave a very simple *BOLD routine which turned on bold character printing for an Epson (compatible) printer. It's a very useful idea to have your own set of printer control utilities which are loaded from disc as in that article, and run by star commands. You'll need to look at your printer manual, but the modification of that program to form enlarged, Near Letter Quality (NLQ), superscript, condensed with 10 line per inch spacing - or whatever - is very straightforward.

It's best to try to choose star commands which are not used by any of the ROMs in your computer (or the MOS) as these 'get the first go' at operating from the command. But if you do, then you can always issue the command prefaced by "*/" to send the command direct to the filing system, for instance */BOLD will load and execute the program BOLD even if one of your ROMs responds to that command.

FX AND OSBYTE

When Acorn designed the architecture of the Beeb, they included the notorious FX commands (said to be short for "Effects") which were largely undocumented when the original models A and B first came out. Full documentation is now available however on about 150 FX commands which are available in the model B, and a dozen or so more added to the Master. These OSBYTE calls (as they are also named) are arranged in no real logical order, and either perform an operation on the machine (such as setting the RS423 speed), or set or read a machine status value (such as the printer ignore character). There are about 100 such operating system calls.

BEEPING

One thing which consistently annoys me about the BBC range is that noisy beep which it issues in response to VDU7.

We can moderate this noise with five FX calls. If we would like the beep to be changed to sound like:

```
SOUND C,-A,P,D
```

then FX calls 211-214 handle this by using:

```
*FX 211,C,0
*FX 212,256-8*(A+1),0
*FX 213,P,0
*FX 214,D,0
```

Note that you can't issue *FX calls with variables in them, so you'll need to work out the values as constants and then give those.

Sound can be turned off completely with:

```
*FX210,1,0
```

and on again with:

```
*FX210,0,0
```

Thus, for instance:

```
*FX211,4,0
*FX212,184,0
*FX213,2,0
*FX214,2,0
```

gives a polite little buzz as in:

```
SOUND 4,-8,2,2.
```

To turn this into an assembler program we need to explore the OSBYTE call. This contains three parameters (like FX) which will be placed in the A, X and Y registers, and its address is &FFF4.

Thus:

```
*FX210,1,0
```

is performed in assembler with:

```
LDA #210:LDX #1:LDY #0
JSR &FFF4 \ JSR osbyte
```

There is a subtle difference between FX and OSBYTE however. FX calls simply perform operations whereas OSBYTE can also be used to read values from the system.

Furthermore, any missing parameters on the *FX call are filled in with zeros (so that the third parameters in the FX calls above do not need to be included). With an OSBYTE call we can't be so lax in omitting the trailing zeros (since the registers may already contain other values), and if we are writing a system value (like a *FX instruction) then we must set X to the value to be written and Y to 0. If we are reading, then X must be 0 and Y must be 255: the returned value will be in X. This is the way in which all the OSBYTE commands over 166 work. On reading, any values other than Y=255 and X=0 will modify the OSBYTE status (whatever it was) in an unexpected way. For that reason it is imperative when handling OSBYTE calls in assembler that we set the X and Y registers properly. Be aware too that X and Y will be changed during the OSBYTE call. The value in A, however, is always preserved.

From Basic, the USR function can be used to check this out and read OSBYTE values. Thus:

```
A%=211:X%=0:Y%=255
value=((USR &FFF4) AND &FF00) DIV 256
```

gives the value of the sound channel which VDU7 is using.

Now in fact all the system variables are stored in page 2 at locations &236 to &28F (corresponding to OSBYTES &A6 (166) to &FF) so that in fact our four parameters for the bell are stored in &263-266. We could equally well poke and peek these locations rather than use the correct OSBYTE call. However, this is bad practice as using the call maintains full compatibility with any future changes Acorn may make to the memory map (however unlikely this may seem), and also retains full Tube compatibility.

PARAMETERS ON STAR COMMANDS

It is possible for a star command, running from disc, to find out and use any parameters attached to it, and this is accomplished in a way very similar to that which is used to handle sideways ROM commands (see Part 1 of this series in Vol.9 No.1), though only valid for the DFS.

On entry to the star command routine, the zero page locations F2-F3 and Y will point to the first byte of the rest of the command string so that:

```
LDA (&F2),Y
```

will obtain the byte after the command string, usually a space (ASCII 32) or Carriage Return (ASCII 13). In this way we can easily add parameter handling to the program, and Listing 1 is a program which illustrates all the points we have made so far in this article.

When run it produces a file called BELL which will load in at address &900. If the command *BELL (or */BELL for tape users) is issued then the program resets the VDU7 noise to its default (SOUND 3,-13,101,7). The command:

```
*BELL +
```

will increase the volume (SOUND 3,-14,101,7), and the call can be issued repeatedly to a maximum amplitude of -15. Similarly:

```
*BELL -
```

will decrease the volume incrementally to a minimum amplitude of -1. Further enhancements provided are:

```
*BELL 0      turns off the sound,
*BELL 1      turns it on,
```

and:

```
*BELL Q      replaces the tone with the
             quiet buzz mentioned
             above,

*BELL        with any other character
             behaves like *BELL alone.
```

Listing 1 is very easy to understand. It skips over any spaces following the command and branches into the 6 cases (+,-,0,1,q/Q,other) mentioned above, and performs the requisite OSBYTE calls. Of interest are the '+' and '-' routines which first have to read the OSBYTE value, modify it by 8 and write it back.

HINTS AND TIPS SECTION

A. In many assembler programs we find the structure:

```
.subprog
JSR routinel
JSR routine2
RTS
```

Now the last two of these lines can be combined so that we have:

```
.subprog
JSR routinel
JMP routine2
```

The reason for this is fairly clear: the last executed command of routine2 will be an RTS, so why not let routine2 handle the RTS which we want in this routine? Watch out for these changes when you are typing in programs, the difference is a couple of letters but the program will crash without it! Listing 1 uses this device, where you will see a JMP &FFF4 instead of JSR &FFF4:RTS. The only reason I know for doing this is to save one byte in the code and 3 milliseconds in time. In my opinion, unless you're really squeezed for space (or time) the lack of clarity is a poor bargain for the small saving.

B. A very common programming instruction is:

```
LDY #0:LDA (zpg),Y
```

Now if the Y register is occupied in holding a loop variable or something else important then saving Y and reloading it afterwards is tedious just to be able to do this indirect load. But maybe the X register is free? You could use the much less common:

```
LDX #0:LDA (zpg,X)
```

which in this situation gives the same answer. Of course 65C12 users have the simple LDA (zpg). Much easier!

```
 10 REM *BELL : Practical Assembler
 20 REM Version B1.0
 30 REM Author  Bernard Hill
 40 REM BEEBUG  December 1990
 50 REM Program subject to copyright
 60 :
100 FOR opt=0 TO 2 STEP 2
110 P%=&900
120 [OPT opt
130 .loop
140 LDA (&F2),Y
150 CMP #32 \ space
160 BNE start:INY:JMP loop
170 .start
180 CMP #ASC"0":BEQ offon
190 CMP #ASC"1":BEQ offon
200 CMP #ASC"q":BEQ buzz
210 CMP #ASC"Q":BEQ buzz
220 CMP #ASC"+":BEQ louder
230 CMP #ASC"-":BEQ softer
240 \ otherwise restore to normal
250 LDA #211:LDX #3:LDY #0:JSR &FFF4
260 LDA #212:LDX #144:LDY #0:JSR &FFF4
270 LDA #213:LDX #101:LDY #0:JSR &FFF4
280 LDA #214:LDX #7:LDY #0:JSR &FFF4
290 RTS
300 .buzz \ SOUND 4,-8,2,2
310 LDA #211:LDX #4:LDY #0:JSR &FFF4
320 LDA #212:LDX #184:LDY #0:JSR &FFF4
330 LDA #213:LDX #2:LDY #0:JSR &FFF4
340 LDA #214:LDX #2:LDY #0:JSR &FFF4
350 RTS
360 .offon
370 SEC:SBC #48:EOR #1
380 TAX \ X=0 for on, X=1 for off
390 LDA #210:LDY #0:JMP &FFF4
400 .louder
410 LDA #212:LDX #0:LDY #255:JSR &FFF4
420 CPX #128:BEQ return \ loudest=128
430 TXA:SEC:SBC #8:TAX \ X=X-8
440 .overl LDA #212 :LDY #0:JMP &FFF4
450 .softer
460 LDA #212:LDX #0:LDY #255:JSR &FFF4
470 CPX #240:BEQ return \ softest=240
480 TXA:CLC:ADC #8:TAX
490 LDA #212 :LDY #0:JSR &FFF4
500 .return RTS
510 ]
520 NEXT
530 c$="SAVE BELL 900 "+STR$~P%
540 PRINT "*";c$
550 OSCLI(c$)
560 END
```

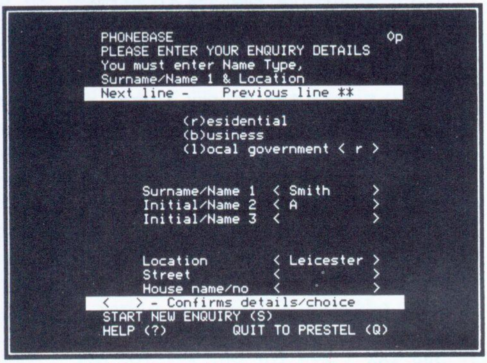
The Comms Spot

Alan Wrigley looks at some recent developments in the world of comms.

It has been a while since we last featured a comms spot, and there have been a number of interesting new developments during that time.

PHONEBASE

From the average domestic user's point of view, possibly the most exciting and potentially useful service launched on Prestel for a long while is Phonebase, which is nothing less than a complete on-line Directory Enquiries. British Telecom has at last taken a leaf out of Minitel's book (for those of you not familiar with Minitel, this is the French equivalent of Prestel which was launched with a national telephone directory as a central feature, and free Minitel terminals given to millions of subscribers).

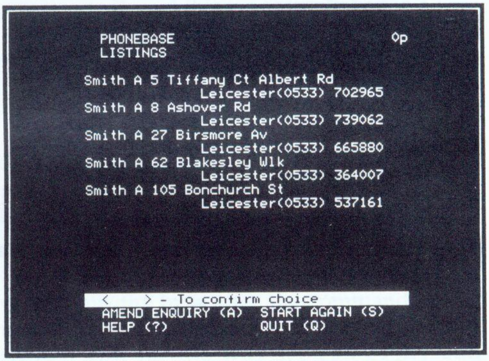


```
PHONEBASE                                   0p
PLEASE ENTER YOUR ENQUIRY DETAILS
You must enter Name Type,
Surname/Name 1 & Location
Next line -     Previous line **

          (r)esidential
          (b)usiness
          (l)ocal government < r >

     Surname/Name 1  < Smith      >
     Initial/Name 2  < A          >
     Initial/Name 3  <            >

     Location        < Leicester  >
     Street          <            >
     House name/no   <            >
 <    > - Confirms details/choice
START NEW ENQUIRY (S)
HELP (?)         QUIT TO PRESTEL (Q)
```

The Phonebase search screen

By typing the keyword **PHONEBASE,* Prestel subscribers can be connected via a gateway to the same database as that used by the operators of the Directory Enquiries service. This database holds all residential, business and local government numbers for the entire country (excluding ex-directory numbers), and is apparently updated over 40,000 times a day. There is no charge for using the gateway, other than the normal Prestel time charges.

The beauty of Phonebase is that you can define your own search criteria and spend as long as you want searching for the right number. Have you ever tried telling Directory Enquiries "The name is Johnson, or it could be Jobson, and I think he lives in Acacia Gardens somewhere in Devon or Cornwall"? Well with Phonebase there is no problem! It may take a little while, but if Mr Johnson or Jobson is on the telephone, you will find him.



```
  PHONEBASE                                   0p
  LISTINGS

Smith A 5 Tiffany Ct Albert Rd
              Leicester(0533) 702965
Smith A 8 Ashover Rd
              Leicester(0533) 739062
Smith A 27 Birsmore Av
              Leicester(0533) 665880
Smith A 62 Blakesley Wlk
              Leicester(0533) 364007
Smith A 105 Bonchurch St
              Leicester(0533) 537161

  <    > - To confirm choice
  AMEND ENQUIRY (A)   START AGAIN (S)
  HELP (?)            QUIT (Q)
```

A list of subscribers found by Phonebase

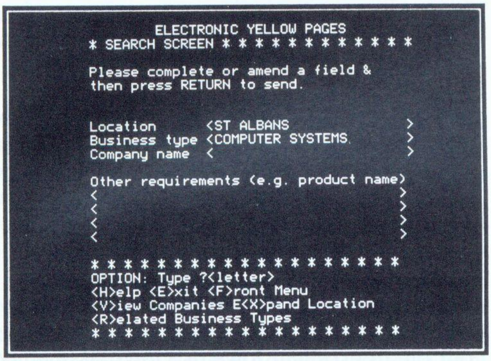
Once through the gateway, Phonebase presents you with a screen on which you specify the requirements for your search. The minimum you need to give is a name (first four letters only will do), a location and whether residential or business etc. The location can be a village, town, city or even a British Telecom region (e.g. ER for Eastern). So you really can find almost anyone if you have the patience. The database is very simple to access, but there is on-line help available at all times if you get stuck.

I can't help feeling that, if services such as Phonebase had been an integral feature of Prestel from the start, then the great dream of a national viewdata service pumping out information to millions of domestic users may just have become a reality. As it is, I can't see many people subscribing to Prestel just for Phonebase, nor can I see the Government following the French lead and giving away free terminals. Nevertheless, the fact that Directory Enquiries now incurs a charge makes Phonebase even more attractive, and will certainly enhance Prestel's usefulness to the business community.

YELLOW PAGES

Electronic Yellow Pages has been around for a little while in a prototype form, but has only recently been expanded to a full national service. Like Phonebase, it is available to Prestel users through a gateway, but there is also a direct access line, which is charged at local call rates from wherever you are. With the Prestel service, there are no Prestel time charges while you are connected to the EYP gateway, which helps to keep access costs down considerably, particularly during the day.

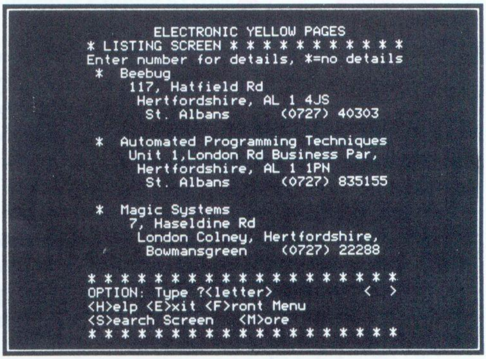
The purpose of the service, as you might expect, is to combine the functions of all the local Yellow Pages volumes into one database. The potential for this kind of information to the business user is enormous - just imagine being able to get a list of every company in any county in the country producing, say, computer software. I can see the output of junk mail rising fast as small businesses begin to discover that they can now get this information for themselves, instead of having to pay someone else to do it.



```
            ELECTRONIC YELLOW PAGES
  * SEARCH SCREEN * * * * * * * * * * * * *

  Please complete or amend a field &
  then press RETURN to send.


  Location      <ST ALBANS                >
  Business type <COMPUTER SYSTEMS.        >
  Company name  <                         >

  Other requirements (e.g. product name)
  <                                       >
  <                                       >
  <                                       >
  <                                       >

  * * * * * * * * * * * * * * * * * * * *
  OPTION: Type ?<letter>
  <H>elp <E>xit <F>ront Menu
  <V>iew Companies E<X>pand Location
  <R>elated Business Types
  * * * * * * * * * * * * * * * * * * * *
```

The EYP search screen

The service is rather more fiddly to use than Phonebase, no doubt reflecting the extra complexity of the information. You can search for companies by name, business category or location. If you try a category which doesn't exist, the system usually offers an intelligent choice of alternatives; for example, if you specify "Magazines", it will suggest "Publishers and Publications", "Journalists" etc. When all the criteria have been specified, a list of matching companies is produced, and in some cases additional information on the company is available, though I'm not sure what form this takes as I have yet to find a company that actually *has* additional information! The only drawback I can see is that, if you ask for a company by name only, the display does not show you which business category that company is in. This would have been useful.



```
          ELECTRONIC YELLOW PAGES
  * LISTING SCREEN * * * * * * * * * * *
  Enter number for details, *=no details
   *  Beebug
       117, Hatfield Rd
        Hertfordshire, AL 1 4JS
         St. Albans       (0727) 40303

   *  Automated Programming Techniques
       Unit 1,London Rd Business Par,
        Hertfordshire, AL 1 1PN
         St. Albans       (0727) 835155

   *  Magic Systems
       7, Haseldine Rd
        London Colney, Hertfordshire,
         Bowmansgreen     (0727) 22288

  * * * * * * * * * * * * * * * * * * * *
  OPTION: Type ?<letter>             < >
  <H>elp <E>xit <F>ront Menu
  <S>earch Screen   <M>ore
  * * * * * * * * * * * * * * * * * * * *
```

Businesses identified by EYP from the search screen shown above

The EYP direct line accepts a range of baud rates, as indicated below. Wait five seconds after the carrier is detected (or the connection is confirmed if you are using a Hayes modem), and press Return. You should see YPUK appear on the screen. Press Return again (or Ctrl-M if you are using viewdata software), and then you will be asked to specify which kind of terminal you are using. Note that parity should be set to 8N1, rather than 7E1 which is normally used to access Prestel.

BULLETIN BOARDS

A few months ago we published a list of bulletin boards which were known to have material for BBC users, and remarked that these were becoming more difficult to find. New boards catering for the Beeb are extremely rare, so I was interested to hear of Modem Mania, a new board operating from Nottinghamshire (see access details below). As might be expected, the board features messaging and downloading facilities, and in common with many other BBC boards these days, there will be a music section. Satellite TV and CB radio will also be featured, and the general intention is that there should not be too much computer emphasis.

Board	**Modem Mania**
Tel. No.	0623 861529
Speeds	V23 viewdata only

Service	**Electronic Yellow Pages**
Tel. No.	0345 444444
Speeds	V21/V22/V22bis/V23

512 Forum

by Robin Burton

This month,let me tell you about a very interesting source of shareware, before going on to talk about piping.

SHAREWARE

I'd guess that the majority of Forum readers are familiar with shareware, but in case there are some new 512 converts who aren't, a brief recap is in order since there is not and has never been an equivalent concept for the BBC micro.

I first covered shareware in 512 Forum back in May 1989 (Vol.8 No.1), so if you didn't see it, dig out the issue (or send for it) and have a look. Shareware is also covered quite extensively in the Dabs 512 User Guide, so there's another useful source of information.

If you're unfamiliar with the concept here's a brief outline. As usual I must appeal to more experienced users for a little patience on behalf of those new to the 512 and PC software. Remember, we all started out knowing almost nothing!

Shareware is software which is not sold in the usual sense of the word, but is supplied in the first place for a nominal charge, so that you can take a look at the programs and decide if they are what you expected. These initial charges vary, but in the U.K. they are typically less than £10.00 and in some cases much less. Moreover, the charge is frequently not made on a 'per application' basis, especially in the case of smaller programs and utilities, but usually per disc, and a disc might contain numerous items as I'll explain shortly.

If the software doesn't suit you that's basically the end of the matter, you've paid a small cost to try it out but that's your only outlay, the disc is yours to keep. On the other hand, if you like the software and intend to continue to use it you are expected to pay a registration fee, after which you can use the software just as if you had bought it. The initial evaluation period is generally accepted to be 30 days, but of course no-one checks on this, it's up to you to be honest and play the game according to the rules. You are, however, encouraged to pass on shareware to other users, and can therefore share the initial cost if you know other PC or 512 users.

Registration fees are generally a fraction of the sort of price you'd expect to pay for software sold through dealers and other commercial channels because there's no advertising and no 'middle man' to take a substantial profit. The most important point for the 512 of course is the usual risk when you try out new PC software, that is it might not work. Shareware is therefore even more of a benefit to us than it is to PC users, since it means that a 'failure' isn't a financial disaster and you only pay for software when you know it does the job.

Finally, don't get the idea that shareware is sub-standard or poor quality software, it isn't. Just like commercial software some of it isn't so good, but some shareware applications and packages are truly excellent and as good as, or better, than anything you can buy anywhere.

That's enough revision, if you didn't know about shareware before, you've been missing out on an excellent low-cost, low-risk method of getting hold of PC software legitimately.

ABSTRACT SHAREWARE

The title of this section does not refer to software which performs theoretical rather than practical tasks, but to the name of a supplier of shareware.

I became aware of this supplier several months ago from a friend who uses PCs at work. The usual way that shareware is supplied on disc is that programs are divided into categories and each disc contains one or more programs concerned with a particular topic or application. In the case of larger applications one disc may well contain only the one application and nothing else.

Of course, some shareware programs are small but useful utilities which, on their own couldn't justify a disc, so shareware suppliers tend to group a number of these together, sometimes a dozen or more at a time, to make up a complete disc. That's fine, but it can sometimes mean that to get three or four particular programs you might be obliged to send for three or four separate discs from the catalogue. It's true the cost isn't huge, but nevertheless it can rather limit the attractiveness of the whole idea of shareware.

Abstract has addressed this problem in a novel, and so far as I know unique way. Initially they send you a couple of floppy discs which contain their complete software catalogue plus programs which you use to make your selections. There's a small gripe here in that these two discs can't be copied to and run from a winchester because the drives used, A: and B:, are permanently coded into the programs, but that's a small criticism which in context may seem a bit churlish.

When you run the selection program you are first presented with a menu which shows all the main categories offered, for example word-processing, spreadsheet, database, programming utilities, disc utilities and so on. You then select the category you're interested in and the selection program reads the appropriate title list from disc. Next you can look at the list and select the particular programs which interest you, each title is accompanied by a brief description to help you. When you've chosen all the titles you want from the current category you return to the main menu and can select another category to browse and select from. There's basically no limit to how many categories or titles you can select, and of course options are also provided to amend or cancel choices you've already made if you change your mind or find a better program later on.

When you've selected all the software you would like to try out, another of the menu options allows you to create an order, and this is where the system is particularly useful. Using the list of software titles you have selected, the ordering routine works out from the sizes of the files, exactly how many discs will be required to hold everything you've ordered. This done it tells you the number of discs and also how much it's going to cost.

The beauty of this system is that there are no 'pre-packed' groups of programs which you must order together, the discs you'll receive will contain nothing but the items you have selected. What's more the programs are archived (compressed) as is usual for shareware, so the number of discs that will be needed is calculated only on the basis of cramming as much onto each one as it can hold. The cost to you is therefore based only on the number of discs needed for your order, it has nothing to do with how many program titles you're going to get, nor with the types of the programs you've selected.

If that doesn't tempt you then the price for this service certainly should. Costs might have increased a bit in the last few months, but if so it's not likely to be much. When I last checked the charge was only £2.90 per disc in 5.25 inch format. There is, as usual, a small premium if you want software on 3.5 inch discs, but even so this is still extraordinary good value.

There's no charge for the two catalogue discs, which can be obtained by telephoning *Abstract* or by writing. The details are given at the end of this Forum.

There are just two final points, again for anyone who's not familiar with PCs and/or is new to the 512. First, don't forget that all PC software issue discs, regardless of the source, will be formatted as (usually) 360K or just possibly sometimes as 720K. Either of these is fine, the 512 can handle both without trouble, but do not expect (or ask for) disks in either 640K or 800K format. Both of these are peculiar to the 512 and PC software suppliers won't know what you're talking about if you mention them.

Finally, whether you write or phone for PC software, don't ask if a particular program will work in the 512. Most suppliers won't even have heard of the 512 and those that have will have no idea of the answer to your question. BEEBUG Vol.8 No.1 contains some guidelines which should be considered when you choose PC software.

PIPING

Finally I'll round off with a mention of piping as promised last month. Piping is a facility which originates in UNIX, but which has been adopted by MS and PC-DOS to extend their capabilities.

I'm explaining piping not because it's likely to be very useful in the 512 - after all, most of you will never have come across the operation before and won't have missed it, but rather because you may find references to it when you're looking at PC software or manuals, so it might be helpful if you know what it means.

The reason piping isn't used in the 512 is because the operating system is DOS Plus, and as I've said many times before, DOS Plus provides numerous capabilities missing from both MS and PC-DOS. The easiest way to explain what I mean is with a simple example.

Suppose you 'TYPE' a file to the screen. One of the things you want it to do is stop at every screenful so that you can read it. In the 512 we can do this in one of three ways. Firstly, you can press and hold Shift and Ctrl, just as you can in BBC native mode. So long as the keys are held the screen won't scroll. Secondly you can press Ctrl-S to stop the scroll, followed by Ctrl-Q to resume it. Both of these methods have the disadvantage that you might miss part of the information if your timing is a bit off. Better and completely reliable is to issue the DOS Plus *pause* command suffix, so to type a file called 'TEXT.DOC' you'd enter:

```
TYPE TEXT.DOC /P
```

and the display will pause at every screen full until you press a key when it continues to the next 'page', all very simple and pretty elementary I'm sure you'll agree.

Well it's not so elementary after all, because MS-DOS and PC-DOS can't do it (and no, that's not what the 'scroll-lock' key on PCs is for either). To achieve this effect in a PC running MS or PC-DOS you need to call into service another program. This extra program, which is called 'MORE', simply passes the text through to the screen transparently, while counting the lines. After counting 24 lines, it's this extra program that pauses the display to give paged mode, not the operating system.

The method employed to do this is not the obvious one. You might expect 'MORE' to do the whole job, in which case the command might be:

```
MORE TEXT.DOC
```

but then the program could be more sensibly called 'TYPE', couldn't it? The problem then would be that screen paging would no longer be optional, so a different approach was needed and this is where piping comes in. Using an essentially similar idea to that used in redirection it allows you to alter the processing of default input or output. In this case, rather than being re-routed to or from, say, a file or a communications port, the data goes through a named program on its way to its standard destination. The character used to implement this is the vertical bar '|', and the way our 'TYPE' command is issued in a PC if you want paged output mode is:

```
TYPE TEXT.DOC |MORE
```

It amounts to the exactly same thing as using '/P' does in DOS Plus, but it makes command entry longer and it involves loading yet another file (MORE.COM) from disc, so it's by no means so convenient to use. There are in fairness a few circumstances where piping can be used to advantage, but to put things in perspective, how many times have you thought such a facility would be the perfect solution to a problem? I thought not.

As I suggested above, piping is generally neither needed in, nor missing from, the 512, but at least you now know what is meant if you come across the term 'piping', or a reference to 'DOS Pipes'. Both names mean the same and refer to inserting a program between the normal source and destination of standard input or output data streams.

That's all for another month, here's Abstract's address:

Abstract
4A, Silchester Road, Pamber Heath,
Basingstoke, Hants RG26 6EA.
Tel. (0734) 700319

BEEBUG Education

by Mark K. Sealey

INTRODUCTION

Software for the BBC micro to support specific Attainment Targets (ATs) of the National Curriculum has now begun to appear apace. This month Beebug Education examines two linked titles from established publishers Sherston that do just that - but which are also well worth a look in their own right.

Products	**Animated Numbers** **Connections**
Supplier	**Sherston Software,** **Swan Barton, Sherston,** **Malmesbury,** **Wiltshire SN16 0LH.**
Prices	**£25.00 plus VAT (Animated Numbers)** **£17.50 plus VAT (Connections)**

These two products have indeed got much in common: chiefly that they are designed specifically to support the mathematics National Curriculum at Levels 1 and 2. This will also mean that teachers working with FE Access-type and basic numeracy classes could also consider using them.

Other features in common are: their ease of use, imaginative presentation on screen, flexibility and an excellent set of documentation and supporting ideas. Teachers will be pleased to hear, as well, that both packages allow a large measure of teacher control in setting parameters of difficulty, graphics and the like.

These two programs should certainly be near the top of the list for anyone who is keen to present mathematical material in a lively if not entirely open-ended way. The maths topics covered by ATs 2, 3 and 4 (*Animated Numbers*), and ATs 2, 3, 5, 8 and 13 (*Connections*) can be practised with imaginative and at times quite sophisticated graphics boasting a good deal of pupil-machine interaction. This has the additional benefit (especially for infants) that few reading skills are required to understand what any given task will require of them.

CONNECTIONS

Turning first to the more complex of the two products, *Connections*, here is a series of simple ideas well executed. There are three 40 track DFS discs. Only the start-up disc is copy-protected. On inserting this and booting, a screen appears which allows the teacher to set the sound level, turn the animation on or off, control the length of game (in terms of the number of screens) and change the specific maths activities.

This latter option is perhaps the most important: four levels of difficulty are catered for plus a section which prompts for selection of topic by National Curriculum AT only.... 'AT2, place value' or 'AT 3, money' etc.

If this is the way you work - instead, perhaps, of a more child or theme centred way - then things are made very easy. Even if it is not, or you want to provide reinforcement activities for children working on a specific maths concept, then most of the "games" that form the substance of the pack go on to present problems and tasks to the pupils in an appealing, consistent and appropriate way.

To get back to the actual 'games', after making teacher selection, press Tab and you are returned to the start screen. You can then insert either the work or the print disc. More of the latter later.

At this point the significance of the connections principle becomes clear. All the children have to do is make true the pairs or connections in whatever context they meet them.

For example, a number activity may present them with two rectangles; one contains a 5p and a 2p coin. The other has the figure 10. By using the cursor keys until either two 5p coins, or a 5p, two 2p and then an additional 1p coin are obtained, the connection is made.

There are no boxes to be filled in. It is the idea of equivalence that rules supreme. Alternatively, the child may choose to decrease the 10 in the other box until 7 appears; this is also equal to the original 5p and 2p coins.

At times, this equivalence is achieved by moving an indicator up and down a linear scale, actually increasing or decreasing the number of physical objects drawn (and well drawn at that) in its own area on the screen until, say, there are five discrete trains in one of the two boxes and the figure 5 or another five objects in the other. Maybe all that differs is the arrangement: 2+3/1+1+1+2/4+1/1+4 etc. There are many other parallel ways of presenting such connections in the pack.

From the point of view of generally accepted child development, this is all very sound stuff. Such essential concepts as one-to-one correspondence, seriation and notions of cardinal numbers are all explored in a vivid and often amusing way.

If skilfully used by the teacher, it is possible to lead the pupils through a carefully thought out progression of concepts: e.g. comparison of number sets through arithmetic operations and knowledge of number facts to solving more general 'problems' as such.

Yet all the time, the knowledge of the software (and the Concept Keyboard can be used as well) relies on consistent use of keys. Delete will ALWAYS decrease the number of objects on screen as described just now. Space ALWAYS adds one. Return ALWAYS means "equals". This is a real advantage to young children - however quickly they grow used to the computer.

There is much variety, too, in the different styles of pattern in which tasks are presented. The program usually selects the configuration of the links (four centrally, a sequence of add/take boxes etc.) according to the activity. Even to the uninitiated, these largely explain themselves!

It is possible, however, to decide any of this for yourself when it comes to printing out the completed screens on Epson or Integrex printers - for use later (or beforehand?) as worksheets. Once again, the documentation is clear and concise on all aspects of this part of the software.

Connections, then, contains a series of activities designed to operate at the child's level in a planned and sequential way. The sequence - if followed properly - provides puzzles and tasks that encourage a measure of what the documentation calls "over-learning".

In other words, it works on one of the soundest of all Piagetian principles: that to really understand what a pupil is doing in maths, it is necessary to approach a new concept from several different directions, and learn it in a variety of ways before moving on. This is doubly necessary because of the extent to which the National Curriculum crams material in, and also because children - while learning maths sequentially - do not necessarily learn it in a linear fashion.

Any program that allows time for you to go back and do again what has been poorly taught or for which inadequate time for practice had been allowed is a real boon. *Connections* has these qualities. Buy it!

ANIMATED NUMBERS

This software is much simpler in use - there is the teacher control option at the start. Then three games: one, 'Ladders', where the children have to count the number of objects in a set at the top left of the screen and guide a "Numberbod" sprite up and down ladders until it hits the corresponding cardinal number.

'Pictures' requires the children to press the correct number to start a relevant numerical animation while 'Links' works with simple set-matching not unlike some of the *Connections* activities.

Animated Numbers is nowhere near as comprehensive as *Connections*. Yet it is not so slight an offering as might at first be thought: Sherston has provided several sound and useful supporting activities and thought has gone, too, into that unhappily neglected area of the relationship between language and children's mathematical development.

In each package, there is some measure of basic technical support (making backups and troubleshooting etc.), and the presentation of both products is stylish and professional.

All in all, these are two well-planned aids for the busy teacher who wishes to be sure that his/her pupils are engaged on a computer activity (once they have had the vital experience with real objects - conkers, counters and the like) which is more than mere time filling. Well worth a look.

Word Processor Input (Part 2)

by Andrew Rowland

If you are using last month's program, WPinput, you are no doubt happy with its word processor like facilities for normal string input, but if you have sideways RAM, it is much better to use this and avoid using main memory for this extension of the operating system. Later in this article I will also deal with more advanced uses of the routine and editing a default string. However, we will start with the changes required to make WPinput a ROM image, and to avail ourselves of the spare space in a bank of sideways RAM to improve the recall facility.

SIDEWAYS RAM VERSION

If you have the monthly disc, just run the program *ROMsrce*. Otherwise, you will need the program Osword0 from last month (with the line numbers unaltered).

Listing 1 shows how to carry out the necessary changes. The easiest way to do this is to type all the instructions given into a word processor such as Wordwise, View, or the Master's Edit, and save the text in the normal way with the name *Changes*. You may omit any comment lines beginning '*|', and when you reach line 5000, you can enter AUTO 5000 and type the remainder without line numbers. With this saved as a text file, return to Basic and type:

```
EXEC Changes
```

ignoring any "Bad Command" messages that appear while the file is EXECed. The result is the source code for the ROM image, which should be saved as *ROMsrce* before running. The resulting ROM image may be saved to disc as *InputROM* or installed in bank 7 straight away - users with non-Acorn RAM boards will need to adapt line 4380 to suit their machines.

The ROM image version effectively frees main memory except for a short patch of 34 bytes which is copied down to &380 when the ROM initialises. The function of this patch will be explained later. The area of memory from &380 to &3DF is only used by the cassette filing system, so should usually be safe (BEEBUG's Master ROM does use another part of this area). However, tape should not be used with InputROM installed.

IMPROVED RECALL FACILITY

The extra space in a 16K sideways RAM bank allows us to store up to 57 strings. When you press Ctrl-R, you recall the last string entered (unless it's the first time that session, when you get the default string from line 4320). Press Ctrl-R again and the string entered before that is recalled, and so on. When it runs out of strings, it goes back to the most recent. With each new string entered, the carousel gets bigger until the maximum size (57 strings) is reached, at which point each new string starts overwriting the previous oldest one.

If your sideways RAM is 8K, alter *maxheapsize* to 25, if 12K to 41 (line 110).

GETTING THE MOST FROM WPINPUT

The OS routine that WPinput and InputROM replace is one of a class of calls known as Osword, with a call address of &FFF1, which are used by Basic and the OS. On entry, the accumulator contains zero to distinguish it from Osword's other functions. The X and Y registers point to a five-byte parameter block (see Table 1).

XY+ 0	buffer LSB
+ 1	buffer MSB
+ 2	maximum string length
+ 3	minimum ASCII value acceptable
+ 4	maximum ASCII value acceptable

Table 1 - Osword 0 parameter block

The calling program specifies in this block the address of the area of memory - or buffer - where the string is to be stored, the length it may be permitted to reach and the maximum and minimum ASCII values that should be accepted.

Normally, this is done automatically by whichever program you are using. If it is in response to Basic's INPUT instruction, for example, the user is limited to a string with up to 238 characters from ASCII 32 to ASCII 255. But if you need to be more specific about your requirements, it is possible to use Osword 0 directly. For example, if your program needs numeric input, you can restrict the permissible keypresses to those between "0" and "9"; or if a filename, to exclude any punctuation which is illegal in filenames and keep the length to 7 characters.

It is strange that the standard OS routine cannot actually be used in this way. Regardless of whether a keypress is acceptable or not, the corresponding character is still displayed on the screen (whether or not it is actually added to the buffer). If the keys have been restricted as above, the screen can soon show something quite different from what is in the buffer, without the user being aware of it.

The reason for this is that some people use the Ctrl key to enter certain codes, for example, Ctrl-L to clear screen. These are known as control codes and should be sent to the VDU driver but not added to the string in the buffer.

The approach taken by WPinput is much more sensible. Control codes are sent to the VDU driver but cannot be added to the string. Apart from that, only acceptable keypresses are shown: anything else is greeted with a warning beep. By the way, if you specifically *don't* want control codes to be acted upon - which may be preferable for use by some programs, especially educational ones, though not for programming - alter line 2040 to:

```
JMP peep.
```

So it is now possible to use WPinput (or InputROM) in your own programs to vet input, and save you the trouble of writing special input routines. All that is required is the additional function FNinput (see Listing 2). This listing contains an example limiting input to strings consisting of up to 10 capital letters. It uses page 9 of memory as the buffer. By adapting this example program, you can specify the form your own string input should take. Remember, however, that if numeric input is required, you may need to allow for plus or minus signs, the decimal point and even ^. A useful compromise is suggested in Listing 3.

EDITING A DEFAULT STRING

It is often desirable to present a given string, such as a previously used filename, to be accepted as it is or to be edited. This is possible with WPinput. If you look at line 1290 in last month's listing, you will see that the first thing WPinput does is to store the contents of the accumulator (A) in *length*, the current string length. Of course, in the normal run of things, the accumulator contains zero on entry, and the string starts with a length of zero. However, if we arrange things so that the buffer contains the default string, A contains its length and we call *transfer* directly (line 1280), we can then edit the existing buffer contents. This is accomplished by FNinput2 (see Listing 3).

Such simplicity is not quite possible for the ROM version. Instead, we will create a new Osword call with A=16 on entry. The parameter block will be one byte longer, the extra byte containing the length of the default string. Some extra code (lines 1265, 1272-1276) places this length byte in the accumulator and calls *transfer*. Listing 3 should be modified as follows:

```
1070 A%=16:CALL &FFF1
```

TECHNICAL NOTE

Machine code programmers wishing to use WPinput will find relevant information in the manuals under Osword 0. However, the manuals are incorrect in two things, at least for the Master. The Carry flag *is* set when input is terminated by Escape, but A always contains zero on exit (which Basic relies on), and Y contains the length of the string (*not* including the Carriage Return).

HOW IT ALL WORKS

WPinput carries out the following actions:

1. Reads the parameter block, calculates the size of the current text window. Notes the cursor position - this is used as the string base co-ordinates.
2. Prints the string in the buffer at base co-ordinates.
3. Positions the cursor or cursors.
4. Accepts a keypress and acts accordingly.
5. Loops back to 2 until Return or Escape is pressed.
6. Prints string for last time to sort out cursors, and exits.

The fact that the entire string is reprinted on each keypress can make it feel sluggish when the string gets very long, but is acceptable at normal typing speeds. However, it slows down EXEC files too much, so a check for this is made at line 1300 and the normal OS routine is then called instead. The repeated printing of the string would cause havoc with SPOOL and the printer, so both are temporarily suspended until just before the final print (stage 6 above).

It is necessary for two spaces to be printed after the string, one to delete the final character (if you pressed Delete and made the string shorter), and one to remove the blob cursor if it was after the last character. You will see those spaces in SPOOL files.

The base co-ordinates are never changed unless scrolling takes place or you press Ctrl-L to clear the screen - this is specifically checked for (line 1750). You should not alter the size of the text window using the Ctrl key, and it is the programmer's responsibility to ensure the text window is large enough for the maximum length of string permitted, including the two extra spaces just mentioned. The routine cannot display just part of a string.

Mention was made earlier of a machine code patch for the ROM version. With well behaved ROMs like Basic and View, which build their parameter blocks in main memory, there is no problem. But certain ROMs, like BEEBUG's Master ROM and the Master's Edit, have parameter blocks 'hard wired' into the ROM itself. When InputROM tries to read these blocks, it finds them inaccessible from within another ROM, hence the need for a patch in main memory to make a copy of such blocks in a convenient location - I use the bottom of page 1 as a relatively safe area.

A final technical point: the routine takes the liberty, as does the OS itself, of addressing locations directly rather than using legal calls. In fact, this is necessary to keep the code compact. So far, all the locations concerned have been in the same place on all machines, but if you have a co-processor, WPinput must run in the I/O processor.

Next month, we conclude this series with a function key editor for the Master and a Basic line editor.

Listing 1

```
*| >Changes
*BASIC
LISTO 1
LOAD "OSWORD0"
   10 REM Program .>ROMsrce
   30 REM for sideways RAM
  100 LOMEM=TOP+&700
  110 mc%=TOP:maxheapsize=57
  230 FOR pass=4 TO 6 STEP 2
  240 O%=mc%:P%=&8000:PROCromhdr
DELETE 1020,1150
 1182 .ptr     EQUB 1
 1183 .gotto   EQUB 0
 1184 .count   EQUB 0
 1185 .heapsize EQUB 1
 1187 .lengths EQUB 23
 1188 EQUS STRI.maxheapsize,CHR$0)
 1265 CMP #16:BEQ newOSWORD
 1272 .newOSWORD
 1274 STX temp:STY temp+1
 1276 LDY #5:LDA (temp),Y:LDY temp+1
 1278 :
*| changes to keep
 2080 .keep CPY #0:BEQ exit:TYA
 2082 LDX ptr:STA lengths,X
 2084 STX gotto:TXA
 2086 CLC:ADC#store DIV &100:STA temp+1
 2088 LDA #store MOD &100:STA temp
 2100 LDA (buffer),Y:STA (temp),Y
 2120 LDA (buffer),Y:STA (temp),Y
 2121 INC ptr:LDA ptr
 2122 CMP #maxheapsize:BNE kpover
 2123 LDA #0:STA ptr
 2124 .kpover LDA heapsize
 2125 CMP #maxheapsize:BEQ exit
 2126 INC heapsize
*| changes to recall
 3850 .recall LDX heapsize:BEQ rexit
 3852 LDX gotto:LDA lengths,X:STA count
 3854 TXA:CLC:ADC #store DIV &100
 3856 STA rloop+2:LDA #store MOD &100
 3858 STA rloop+1
 3930 .rlov INX:CPX count:BNE rloop
 3942 DEC gotto:BPL rexit
 3944 LDX heapsize:DEX:STX gotto
 4310 .store
 *| changes to SAVE
 4380 IFM$="I" OS. "SRWRITE "+STR$~mc%+"
"+STR$~O%+" 8000 Z":P."Press CTRL/BREAK"
 4390 IFM$="S" a$="SAVE InputROM
"+STR$~mc%+" "+STR$~O%+" 8000 8000"
 5000 :
 5010 DEF PROCromhdr
 5020 [OPT pass
```

```
5030 .romstart
5040 EQUW 0:EQUB 0
5050 JMP serventry
5060 EQUB &82:EQUB copyrt-romstart
5070 .title
5080 EQUB 0:EQUS "OSWORD 0 extention"
5090 .copyrt EQUB 0
5100 EQUS "(C) BEEBUG 1990":EQUB 0
5110 .serventry
5120 CMP #9:BEQ help
5130 \ could use &27 in next line
5140 \ in Master only
5150 CMP #&FE:BEQ initialise
5160 RTS
5170 :
5180 .help
5190 PHA:LDA (&F2),Y:CMP #13:BNE hlpout
5200 TXA:PHA
5210 JSR osnewl:LDX #1
5220 .hlploop
5230 LDA title,X:BNE hlpover
5240 LDA #32
5250 .hlpover JSR oswrch
5260 INX:CPX #serventry-title-1
5270 BNE hlploop:JSR osnewl
5280 PLA:TAX
5290 .hlpout PLA:RTS
5300 :
5310 .initialise\ alter vectors
5320 PHA:TXA:PHA:TYA:PHA
5330 LDX #0 \ copy patch to main memory
5340 .loop LDA loc%,X:STA patch,X
5350 INX:CMP #&FF \ last byte is &FF
5360 BNE loop
5370 \ alter vectors
5380 LDA &20C:STA oldv
5390 LDA &20D:STA oldv+1
5400 \ get extended vector table address
5410 LDX #0:LDY #&FF
5420 LDA #&A8:JSR osbyte
5430 STX temp:STY temp+1
5440 LDY #3*6:SEI
5450 LDA #entry MOD &100:STA (temp),Y
5460 INY
5470 LDA #entry DIV &100:STA (temp),Y
5480 INY
5490 LDA &F4:STA (temp),Y \ ROM no.
5500 LDA #3*6:STA patchv
5510 LDA #&FF:STA patchv+1
5520 LDA #patch MOD &100:STA &20C
5530 LDA #patch DIV &100:STA &20D
5540 CLI:PLA:TAY:PLA:TAX:PLA
5550 RTS
5560 .loc%
5570 ]LOCAL temp,block
5580 patch=&380:REM only if TAPE not used
5590 temp=&E8:block=&100:P%=patch
5600 [OPT pass
5610 \ vectored entry point
5620 CMP #0 :BEQ dopatch
5630 CMP #16:BEQ dopatch
5640 JMP (patchv) \ not OSWORD 0
5650 .dopatch
5660 PHA:STX temp:STY temp+1
5670 LDY #5 \ transfer param block
5680 .translp
5690 LDA (temp),Y:STA block,Y
5700 DEY:BPL translp
5710 LDX #block MOD &100
5720 LDY #block DIV &100
5730 PLA:EQUB &4C \ JMP
5740 .patchv EQUW &FF12
5750 ]P%=loc%+P%-patch
5760 ENDPROC
```

Listing 2

```
  10 REM .>EXAMPLE1
  20 :
 100 DIM block% 5
 110 PRINT"Enter string: ";
 120 A$=FNinput(ASC"A",ASC"Z",10)
 130 PRINTA$
 140 END
 150 :
1000 DEFFNinput(min,max,L%)
1010 !block%=&900
1020 block%?2=L%
1030 block%?3=min
1040 block%?4=max
1050 A%=0:X%=block%:Y%=X% DIV &100
1060 CALL &FFF1:=$&900
```

Listing 3

```
  10 REM .>EXAMPLE2
  20 :
 100 DIM block% 6
 110 num=356
 120 PRINT"Enter number: ";
 130 num=VAL(FNinput2(STR$(num),ASC"+",
ASC"9",5))
 140 PRINT num:END
 150 :
1000 DEFFNinput2(A$,min,max,L%)
1010 $&900=A$:!block%=&900
1020 block%?2=L%
1030 block%?3=min
1040 block%?4=max
1050 block%?5=LEN(A$)
1060 X%=block%:Y%=X% DIV &100
1070 A%=block%?5: CALL &134A
1080 REM 1070 A%=16:CALL &FFF1
1090 =$&900
```

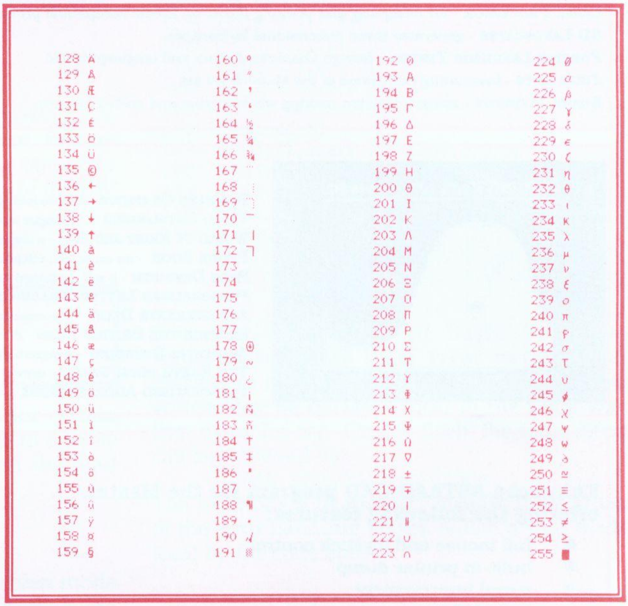
Redefining Printer Characters 128-255

We present an improved set of printer character definitions for characters 128 to 255 supplied by Dr J.R.Barker.

In BEEBUG Vol.8 No.9 Lance Vick introduced his program for redefining, on any Epson compatible printer with a down-loadable character facility, characters 128 to 255 to correspond to those displayed on screen by the Master 128 for the same ASCII character range. Because the basic matrix used by screen characters differs from that used by printer characters, the results, while very acceptable, cannot produce as good a match as some might like.

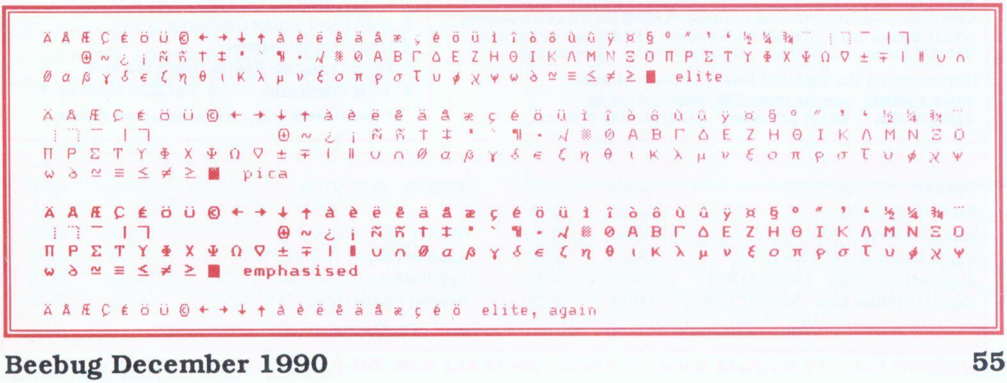
To overcome this problem, all the relevant characters have been hand-coded as a set of DATA statements, as far as this is possible, and the results correspond much more closely when printed out with those displayed on screen by the Master 128 (see illustrations).

Because of the length of the program, and the likelihood of mistakes being introduced if it were to be typed in from a listing, the program is included only on the magazine disc for this issue of BEEBUG. However, for those who wish, a photocopy of the listing can be supplied on receipt of an A5 stamped addressed envelope (sent to BEEBUG's Editor and clearly marked "Font Program".



```
128 Ä     160 °     192 Ø     224 ø
129 Å     161 ″     193 A     225 α
130 Æ     162 ’     194 B     226 β
131 Ç     163 ‘     195 Γ     227 γ
132 É     164 ½     196 Δ     228 δ
133 Ö     165 ¼     197 E     229 ε
134 Ü     166 ¾     198 Z     230 ζ
135 ©     167 ⋯     199 H     231 η
136 ←     168 ⋮     200 Θ     232 θ
137 →     169 ⌜     201 I     233 ι
138 ↓     170 ‾     202 K     234 κ
139 ↑     171 |     203 Λ     235 λ
140 à     172 ┐     204 M     236 μ
141 è     173       205 N     237 ν
142 ë     174       206 Ξ     238 ξ
143 ê     175       207 O     239 ο
144 ä     176       208 Π     240 π
145 â     177       209 P     241 ρ
146 æ     178 ⊕     210 Σ     242 σ
147 ç     179 ∼     211 T     243 τ
148 é     180 ¿     212 Y     244 υ
149 ö     181 ¡     213 Φ     245 φ
150 ü     182 Ñ     214 X     246 χ
151 ì     183 ñ     215 Ψ     247 ψ
152 î     184 †     216 Ω     248 ω
153 ò     185 ‡     217 ∇     249 ∂
154 ô     186 ▪     218 ±     250 ≅
155 ù     187 `     219 ∓     251 ≡
156 û     188 ¶     220 |     252 ≤
157 ÿ     189 ·     221 ‖     253 ≠
158 ¤     190 √     222 ∪     254 ≥
159 §     191 ▒     223 ∩     255 █
```

The alternative Printer character definitions can equally be used on a model B if combined with Lance Vick's original program which also defined the screen characters for that machine.



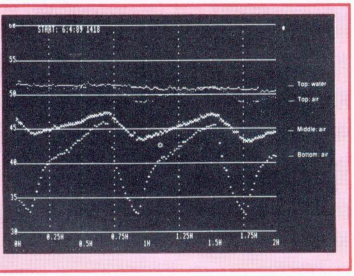
```
Ä Å Æ Ç É Ö Ü © ← → ↓ ↑ à è ë ê ä â æ ç é ö ü ì î ò ô ù û ÿ ¤ § ° ″ ’ ‘ ½ ¼ ¾ ⋯ ⋮ ⌜ ‾ | ┐
          ⊕ ∼ ¿ ¡ Ñ ñ † ‡ ▪ ` ¶ · √ ▒ Ø A B Γ Δ E Z H Θ I K Λ M N Ξ O Π P Σ T Y Φ X Ψ Ω ∇ ± ∓ | ‖ ∪ ∩
ø α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω ∂ ≅ ≡ ≤ ≠ ≥ █ elite

Ä Å Æ Ç É Ö Ü © ← → ↓ ↑ à è ë ê ä â æ ç é ö ü ì î ò ô ù û ÿ ¤ § ° ″ ’ ‘ ½ ¼ ¾ ⋯
⋮ ⌜ ‾ | ┐           ⊕ ∼ ¿ ¡ Ñ ñ † ‡ ▪ ` ¶ · √ ▒ Ø A B Γ Δ E Z H Θ I K Λ M N Ξ O
Π P Σ T Y Φ X Ψ Ω ∇ ± ∓ | ‖ ∪ ∩ ø α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ
ω ∂ ≅ ≡ ≤ ≠ ≥ █ pica

Ä Å Æ Ç É Ö Ü © ← → ↓ ↑ à è ë ê ä â æ ç é ö ü ì î ò ô ù û ÿ ¤ § ° ″ ’ ‘ ½ ¼ ¾ ⋯
⋮ ⌜ ‾ | ┐           ⊕ ∼ ¿ ¡ Ñ ñ † ‡ ▪ ` ¶ · √ ▒ Ø A B Γ Δ E Z H Θ I K Λ M N Ξ O
Π P Σ T Y Φ X Ψ Ω ∇ ± ∓ | ‖ ∪ ∩ ø α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ
ω ∂ ≅ ≡ ≤ ≠ ≥ █ emphasised

Ä Å Æ Ç É Ö Ü © ← → ↓ ↑ à è ë ê ä â æ ç é ö elite, again
```

Special Christmas Offers!

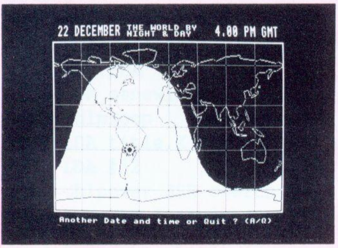
on Best of BEEBUG software

Applications II Disc

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



Applications I Disc



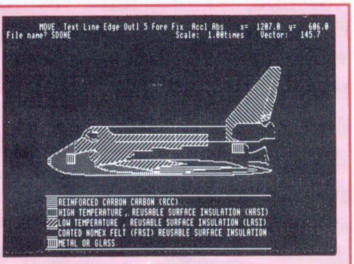
```
22 DECEMBER THE WORLD BY NIGHT & DAY    4.00 PM GMT

Another Date and time or Quit ? (A/Q)
```

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

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



```
       MOVE  Text Line Edge Outl 5 Fore Fix  Accl Abs    x=  1207.0  y=   606.0
File name? SDONE                          Scale:  1.00times    Vector:   145.7
```

REINFORCED CARBON CARBON (RCC)
HIGH TEMPERATURE , REUSABLE SURFACE INSULATION (HRSI)
LOW TEMPERATURE , REUSABLE SURFACE INSULATION (LRSI)
COATED NOMEX FELT (FRSI) REUSABLE SURFACE INSULATION
METAL OR GLASS

EDIKIT ROM

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

***FTEXT** (find text) ***FBASIC** (find Basic) ***FPROCFN** (find procedure/function)
***LPROC** (list procedure) ***LFN** (list function) ***LFROM** (list 8 lines of a program)
***RTEXT** (replace text) ***RBASIC** (replace Basic) ***SYSINF** (system information)
***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**

General Utilities Disc

- ✷ Printer Buffer *
- ✷ 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 †

*** Master series only. † Requires sideways RAM.**

	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 - 60p for the first item and 30p for every additional item.

BEEBUG Ltd, 117 Hatfield Road, St Albans, Herts AL1 4JS. Tel. (0727) 40303 Fax. (0727) 860268

HINTS *and tips* HINTS *and tips* HINTS *and tips* HINTS *and tips* HINTS *and tips*

Our thanks to Andrew Rowland in particular for this month's hints. Hints and tips on almost any subject relating to the BBC micro and Master series are always welcome, and we pay £5 for each one published.

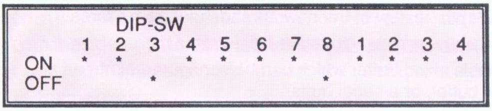
STAR PRINTER DIP SWITCHES

Andrew Rowland

The Star LC-10 (and some other Star printers) has an undocumented Escape code sequence, Escape-0 (figure 'zero') which prints (in the currently selected print style) the current settings of the DIP switches. For example, from Basic type:

```
    VDU2,1,27,1,0,3
```

to obtain a print out like this:



```
          DIP-SW
        1  2  3  4  5  6  7  8  1  2  3  4
ON      *  *     *  *  *  *     *  *  *  *
OFF           *                 *
```

If you need to change DIP switches for certain programs, why not print the settings in double or even quad sized letters to pin on the wall over the printer?

NEW SCREEN MODE

by Andrew Rowland

This program demonstrates a new screen mode which provides 40 columns by 25 lines of text in four colours, and takes 4.4K less than a mode 1 screen. It is most useful in situations where user defined characters and maximum readability on colour monitors are required, without sacrificing the use of colour.

The program starts in mode 1 and alters the vertical timing registers to create a gap between each line. As the mode is not recognised by the OS, lines 1080 to 1090 define text and graphic windows to ensure scrolling takes place properly and that the freed memory is not corrupted.

```
  10 REM Program Mode1A
  20 REM Author  Andrew Rowland
  30 :
 100 MODE 1:PROCmode1A
 110 IF HIMEM<>&8000 THEN HIMEM=start
 120 COLOUR 0:COLOUR 131
 130 PRINTTAB(12,10)"40x25 lines text"
 140 COLOUR 128
 150 COLOUR 1:PRINTTAB(7,12)"in four";
 160 COLOUR 3:PRINT" glorious";
 170 COLOUR 2:PRINT" colours"'
 180 MOVE 232,380:DRAW 1000,380
 190 END
 200 :
1000 DEF PROCmode1A
1010 VDU23;9,9;0;0;0;
1020 VDU23;6,25;0;0;0;
1030 VDU23;5,2;0;0;0;
1040 VDU23;4,30;0;0;0;
1050 VDU23;7,27;0;0;0;
1060 start=&4180
1070 VDU23;12,start DIV 2048;0;0;0;
1080 VDU23;13,start MOD 2048 DIV 8;0;0;0;
1090 VDU28,0,31,39,7
1100 VDU24,0;0;1279;796;
1110 ENDPROC
```

ON ERROR LOCAL

by Andrew Rowland

One of the features of Basic V on the Archimedes is its local error handling. A procedure may have its own ON ERROR line which may be declared local in the same way that variables can. On the Beeb, the same effect can be achieved by:

```
        LOCAL ?&16,?&17:ON ERROR ....
```

The locations &16 and &17 contain the address of the error handling vector, which is thus made local by the code above.

SHIFTING CASE

Mike Trace & P.Vincent

OSBYTE 202 (or *FX202) can be used to control the settings of the keyboard for subsequent input as shown in table 1. The value is the argument specified in the OSBYTE call or following *FX202.

Value	Caps Lock	Shift Lock	Shift Enable
0	On	On	No
16	Off	On	No
32	On	Off	No
48	Off	Off	No
64	On	On	Yes
128	On	On	Yes
144	Off	On	Yes
160	On	Off	Yes

If you try any of these then you should see the keyboard indicator lights change accordingly. This technique can be very useful for ensuring that user keyboard entry is correct and in the format expected by the program.

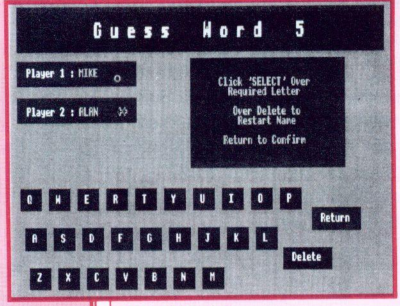
RISC USER

The Archimedes Magazine & Support Group

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

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

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

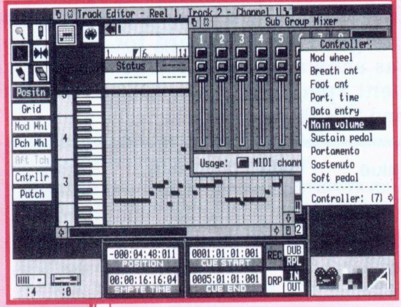


Click 'SELECT' Over Required Letter

Over Delete to Restart Name

Return to Confirm

GuessWord 5



Inspiration

GUESSWORD 5

A special treat for Christmas - an entertaining game which is a five-letter variant of the well known board game Mastermind.

SCHEMA

An in-depth review of the new spreadsheet from Clares.

PRINTER AND SERIAL BUFFER FOR THE ARC

A variable sized buffer which can be configured either as a printer buffer, or a serial buffer.

INSPIRATION - MIDI SEQUENCER

A review of the Pandora's long awaited professional MIDI sequencer.

WIMPAID

A very useful application for Wimp programmers allowing window parameters to be dynamically displayed and updated on screen.

ACORN'S NEW ARCHIMEDES FLAGSHIP: THE A540

A look into this fascinating new machine from Acorn.

ASSEMBLER WORKSHOP

A major series for the more advanced ARM processor programmer. The latest one looks into heap management.

MASTERING THE WIMP

A major series on the Wimp programming environment. The most recent installment explains about Implementing a Save Box.

INTRODUCING C

A wide ranging series on the major programming language C.

INTO THE ARC

A regular series for beginners. The latest article explores some aspects of using colour on the Archimedes .

WP/DTP

A new column which offers hints on using different DTP and WP packages.

ARCADE

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

As a member of BEEBUG you may extend your subscription to include RISC User for only **£8.10** (overseas see table).

Phone your instructions in, or send a 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.

SUBSCRIPTION DETAILS

Destination	Additional Cost
UK,BFPO &Ch Is	£ 8.10
Rest of Europe and Eire	£12.00
Middle East	£14.00
Americas and Africa	£15.00
Elsewhere	£17.00

RISC User, 117 Hatfield Road, St Albans, Herts AL1 4JS, Telephone (0727) 40303, FAX (0727) 860263

BEEBUG 1990 A.I.

POSTBAG

BEEBUG 1990 A.I.

POSTBAG

VIEW TEXT TO ASCII

I use View on my BBC Master 128 to write short articles which are usually required in two column format on A4 paper, to be printed by my FX-compatible Citizen 9-pin LSP-10 printer. The MULTCOL program from BEEBUG Vol.7 No.1 seemed a good alternative to splitting the article in two, saving the two halves and using RULER to reformat them into left and right half separately.

However, author Jan Stuurman says "first strip out any View formatting commands". That is easy enough for the ruler and top left-hand side commands, but what about other embedded commands and the invisible Returns at the end of each line? These remain in the text, and ruin any re-formatting.

The only way I have found is to use the Master's Edit program, globally replacing |M (the Return character), but then all new paragraphs disappear, making use of the program an unprofitable exercise.

D.L.Smith

There are two complementary answers to this problem. First of all, a simpler way (possibly) to save the text file in the first instance, is to set markers at the start and end of the text proper, and then use the WRITE command to save this to a file. In this way, the embedded commands and rulers which often appear at the head of a View file are easily removed. For example:

```
    WRITE MyText 1 2
```

On the whole, other embedded commands and rulers are best removed from within View before saving the file.

The Master's Edit is indeed a powerful and useful utility, and the problem with removing the single Return characters at the end of each line while leaving the double Return at the end of a paragraph intact can be solved quite easily. After loading the text into Edit, search and replace every occurrence of a double Return ($$ in Edit) with some other control character, say |J. Then replace every remaining (single) Return character with a space. Finally, restore the |J character to a double Return ($$) as before. The text will then be in the format required by the MULTCOL program.

The idea of temporarily changing a character sequence to assist editing can be quite useful, and I have used the same technique to preserve Return followed (on the next line) by Tab (|I), or Return followed by space, again situations where you do not want the Return character to be lost. Although this may sound tedious, with practice it takes very little time at all, and proves a highly effective way of converting a text file from one format to another.

LIMITATIONS OF INTERWORD

Following the recent survey of Word Processors (I have InterWord and wouldn't use anything else), I should like to point out that you should have shadow RAM fitted to your machine otherwise barely 3.5K of memory is available for text. I think the omission of this fact is misleading.

PROBLEMS WITH SCREEN DUMPS

I have found the Petrol Consumption program (BEEBUG Vol.9 Nos.4 & 5) very useful. Having Watford's Shadow RAM installed on my model B, I typed in the Master version which works perfectly, until I get to the screen dump procedure. I discovered that Dumpout3 worked, when Printmaster and the Mode 0 Screen Dump (from BEEBUG Vol.8 No.7) failed completely. In addition, before printing with Dumpout3, I would like to be able to *SSAVE the screen display, but I was further surprised when this also failed. Have any other members any ideas on these problems

*SRWRITE SUBSTITUTE

Finally, how can I emulate the *SRWRITE command (see article on the RFS in BEEBUG Vol.8 Nos.8 & 9) on my Watford SWR board?

T.D.Parsons

The Petrol Consumption program works without problems using the BEEBUG dump on a Master. Any suggestions from readers on this, and the SRWRITE command, will be welcome.

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 15th of each month.

Model B issue 7, 1770 DFS, Acorn ADFS, Microvitec 1441 hi-res. monitor, Pace 5.25" double drive, single sided, 40/80 (power from BBC), similar spare single, Micropulse switchable sideways ROM box, 7 sockets and 1 ZIF socket, Mannesman MT80 tractor/single sheet printer, BBC data recorder and tapes, View, Wordwise Plus II, Spellmaster, Wordease, Masterfile, File-Plus, Ultracalc II, Printmaster, Disc Doctor, box with 30 discs formatted ADFS and utilities for above, all leads and full documentation £450 the lot, can't split, upgrading to Arch. Tel. (0367) 710223.

Master 128 with Cumana CD800S40/80T disc drive and Sony RGB colour monitor and 2 ROM cartridges. Package includes Cumana CS100 disc drive and Acorn data recorder model ALF03. All in mint condition and complete with all manuals and original packing and range of software including Quickcalc, Masterfile, Database, Dumpmaster and Joysticks etc. Unbeatable value £450. Tel. (0753) 887681.

Beeb video digitiser by Watford £65, Wordwise Plus £22, "The Advanced User Guide for the BBC Micro" £8, Astaad CAD Package £9, Masterfile II (ADFS) £10, Speech! £5. Tel. (0582) 410773.

Island Logic Music System, superb, sophisticated software for the model B but not compatible with Master 128. Icons for Editor, Keyboard, Synthesiser, Printer and Linker functions. System, Song & Sound library discs plus detailed and very clear manual. Bargain at £14 to include p&p. Tel. (0903) 784328.

WANTED: Flight simulator for BBC M128, particularly Acorn Aviator. Tel. (0733) 317234 day or (0733) 235505 after 6pm.

Acorn Teletext with ATS £30, Prism modem (75/1200) £15, Silver Reed EXP400 daisywheel £30, Smith Corona D200 9 pin matrix £40, View £20, Viewstore £20, BCPL £15, Disc Doctor £10, BEEBUG Help ROM £7, BEEBUG bound vols. 1-8 £6 each, Advanced User Guide £6, Advanced Disc User Guide £6. Tel. (0734) 791436.

Dual 40/80 single sided 5.25" disc drives £100. BEEBUG Toolkit ROM £5, Masterfile DFS 40T £5, both with original packing & manuals, Prism 1000 1200/75 modem £10, Silent Computers metal console for BBC B with shelf for disc drives etc. £20. Acornsoft Forth on the BBC Micro book £3, BBC B basic ROM Guide £3. Tel. (0582) 601013.

WANTED: Working reliable hard disc 30Mb or larger, suitable for dual partioning for use with BBC Master 512. (0904) 761019 eves.

Archimedes A310 base unit with RISC OS, 18 months old immaculate condition, boxed with manuals, welcome disc and first word plus. Offers around £495. Tel. (0222) 705304.

Kaga Taxan KP-810 dot matrix printer (NLQ). Excellent condition - on second ribbon only. With printer lead (Arch.), dust cover and printer stand £95. Collection prefered (Orpington, Kent), carriage extra. Tel. (0689) 57245.

aster & 512 co-processor, mouse and Gem software. Twin Cumana DS/DD drives and green monitor, ROMs include View & Viewsheet, Master, Wordwise Plus and Interword. Magic modem and Commsoft fitted, tape deck and Epson RX80. All cables and good selection of manuals included (as new) £550. Tel. (0844) 52547.

M128, Opus dual disc drive, BBC Teletext Adaptor, Telemod 2 modem, Sanple daisy printer, MasterROM, Overview, Moneywise ROM, Z88 link, all with manuals £395. Tel. (06285) 25519.

M128 in excellent condition, hardly used, geniune bargain price £275, Walkom L-3300 IBM AT compatible portable, 1Mb RAM (expandable to 4) 40Mb fast hard disc drive, 1.44 Mb floppy disc drive, high resolution EGA paper white screen, all ports, hardly used, excellent value. Cost £2200 sell £1500 o.n.o. Tel. 061-275 6768 (day) or 061-445 4638 (eves).

Master 512 with dual 80T drive, cartridges with ROMs, manuals, BBC & PC software £450, Hybrid Technology Music 4000/2000 system including keyboard, synthesizer, MIDI interface, toolkit software and all manuals £180, EMR MIDI interface with sequencer, editing and scoreprinting software for Master computer £120, Acorn Teletext adaptor £15. Tel. (0462) 686818.

Hybrid Music 5000 synth, 4000 keyboard, 2000 MIDI interface, programmers manual, AMPLE toolbox £250, dual 80T drive £85, AMX mouse & Super Art for M128 £30, View Professional £30, Commstar II £15, Termulator M128 £15, Revs £5, Repton II £5, Micron EPROM programmer £10, M128 ROM cartridges £5 each, Master Reference Manuals (pair) £15, Advanced User Guide £5, Acorn Speech system & Easytalk ROM £10. Tel. (0522) 542062.

BBC model B with Opus DDOS, APTL Sidewise ROM board, Toolkit, Wordwise, Pendown and Replay ROMs, Cumana CD800S 40/80T twin double sided disc drive, Microvietc med. res. colour monitor, Canon PW-1080A printer, BBC Acorn joysticks, leads, manuals and a wide selection of cassette and disc software £500. Sensible offers will be considered. Tel. (0622) 726599 anytime.

M128, monochrome monitor, data recorder, modem, several ROMs including Wordwise Plus, leads and manuals £350. Tel. 091-536 2066 (not w/ends).

M128, Microvitec 1441 very high res. monitor, Akhter twin 40/80 drives in bridge, 64k RAM cartridge, Dumpmaster II, Master ROM, Printwise, Viewspell, all manuals (including advanced), books on operating system and assembly language with discs and extensive supporting software £500. Tel. (09274) 23659.

WANTED: Barcode reader and software for youth club. Tel. (0263) 822928.

Morley 2Mb RAMdisc, good working order, complete with ROM software and instruction manual £125. Tel. 081-907 9933.

BBC B issue 7 Acorn ADFS DFS, Watford 32k Shadow RAM, ATPL ROM board with 2x8k S.R. chips, 40T disc drive, Sanyo green mono monitor, Elite, Studio 8 £285. ISO Pascal on ROM cartridge, manuals £25 could deliver in Midlands. Tel. (0384) 377710.

Would like to correspond with Master users to exchange ideas on 64k ROM usage of the Terminal software. Write to; Don De Niet, Wilhelminastraat 15, 9401 NL Assen, Netherlands.

Watford Master plinth monitor stand with 5.25" (40/80) and 3.5" (80) disc drives, both drives are very quiet. The unit has a mains power supply. Price inc. delivery (uk) £140. Tel. 071-494 1365 (office hours only).

ROM podule (CC) £35, Spellmaster ROM (CC for Arch) £35, EPROM Programmer (HCR for BBC) £40. Tel. 04243-4500.

BBC B issue 7, twin Opus 40T SS drives, and mono monitor £245. Also BBC B issue 7 twin Pace 40T SS drives with monitor and ATPL board. £255. Acorn Data cassette recorder £12. Epson FX80 printer with cut sheet feeder £125. p&p extra. Tel. (0768) 64890.

WANTED: Aviator on disc for a Master. Contact; Laurence Butler, 20 Coniston Road, Barnehurst, Bexleyheath, Kent, DA7 6PY.

WANTED: Z88 with or without add-ons. Tel. (0494) 672221.

BBC Master 128, 800k twin drive, Z80 co-processor, monitor, NLQ printer, Viewstore, Interword, excellent condition £450. Tel. (0707) 371733.

512/1024 (Solidisk PC+) co-processor, with all discs books, mouse etc. including Dabhand Master 512 user guide and disc £150 o.n.o. Tel. (0772) 612680.

BBC B issue 3, 1.2 OS, basic 2, (fault on DDFS but otherwise in good condition), inc. manual, workshop manual and software £90, joysticks £10, View Professional £44, View 2.1 £22, Commstar II £16, Spellcheck II £15, Dumpout 3 £15, Diagnostic Disc £12, Elite £8. Tel. 081-789 2663.

BBC B issue 7 1.2 OS with disc interface, cables and disc drive complete. Perfect condition, boxed. Word processor ROM included, quick sale only £180. Tel. (0702) 461220.

BEEBUG C & dual ROM cartridge & Stand Alone Generator £45, Digimouse + Chauffeur & Illustrator software £20. Ferguson 12" green screen monitor £20, Viewspell ROM £15, View & Viewsheet manuals £5 each, both reference manuals £8 each, Advanced Ref. Manual £10, 15 used discs £8. Tel. (0427) 890 750 (eves).

Overview package and 65C102 'Turbo' co-processor for Master £50 each. Master MOS upgrade £25, also Beebugsoft Exmon II and Toolkit Plus £10 each, the following books also; The New Advanced User Guide, Master Reference manuals 1&2 all £10 each, Dabhand Guides to View and Viewsheet/store £8 each (including discs), postage included. Tel. (0788) 521189.

M128 with Torch Z80 sec/processor. Twin switched 40/80T drives. Colour monitor & plinth, Morley AA board, Tape cas/player, numerous ROMs, games, w/p, art, programming tools, Microsoft Basic Interpreter, CPM & BBC Basic manuals £750 will accept sensible offer. Tel. (0932) 783252 (mon-fri 6-7pm).

Story writing, artificial creativity. Postal discussion/programming group proposed. Write to; Tony Kelly, Can y Lloer Ffarmers, Llanwrda, Dyfed.

Pace Linnet V21/23 modem with BBC lead and Modem Master software. Less than 4hours use £75. Tel. (0202) 763271 (anytime).

Master 128 with reference manuals, View manual, Wordwise and Forth ROMs with manuals, dual double sided disc drive 40/80 switchable, Microvietc Cub monitor £350 the lot. Tel. 09274 21326 eves.

Watford viedo digitiser, manuals and utility disc £90. Tel. (0245) 461043.

Philips amber-white monitor BM7522 excellent condition with original packaging £55. Tel. 061-969 9428.

Master 128 with software and manuals £200. Watford 40/80T switchable dual drives together with plinth/bridging unit £125. Tel. (04024) 58772 eves/wkends.

BBC model B with ROM/RAM board fitted, Toolkit, Help, Spreadsheet & Max ROMs fitted, also Zip socket on front. 40/80T disc drive, discs with progs & utilites, all books etc. £300 o.n.o. Tel. (0480) 65696 after 7pm any eve.

WANTED: for BBC B Interword & Spellmaster, also Watford ROM/RAM board, all in as new condition as poss. with complete documentation. Tel. (0661) 71340.

M128 65C102 Turbo (inc 2nd Processor) £70, Morley AA board £25, Universal 2nd Processor £25, Video Digitiser (Watford) £50, all new and unused. WANTED: Interword and/or Megarom 512 co-processor. Tel. (0293) 542288 eves & w/ends.

M128 as new £280, Turbo co-processor £80, Z80 co-processor complete £80, Philips green screen monitor £40, AMX Superart with colourart and mouse £30, Viewplot £12, Viewstore £25, Mega 3 ROM package £50, Datachat 1223 modem with MNET ROM £45, W.E. co-pro adaptor £30, Fleet St. Editor (BBC 40T) £20, Quest mouse £20, W.E. ROM manager £10, Printmaster, Print monitor ROM, Dumpout 3 £7 each, all excellent and complete (upgrading). Tel. 051-647 5367.

BBC B issue 4 with DFS £150, Universal 2nd processor (unused) £20, Watford double BBC plinth £8. Tel. 081-858 6086 eves.

Archimedes 440/1 4Mb RAM, 50Mb hard disc with external 5.25" drive, colour monitor, Tandy wide carriage daisywheel printer and Watford modem. All manuals, software includes 1st Word Plus, Fortran, Hearsay and PC Emulator (including large range of MSDOS software). Just 1yr old and has had little use. Offers in the region of £1950. Also A3000, colour monitor and RAM upgrade, this machine is practically unused and still under warranty, offers in region of £760. Tel. (0223) 872178.

BEEBUG Toolkit plus ROM for BBC B complete with instruction manual £15. Tel. (0246) 274436.

For sale, individually or as a job lot; 14 station Master network: 7 Master 128's with network interfaces £300 each, 7 Master ET computers £250 each, 14 Microvitec/Philips colour monitors £100 each, 14 plinths £5 each, one model B fileserver with 20Mb drive and tape backup £300, two disc drives £50 each, 2 dot matrix printers £50 each, 16 station network wiring clock etc. £50, Most items 1-3 years old. Or £4000 for all above items and miscellaneous books etc. Tel. Mr C Burton, (0788) 541835 (afternoons) 814040 (eves).

Brother HR-15 daisy printer with 3 wheels & ribbons, boxed £250, Acorn Master cartridges £6 each, Original double Acorn joysticks £11, Voltmace Datapad 16 £18, Voltmace Delta 14 joystick & keypad £12, Interbase £35, ACP Toolkit ROM £20, Master Ref. manuals 1&2 (new) £10 each. Tel. 081-989 2666.

Collecting or sending a work station from John O' Groats to Sussex is a different matter from a box of discs!! Tel. (0424) 219925.

M128 fitted with ADFS, Basic Editor, Terminal, View, Viewsheet and Viewspell, 18 months old, hardly used complete with manuals and packaging as new £325. Tel. 086 732 8776.

RX8 4 months old & satellite decoder module, total cost £320, accept £200 or part exchange for Epson compatible printer. Write to; Mr E Davis, 34 Edmonton Square, Carly Hill, Sunderland, Tyne & Wear SR5 2RD.

M128 £250, Philips green monitor £35, AMS double 2x100k 3" disc drive £35, Watford single 40/80T 5.25" disc drive £35, Viewstore ROM £10, Interword ROM £20, Wordwise ROM £12, Star 10X printer £50. Tel. (0689) 857403.

M512 second processor, full Gem software mouse etc., few hours use only £100, Viglen master pc style case, £15, Acorn Electron with Plus One, manuals, books, games and Morley data recorder and leads, very little used £55. Tel. 081-864 0309.

Heritage Genealogical complete package as new for Master. Tel. (0252) 715646.

BBC B with DFS, sideways ROM/RAM board with Wordwise Plus, Exmon, Commstar ROMs, lots of discs - (Elite, Aviator, Revs, Cholo, Sentinal etc.) and books £200 or can split, Music 5000 £80, Microvitec 1451 monitor £100, Canon PW1080A printer £100. Tel. (0734) 874769 after 6pm.

Cumana 40T single sided disc drive with PSU £35, Spellcheck II £15, Worswise Plus £20, all plus p&p. Tel. 03503 222 anytime.

Books: Econet Advanced User Guide £4, Designing Software Projects BBC Micro £4, Master Reference manuals 1&2 £15, A&B Computing, complete to current issue, buyer collects £30. **Discs:** Repton Infinity £6, The Real You (IQ and personality tests) £5, Disc User issues 2,4,7,9,10,11 (includes Repton Map Printer and extra screens) £8. **Tapes:** Speed Read course £4. Master 512, Acorn amber monitor, dual 5.25" 3.5" drives, mouse. All boxed, excellent condition £500. Tel. 081-698 3772.

M128, 512 co-processor, mouse, Gem software, Shibumi Problem Solver, dual Viglen 40/80T DS/DD disc drives, high resolution Digivision mono monitor and stand, 2 Peartree cartridges, WE EPROM programmer, complete set of BEEBUG magazines, ROMs including Interword, MOS+, Printmaster, Accelerator, Floppywise Master, many books and manuals including Advanced User Guide, M512 User Guide, DOS+ Amstrad manual, M512 Technical Guide, Advanced Sideways RAM User Guide, approx. 100 discs packed with software. Everything in superb condition £800 complete. Tel. 021-745 7458.

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		BEEBUG & RISC USER
£16.90	1 year (10 issues) UK, BFPO, Ch.I	£25.00
£24.00	Rest of Europe & Eire	£36.00
£29.00	Middle East	£43.00
£31.00	Americas & Africa	£46.00
£34.00	Elsewhere	£51.00

BACK ISSUE PRICES (per issue)

Volume	Magazine	5"Disc	3.5"Disc
5	£1.20	£4.50	£4.50
6	£1.30	£4.75	£4.75
7	£1.30	£4.75	£4.75
8	£1.60	£4.75	£4.75
9	£1.60	£4.75	£4.75

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	60p	30p
Europe + Eire	£1	50p
Elsewhere	£2	£1

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

December 1990 DISC CONTENTS

JUMBO CROSSWORD - Design your own crossword patterns or produce duplicates of existing crosswords to be used with different clues or by different members of the family.

CONTINUED FRACTIONS FOR PLEASURE - a program for mathematics lovers providing hours of entertainment with continued fractions.

MIKROTEL - a versatile database for storing telephone numbers,names and addresses, and a short routine for scrting entries in alphabetical order.

BEEBUG WORKSHOP: MONTE CARLO METHODS -three short programs to demonstrate the Monte Carlo method for simulating a table-tennis game, a lawn tennis game and roulette.

KRYPTON - FILE ENCRYPTION ON THE MASTER 128 - a file encryption utility for coding View files on the Master 128, which will run from sideways RAM.

PRACTICAL ASSEMBLER (7) - a program demonstrating the use of star commands from within Assembler.

SHUZZLE GAME - an entertaining and challenging puzzle, where 16 coloured shapes are shuffled to form a pattern.

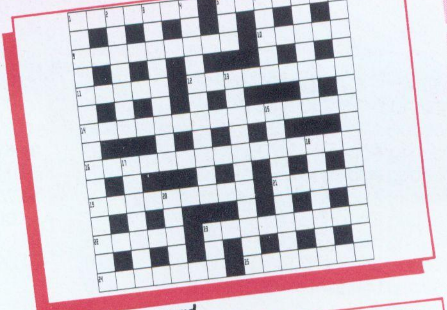
WORDPROCESSOR STYLE INPUT (2) - last month's routine, adapted to run from sideways RAM, and two example programs demonstrating more advanced uses of this routine.

★ **REDEFINING PRINTER CHARACTERS 128-255** - an improved set of printer character definitions, available only on the disc.

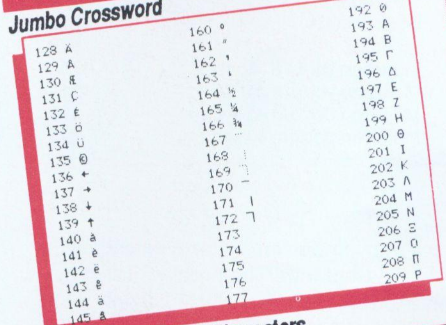
★ **MUSIC PIECES** - two sample music pieces 'Moris the Cat' and 'Purcell's Rondo' from the two of the discs reviewed this month.

★ **PLANET Z - BONUS ITEM !** - a 'try to run the planet' adventure game.

MAGSCAN DATA - bibliography for this issue (Vol.9 No.7).

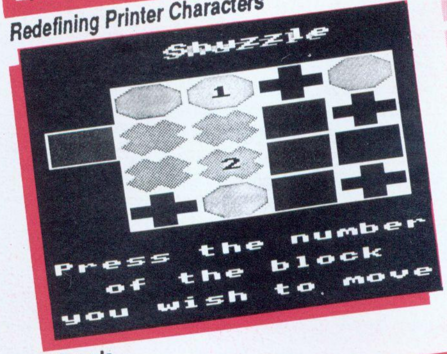


Jumbo Crossword



```
                      192 Ø
          160 °       193 A
128 Ä     161 "       194 B
129 Å     162 '       195 Γ
130 Æ     163 ‘       196 Δ
131 Ç     164 ½       197 E
132 £     165 ¼       198 Z
133 ö     166 ¾       199 H
134 ü     167 ¨       200 Θ
135 ©     168 ⋮       201 I
136 ←     169 ⌝       202 K
137 →     170 —       203 Λ
138 ↓     171 |       204 M
139 ↑     172 ┐       205 N
140 à     173         206 Ξ
141 è     174         207 O
142 ë     175         208 Π
143 ê     176         209 P
144 ä     177
145 å
```

Redefining Printer Characters



Shuzzle

Press the number of the block you wish to move

Shuzzle

Special Christmas Offers to BEEBUG Members

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**
1421B	Beebug Binder	3.99
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

Code	Product	Members Price inc.VAT
0077B	C - Stand Alone Generator	14.25
0088C	Masterfile ADFS BBC 40 T	16.50
0089C	Masterfile ADFS BBC 80 T	16.50
0081C	Masterfile ADFS M128 80 T	16.50
0024C	Masterfile DFS 40 T	16.50
0025C	Masterfile DFS 80 T	16.50
0085C	Printwise 40 Track	22.50
0086C	Printwise 80 Track	22.50
0009C	Studio 8	16.50
0074C	Beebug C 40 Track	44.25
0075C	Beebug C 80 Track	44.25
0084C	Command	29.25
0073C	Command(Hayes compatible)	29.25
0053C	Dumpmaster II	23.25
0004C	Exmon II	24.00
0087C	Master ROM	29.25

OTHER MEMBERS OFFERS

Due to a popular demand we are extending the offers from last month. The products listed below will be available for a limited period only while stocks are available.

Acorn's 512 Co-processor

Offer price £185 (inc VAT)

We have managed to obtain the last 25 of these from Acorn. These are brand new and will probably be the last 512s ever sold. Only while stocks last.

Stock code 0231G

ViewSheet

Normal Members price £42.49 (inc VAT)

Offer price £12.50 (inc VAT)

Acorn's 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

View 3 ROM

Normal Members price £51.75 (inc VAT)

Offer price £12.50 (inc VAT)

The enhanced version of the excellent View wordprocessor from Acorn. Supplied with manual,keystrip and printer driver generator on disc or tape. Many features over the original version of View.

Stock code 1022C

ViewSpell

Normal Members price £28.75 (inc VAT)

Offer price £7.00 (inc VAT)

The 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

ViewStore

Normal Members price £42.49 (inc VAT)

Offer price £13.80 (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, much more. Excellent value for money.

Stock code 1019B