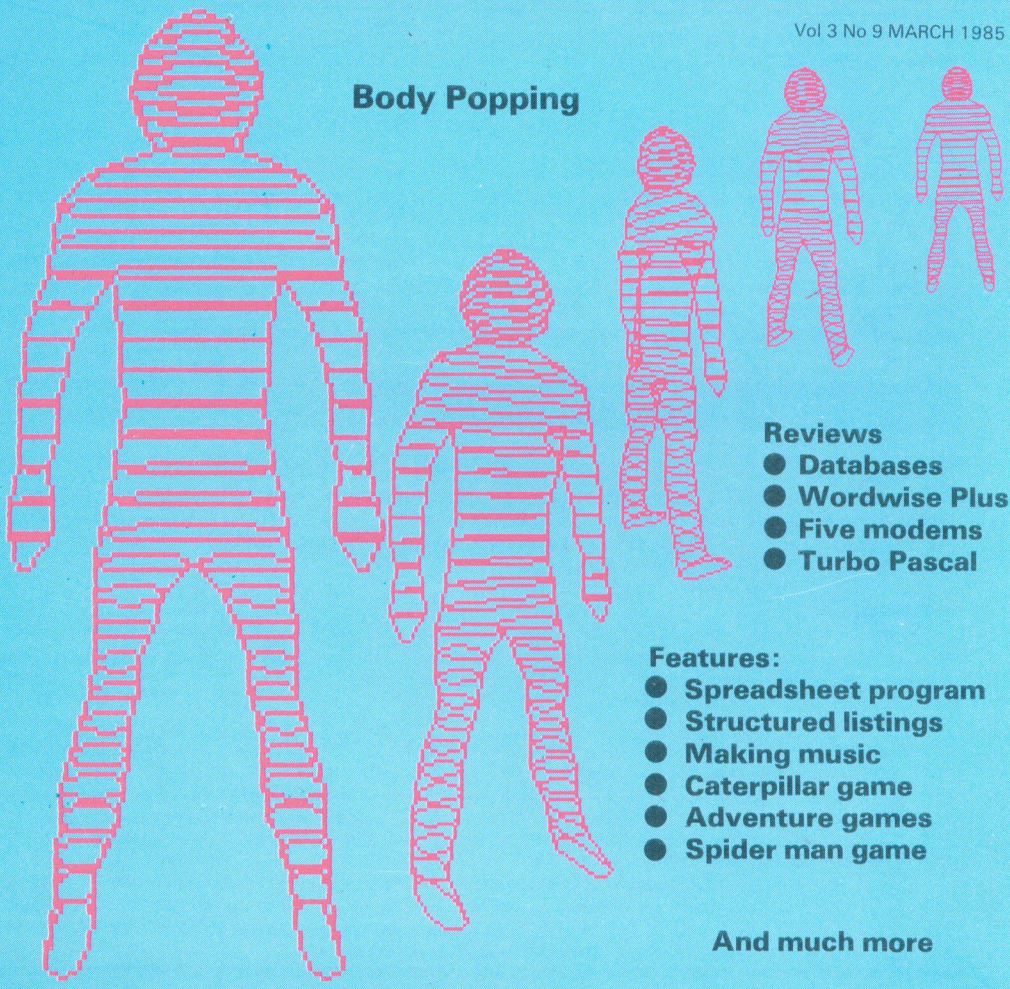
£1.20

BEEBUG

FOR THE BBC MICRO

Vol 3 No 9 MARCH 1985



Body Popping

Reviews
- Databases
- Wordwise Plus
- Five modems
- Turbo Pascal

Features:
- Spreadsheet program
- Structured listings
- Making music
- Caterpillar game
- Adventure games
- Spider man game

And much more

BRITAIN'S LARGEST COMPUTER USER GROUP
MEMBERSHIP EXCEEDS 25,000

EDITORIAL

THE QUEST FOR IMPARTIALITY

BEEBUG is a user group and this is the magazine of that user group. As such we feel that we have a responsibility to present informative, authorative, and above all impartial, reviews of hardware and software for the Beeb.

As most readers will be aware, BEEBUG (in the form of BEEBUGSOFT) is a major producer serious software for the BBC micro, and this raises the issue of whether the magazine should include reviews, even by independent reviewers, of our own products. We feel that if our members, and the suppliers of hardware and software, are to believe in the objectivity of our reviews, we should not include our own software in such assessments. But we cannot just ignore BEEBUGSOFT products when reviewing other comparable software. Instead we shall endeavour, in such cases, to include some factual information on the equivalent BEEBUGSOFT product for comparison. This is the policy that we have applied to this month's review of database packages, and BEEBUGSOFT's Masterfile.

We hope that in this way we can continue to publish objective and independent reviews and yet still provide the information that members will want. We also believe that it is better to face up to this situation rather than ignore the issue. Your own views and comments would be most welcome.

PAYMENTS TO AUTHORS

We have recently revised our rates of payment to authors and we will now be paying up to £40 per page for good quality programs and articles published in the BEEBUG. This represents a substantial increase on the previous rate.

It is important that all material submitted for potential publication should be to a high standard and clearly appropriate to BEEBUG in content and style. To assist potential authors, a leaflet entitled "Notes of Guidance to Contributors" is available on receipt of an A5 (or larger) SAE from the editorial address.

Mike Williams

NOTICE BOARD NOTICE BOARD NOTICE BOARD NOTICE BO

WORDWISE PLUS

We have arranged a discount on the new Wordwise Plus chip from Computer Concepts (see review in this issue). Details of this offer will be found in the supplement. We are not able to offer an upgrade service to existing Wordwise users who should deal directly with Computer Concepts.

AMX MOUSE

Many of you will have seen the advertisements for this impressive product which we shall be reviewing next month. We are arranging to offer this at a discount to members and full details will be available at the same time.

HINT WINNERS

We are awarding another £15 prize, this time to T.Thomson for his hint on coping with cassette errors, and a £10 prize to N.Silver for his hint on blanking screens. More hints, including ones for second processors and popular software, are always welcome.

MAGAZINE CASSETTE/DISC

This month's magazine cassette/disc contains one extra item, the winning entry in the "Sumsquares" Brainteaser competition by Bill Wilkinson. We had intended to publish this last month but ran out of space on the magazine disc. This was both an impressive and entertaining solution to the problem set.

BEEBUG MAGAZINE

GENERAL CONTENTS

2 Editorial
4 News
5 Body Popping
9 Wordwise Plus from Computer Concepts
12 A Spreadsheet Program (Part 1)
17 Adventure Games
19 Communicate with your Beeb
Five Modems Reviewed
24 Structured Listings of Basic Programs
27 Making Music on the Beeb (Part 2)
31 Beginners Start Here
Introducing Machine Code (Part 2)
33 Database Packages Reviewed
36 Masterfile II
37 Caterpillar
41 BEEBUG Workshop
Using Error Trapping
43 Turbo Pascal Reviewed
44 Postbag
46 Spider Man

HINTS, TIPS & INFO

11 Tube Mode Changes
11 Indirected Data
11 Blank Looks
26 Bug in Use of String Indirection Operator
26 Overcoming Cassette Data Errors
26 *RUN
30 OKI Pound Patch
49 Alternative REM
49 Print Formatting
49 Irretrievable Crash

PROGRAMS

5 Body Popping Program
11 Spreadsheet Program (Part 1)
15 Structured Listings
27 Making Music Examples
31 Beginners Machine Code Examples
37 Caterpillar Game
41 Workshop Examples on Error Trapping
44 Spider Man Game

NEWS NEWS NEWS NEWS

AN EDUCATION FOR ACORNSOFT

Acornsoft has attacked the home education market with a vengeance. The four titles - Workshop, ABC, Talkback, and Spooky Manor - all aim to "create opportunities for learning... instead of exercises with narrow, pre-determined goals", says Acornsoft. Each cost £9.95 on cassette and £11.50 on disc. Further details from Acornsoft on 0223-316039.

GETTING YOUR EDWORDSWORTH

CTL's popular Edword schools word processor is now available in a special home pack aimed at the home user. The pack contains the 16K word processing ROM, a user manual, function key insert, and printer configuring program. The whole package is available for £39.95 (incl. VAT) from CTL on 035283-751.

MORE SOFTWARE

The level of new software releases for the Beeb is still healthy. Three new animated adventures of the "rush around the screen collecting bits and pieces and avoiding nasties" type hail from Software Projects. These are "Crypt Capers", "Ledgeman", and "Ewgeebez" (no - even Software Projects doesn't know how to pronounce that), all at £7.95. The old favourite from the arcades are still comming in the form of "ERBERT" (a Q*BERT clone) from Microbyte (£6.95) and "Ultron" (a revamped Space Invaders) from Visions at £7.95. Two more arcade games are "Sim" from CSM and "Statix" from Psion, both at £7.95. If a more leisurely game is your idea of fun then the excellent Spectrum "Scrabble" from Leisure Genius has finally made it to the Beeb for £12.95 along with "Kensington" at the same price. Or try your hand at "Flight Path" from Storm Software for £9.95. Stay firmly on the ground with Acornsoft's "Cocktail Maker" or "Go" (a computerised implementation of the ancient board game) or even "Paul Daniel's Magic Show", all at £9.95. Another Spectrum conversion is the acclaimed graphic adventure, "Sabre Wulf" from Ultimate at £9.95. Acornsoft also have adventure offerings in the form of "Acheton", a colossal text adventure on TWO discs for £17.95, and Quondam, a more normal affair for a more normal price of £9.95.

VINE ROM AVOIDS THE GRAPES OF WRATH

Vine Micros, publishers of the excellent (if rather incompatible) Addcomm utility ROM has now produced another unusual ROM in the form of the 'T.D.ROM'. This unlikely sounding piece of software will transfer your tape programs to disc without (Vine Micros hopes) getting involved in the nasties of copyright law. T.D.ROM will copy tape software to disc complete with the existing protection and load back such files into the computer ready for use. In addition the T.D.ROMs themselves are varied and a disc saved with one ROM will not load back into a machine fitted with another. Clever stuff, especially since T.D.ROM will work with Acorn, Watford Electronics, and Amcom disc filing systems. Further information on T.D.ROM from Vine Micros on 0304-812276.

MONITORS ON THE CHEAP

If you fancy a 14 in colour monitor, ex-equipment but guaranteed for 90 days, for only £115 each, Give T. Lucas a ring on (0482) 701437. The monitors are TTL RGB models from a variety of name sources.

MORE LOGO

To add to the increasing number of 'full implementations' of Logo available for the BBC micro, Logotron has its version now available. Logotron Logo is contained in a 16K EPROM, is compatible with the second processor and costs £70 from Logotron on 0509-230248. BEEBUG will be reviewing all the full implementations of Logo in the near future.

COLLINS GOES SOFT

Collins the book publishers have entered the software market under the name of Collins Soft. To start it all off, Collins Soft has bought the rights to the "Brainpower" series of programs for small businesses produced by Triptych Publishing. For the BBC micro these include a Project Planner and Forecaster - both on tape/disc for £20/£25. Collins Soft is on 01-493 7070.

BODY POPPING

Bill Walker has come up with BEEBUG'S own body popping program, or more precisely, a program that allows you to display a realistic 3D representation of the human body on the screen. If you'd like to see yourself as others really see you, then this is the program for you.

Several programs have been published for the BBC micro which draw perspective views of three-dimensional objects. In each case, where the object is irregular and cannot be represented by a mathematical formula it has to be defined by a list of coordinates of key points on its surface. The 'Spitfire' by I. C. Grant (Beebug Vol.3 No.1) is a typical example of this, with over 1200 measurements defining 400 points on the Spitfire's surface.

This program deals with a very different subject and draws a representation of a three-dimensional human body, first from the front, and then from other directions. The program is complete with data (although you can enter your own data as well) so that you can try it straight away. Just type it in and let it run, making sure you have saved a copy first. The program will then display the body as viewed in turn from twelve different directions controlled by the spacebar. Be patient as many calculations are involved.

The program makes several simplifications which reduce the amount of information that has to be stored. The measurements are so straightforward that you can even replace the values given for those of your own body, and view yourself from different angles!

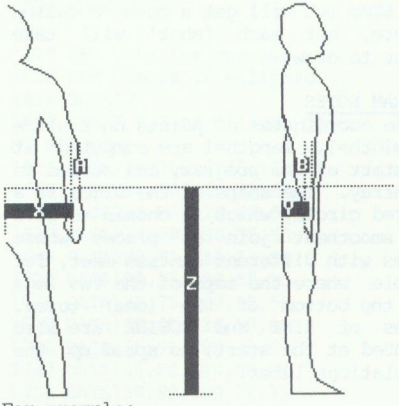
The first simplification assumes that the body is symmetrical about the middle. This makes the left leg the mirror image of the right, so we only need to measure and store one leg. The same applies for the arms.

The body is then carved up into 'slices'. In the measurements given, the slices are taken at 60mm intervals for the leg, arm and torso, and 30mm intervals (giving more detail) for the head.

Finally, the slices are assumed to have the same shaped outline, which is roughly circular, but to have different lengths and widths. For example, a slice through the foot (a footprint) is long and narrow, a slice through the head is more circular.

This means that the program only stores the position of each slice, and its length and width (five measurements), and from this it can compute the coordinates of 9 points (45 measurements) on the surface. Consequently the number of measurements stored is reduced by a factor of five!

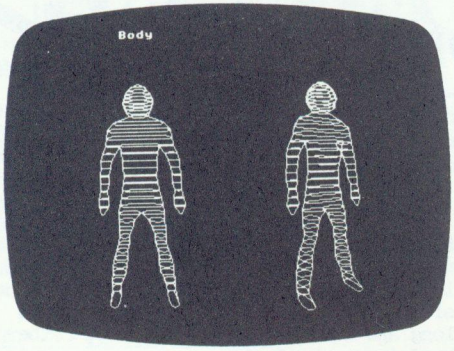
FIGURE 1: A slice through the arm
DATA X,Y,Z,A,B



For example:

DATA 240,160,0,40,120

represents a slice through the foot. Its centre is at X=240, Y=160, Z=0, ie 240 mm from the middle of the body, 160mm from the back of the body, and 0mm from the ground. The width of the slice (from centre to side, in the X direction) is 40mm, and the length (from centre to front, in the Y direction) is 120mm. (No prizes for working out what size feet I have!).



To take your own measurements, draw round your body to give a front and side outline, and take the measurements shown in figure 1 for each slice. All the measurements should be in millimetres. These measurements replace those in the data statements (the values for the leg start at line 1610, the torso at line 1910, the arm at 2160 and the head (and shoulders) at 2440). The first value in each set is the number of slices that follow. If you take slices at closer intervals than 60mm you will get a more detailed picture, but each 'shot' will take longer to draw.

PROGRAM NOTES

The coordinates of points on a slice of width=1, length=1 are computed at the start of the program, and stored in an array. The shape of the slice is a squared circle, which is chosen to give the smoothest join at places where slices with different centres meet, for example where the tops of the two legs meet the bottom of the (one) torso. Values of SINE and COSINE are also computed at the start, to speed up the calculations later.

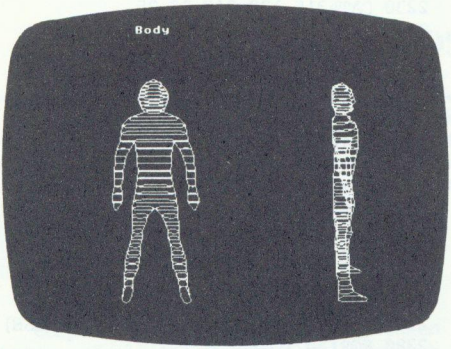
The main section of the program draws a front view of the body on the left of the screen, and then views it from different angles on the right.

PROCBODY draws the two legs, and notes the leftmost and rightmost points at the tops of the legs. It joins these to the bottom of the torso, then draws the arms. The left and rightmost points of the torso or arms are then joined to the shoulders and head.

PROCe calculates the coordinates of the points around the slice, and draws these on the screen, using PROCm to map the three-dimensional coordinates to the two-dimensional screen. The left and rightmost points on the slice are recorded for when the slice is joined to the next one.

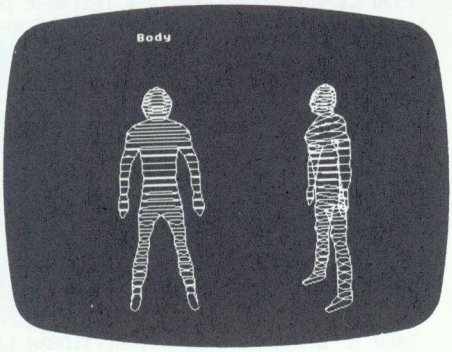
PROCLEG, PROCARM, PROCTORSO and PROCHEAD each draw bits of the body, joining them up to the other bits where appropriate. See also the many comments in the program for further information.

```
   10 REM PROGRAM BODY
   20 REM VERSION 0.03
   30 REM AUTHOR  B.WALKER
   40 REM BEEBUG  MARCH 1985
   50 REM PROGRAM SUBJECT TO COPYRIGHT
   60 :
  100 MODE4
  110 ON ERROR GOTO 2720
  120 PRINT'SPC(8)"Body"
  130 VDU23,1,0;0;0;0;
  140 REM Build lookup table and ellipse
  150 DIM S(35),C(35),SI(35),CO(35)
  160 FF=.3:REM fiddles elipses,0=no fi
ddle
  170 FORI%=0TO35
  180 SI(I%)=SINRAD(I%*10)
  190 CO(I%)=COSRAD(I%*10)
  200 S(I%)=SI(I%)*(1+CO(I%)*CO(I%)*FF)
  210 C(I%)=CO(I%)*(1+SI(I%)*SI(I%)*FF)
  220 NEXT
  230 :
  240 oLXP%=0:oLYP%=0:REM leftmost point
  250 oRXP%=0:oRYP%=0:REM rightmost poi
nt
  260 ANGLE=0:REM rotation angle/10
  270 VDIST%=2000:REM viewing distance
  280 ZOOM%=800:REM scale of drawing
  290 H%=1000
  300 :
  310 REM main loop
  320 VDU29,320;512;
  330 PROCBODY(0)
  340 VDU29,0;0;
  350 VDU24,640;0;1279;1023;
  360 VDU29,960;512;
  370 FORA=3 TO 35 STEP3
  380 CLG
  390 PROCBODY(A)
  400 Q=INKEY(500)
  410 NEXT
  420 END
  430 :
 1000 REM Draw Body
 1010 REM ...Draw legs, Draw Torso, Joi
n legs with torso
```




```
1020 REM ...Draw Arms, Draw head, Join
 arms with head
1030 DEFPROCBODY(ANGLE)
1040 PROCLEG(-1)
1050 TLX%=oLXP%:TLY%=oLYP%
1060 TRX%=oRXP%:TRY%=oRYP%
1070 PROCLEG(1)
1080 IF TLX%<oLXP% THEN oLXP%=TLX%:oLY
P%=TLY%
1090 IF TRX%>oRXP% THEN oRXP%=TRX%:oRY
P%=TRY%
1100 PROCTORSO
1110 TLX%=oLXP%:TLY%=oLYP%
1120 TRX%=oRXP%:TRY%=oRYP%
1130 PROCARM(-1)
1140 IF oLXP%<TLX% THEN TLX%=oLXP%:TLY
%=oLYP%
1150 IF oRXP%>TRX% THEN TRX%=oRXP%:TRY
%=oRYP%
1160 PROCARM(1)
1170 IF TLX%<oLXP% THEN oLXP%=TLX%:oLY
P%=TLY%
1180 IF TRX%>oRXP% THEN oRXP%=TRX%:oRY
P%=TRY%
1190 PROCHEAD
1200 ENDPROC
1210
1220 REM Draw ellipse, return lefmost
and rightmost points
1230 DEFPROCe(XOFF%,YOFF%,Z%,A%,B%,LIN
K%)
1240 REM XOFF YOFF = center of ellipse

1250 REM ellipse is in plane z=Z
1260 REM ellipse dia on x axis =A
1270 REM ellipse dia on y axis =B
1280 REM LINK=TRUE to join this and la
st ellipse
1290 LOCAL X%,Y%,K%
1300 X%=XOFF%:Y%=B%+YOFF%:REM start of
 ellipse
1310 PROCm(X%,Y%,Z%)
1320 MOVE XP%,YP%
1330 LXP%=XP%:LYP%=YP%:REM leftmost po
int
1340 RXP%=XP%:RYP%=YP%:REM rightmost p
oint
1350 FORK%=4TO35STEP4
1360 X%=A%*S(K%)+XOFF%
1370 Y%=B%*C(K%)+YOFF%
1380 PROCm(X%,Y%,Z%)
1390 DRAW XP%,YP%
1400 IF XP%<LXP%THEN LXP%=XP%:LYP%=YP%
1410 IF XP%>RXP%THEN RXP%=XP%:RYP%=YP%
1420 NEXT K%
1430 X%=XOFF%:Y%=B%+YOFF%
1440 PROCm(X%,Y%,Z%)
1450 DRAW XP%,YP%:REM close ellipse
1460 IF LINK% MOVE oLXP%,oLYP%:DRAW LX
P%,LYP%:MOVE oRXP%,oRYP%:DRAW RXP%,RYP%
1470 oLXP%=LXP%:oLYP%=LYP%
1480 oRXP%=RXP%:oRYP%=RYP%
1490 ENDPROC
1500
1510 REM Map 3D coords to 2D screen co
ords
1520 DEFPROCm(X%,Y%,Z%)
1530 LOCAL D,RX%,RY%
1540 RX%=X%*CO(ANGLE)+Y%*SI(ANGLE)
1550 RY%=-X%*SI(ANGLE)+Y%*CO(ANGLE)
1560 D=ZOOM%/(VDIST%-RY%)
1570 XP%=RX%*D
1580 YP%=(Z%-H%)*D
1590 ENDPROC
1600
1610 REM Data for one leg
1620 REM number of ellipses
1630 DATA12
1640 REM X,Y,Z of ellipse centre, Axes
 of ellipse
1650 DATA240,160,0,40,120
1660 DATA235,100,60,38,60
1670 DATA225,90,120,48,50
1680 DATA210,80,180,58,55
1690 DATA200,75,240,63,55
1700 DATA190,75,300,55,55
1710 DATA180,75,360,48,50
1720 DATA160,85,420,50,48
1730 DATA155,95,480,45,63
1740 DATA145,80,540,58,65
1750 DATA130,80,600,70,73
1760 DATA103,80,660,95,80
1770
1780 REM Draw one leg
1790 DEFPROCLEG(SIDE%)
1800 LOCAL I%,C%,X%,Y%,Z%,A%,B%
1810 RESTORE 1630
1820 READC%
1830 READX%,Y%,Z%,A%,B%
1840 PROCe(X%*SIDE%,Y%,Z%,A%,B%,FALSE)
1850 FORI%=2 TO C%
1860 READ X%,Y%,Z%,A%,B%
1870 PROCe(X%*SIDE%,Y%,Z%,A%,B%,TRUE)
```




```
1880 NEXT
1890 ENDPROC
1900
1910 REM Data for torso
1920 REM number of ellipses
1930 DATA9
1940 REM X,Y,Z of ellipse centre, Axes
of ellipse
1950 DATA0,85,720,185,83
1960 DATA0,85,780,175,85
1970 DATA0,85,840,160,83
1980 DATA0,90,900,150,85
1990 DATA0,100,960,140,90
2000 DATA0,105,1020,145,95
2010 DATA0,115,1080,150,100
2020 DATA0,130,1140,155,105
2030 DATA0,120,1200,170,113
2040
2050 REM draw torso
2060 DEFPROCTORSO
2070 LOCAL I%,C%,X%,Y%,Z%,A%,B%
2080 RESTORE 1930
2090 READC%
2100 FORI%=1 TO C%
2110 READX%,Y%,Z%,A%,B%
2120 PROCe(X%,Y%,Z%,A%,B%,TRUE)
2130 NEXT
2140 ENDPROC
2150
2160 REM Data for one arm
2170 REM number of ellipses
2180 DATA10
2190 REM X,Y,Z of ellipse centre, Axes
of ellipse
2200 DATA330,145,660,10,5
2210 DATA320,135,720,45,28
2220 DATA320,130,780,40,23
2230 DATA315,120,840,30,25
2240 DATA310,105,900,38,35
2250 DATA310,100,960,38,40
2260 DATA300,100,1020,40,43
2270 DATA290,90,1080,48,40
2280 DATA265,85,1140,53,55
2290 DATA230,85,1200,60,75
2300
2310 REM Draw one arm
2320 DEFPROCARM(SIDE%)
2330 LOCAL I%,C%,X%,Y%,Z%,A%,B%
2340 RESTORE 2180
2350 READC%
2360 READX%,Y%,Z%,A%,B%
2370 PROCe(X%*SIDE%,Y%,Z%,A%,B%,FALSE)
2380 FORI%=2TOC%
2390 READX%,Y%,Z%,A%,B%
2400 PROCe(X%*SIDE%,Y%,Z%,A%,B%,TRUE)
2410 NEXT
2420 ENDPROC
2430
2440 REM Data for head and shoulders
2450 REM number of ellipses
2460 DATA12
2470 REM X,Y,Z of ellipse centre, Axes
of ellipse
2480 DATA0,115,1260,280,115
2490 DATA0,105,1320,268,105
2500 DATA0,85,1380,220,80
2510 DATA0,75,1440,78,43
2520 DATA0,115,1470,110,78
2530 DATA0,100,1500,123,88
2540 DATA0,105,1530,128,108
2550 DATA0,100,1560,125,100
2560 DATA0,95,1590,115,95
2570 DATA0,90,1620,100,90
2580 DATA0,85,1650,75,70
2590 DATA0,90,1680,30,25
2600
2610 REM Draw one head
2620 DEFPROCHEAD
2630 LOCAL I%,C%,X%,Y%,Z%,A%,B%
2640 RESTORE 2460
2650 READC%
2660 FORI%=1TOC%
2670 READX%,Y%,Z%,A%,B%
2680 PROCe(X%,Y%,Z%,A%,B%,TRUE)
2690 NEXT
2700 ENDPROC
2710
2720 :
2730 ON ERROR OFF:MODE 7
2740 IF ERR=17 END
2750 REPORT:PRINT" at line ";ERL
```

The techniques described above offer many opportunities for further experiment. You can easily change the overall shape and appearance of the figure by changing the values assigned to the variables ANGLE, VDIST%, ZOOM% and H% in lines 260 to 290. The viewing angle can be changed by modifying the parameters in the main loop from line 370 to 410. Other ideas worth considering might include views from different vertical angles, and even a form of animation using VDU19.

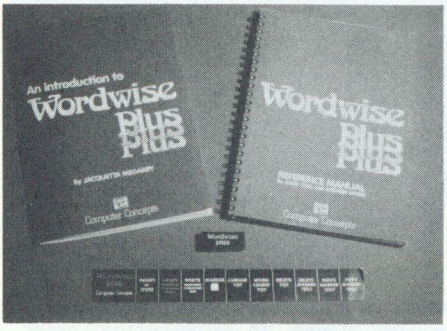
WORDWISE PLUS FROM COMPUTER CONCEPTS

Wordwise has proved to be one of the most popular and user-friendly of word processors for the BBC micro. Now that Wordwise Plus has been released with many new features, we asked Stephen Ibbs, a long-time user of both Wordwise and View, to take a detailed look at Computer Concepts' new ROM.

```
Wordwise Plus  £56.25
Upgrade from Wordwise  £19.55
'High' Wordwise Plus disc  £4.95
```

Computer Concepts have clearly taken notice of all the comments and feedback about the already popular Wordwise. The result is a 16k version, compatible with the Aries B-20, Watford RAM board, and the 6502 second processor. It has virtually all the additions that one could ask for, accompanied by a superb 176 page manual. There is also an introductory booklet which is eminently readable and 17 pages longer than the original Wordwise manual! The manual bears a strong similarity to the BBC User Guide, hardly surprising, as John Coll is one of the authors.

Wordwise Plus is compatible with any text files written with Wordwise 1.17 or later, and all the old commands still work with one or two minor changes. The default values for TS and BS have been reduced from 7 to 6, and in the cases where the pad character (PC) needs to be redefined, the new character MUST be enclosed in quotes.



The manual lists over 40 new features, including fast disc loading and saving (10 times faster), and 'safety-netting' to prevent accidentally overwriting existing files etc. There are 15 new embedded commands (some defaulting to work with the Epson range of printers), as shown in the table below.

The underlining and double strike modes for the Epson are shown as underlined, or inverse video when previewed in 80 column mode on the screen. The FI command is the same as View's 'RJ' and places, for example, an address hard on the right hand edge. One small but useful feature is that a page boundary is now marked in preview mode with a dotted line. This makes adjusting the header and footer positions extremely easy. The BP command has been slightly altered so that if it is the final command on the page, with no f2 (white) command or Return after it, then the printer stops at the end of the page, and doesn't produce the top spaces and header of the next page, a most welcome improvement!

COMMAND	RESULT
*	Operating system call (duplicates OS in version 1.2)
DE	Double strike end
DS	Double strike start
ES	Escape sequence (for the start of printer sequences)
FI	Fully indent (to justify hard on the right hand margin)
LNE	Line number end
LNS	Line number start (line number displayed on screen left)
OPS	Output sequence (10 printer commands definable)
PA	Pause
PF	Print File
PS	Print string
RPS	Redefine print sequence
SEG	Execute segment
UE	Underline end
US	Underline start

Three further Ctrl commands have been added, namely 'Ctrl-R' to delete markers, 'Ctrl-W' to recount the words, and 'Ctrl-F' to toggle the on-screen formatting. The number of possible TAB positions has been increased from 10 to 14, and the 'Search and Replace' option has been improved to accept the wildcard character # as well as TAB (|T), Return (|R), f1 green (|G), and f2 white (|W).

However, perhaps the most important addition is the new menu option 9, entitled 'Segment menu'. There are now 10 segments of memory available in addition to the main text area, and there is no limit on the available memory for each area (subject to the computer's overall total of course). Each segment can store either text or a procedure, and can be called up from the main text edit mode using Shift and f0-f9.

The segment area has its own menu, very similar to the main menu, for loading, saving, previewing, printing etc., so if for example you are writing an article, and you want to jot something down, or generate a letter quickly, there is no need to save and start again, simply go to a segment, write the letter, print it and then go back to the main text. Another possibility is to store standard paragraphs in different segments, then call them up to produce various letters.

In addition, Wordwise Plus provides a complete programming language for generating procedures. These can be written in segments, then recalled, and the manual gives the following 4 line example:

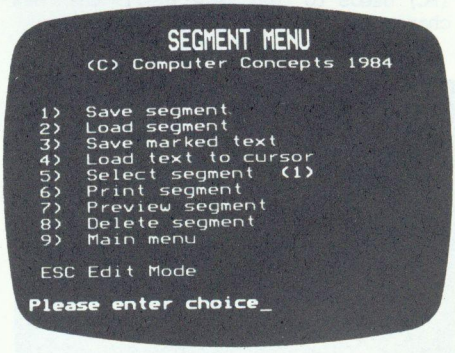
```
SELECT TEXT
DOTHIS
 PRINT TEXT
TIMES 6
```

Typing Shift-f0 from either menu will cause the six copies of the main text to be printed automatically. The above example shows that new commands and functions are available and the manual lists over 70, many recognisable from their Basic counterparts. Most can be used either within a program or as an immediate command from a menu, and some actually duplicate the menu and 'red-key' functions. To attempt to condense the manual down for this review would obviously be hopeless, but it is clear that the segment concept transforms Wordwise into a very powerful tool for home, office and educational use.

The new language allows powerful features such as mail-merging and index-generation to be created. The package includes a cassette of example procedures which illustrate just how sophisticated this word processor is.

There is a mail-merging program which will take a name and address (or any text) from either file or segment, insert it at the top of the main text, print the combined document, and continue this process until the file of addresses is exhausted. There is also a continuous processing program, going some way to solving a major criticism of Wordwise, its limit on text length. Whole files can be strung together, with any formatting commands retained. The files still have to be edited and stored separately, but now whole books can be written with relative ease!

Another program will enable two-column printing, (the style used on this BEEBUG page), working on a file that has been spooled using option 8.



There is a 'sorting' program that will arrange lists in alphabetical order, not only useful for names and addresses, but also for use with a further indexing program. This will enable users to pick key words from the

text which will then be listed with every page reference next to them. When these are linked with the continuous processing program, readers will realise that the introduction of segments has really pushed back the boundaries of Wordwise.

The obvious competitor, both in price and performance is View 2.1. I have used this for some time, but now gladly return to Wordwise Plus because of its ease of use and seemingly limitless possibilities. View will allow editing in any mode (a rather over-rated advantage I feel), but now Wordwise Plus will also permit mode changes, if the second processor or Aries B-20 board is fitted. I like being able to delete whole lines of text in View, not possible in Wordwise Plus unless set up by a segment program or function key. Like View, files can now be printed or previewed without having to load them into memory, a most useful feature if you are working on a document and need to call on some other information. The continuous processing is better in View, and this, together with its on-screen formatting, is where View scores.

However, View is well-known for being difficult to master, certainly for the first time word processing user, whereas Wordwise Plus positively encourages users to 'have a go'. There is no need to purchase printer-generation programs, and there are many more printer controls available. The manuals need to be read and understood thoroughly, not so much to learn how to use the system, but to discover all the possibilities.

It is churlish, as well as difficult, to find any faults at all. Wordwise Plus is less informative than it might be at times so that you can be left not knowing whether a command has worked correctly, if at all. There are one or two tiny errors in the manuals... the comment about all default values remaining the same is ALMOST true, and I wish a list of the keywords had been included to enable users to program the function keys, as displayed in the old manual. Overall this is a stunning package, carefully thought out, and beautifully presented.



HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

TUBE MODE CHANGES

Users of the 6502 Second Processor may be interested to know that changing mode within a procedure will not generate an error in Basics I, II and Hi as happens on a single processor system. There is specific check within Basic to see if it is running in the Second Processor and within a procedure; if it is it skips the error generation. This is accomplished by testing the high order address, and so trapping this call (i.e. the read high order address OSBYTE call) provides a method of changing mode within a procedure on a normal machine if it is 'known' to be 'safe' (e.g. when the new mode occupies the same amount of memory).



INDIRECTED DATA - D. Morgan

If you ever find yourself in the situation of needing to read a lot of data into memory using indirection operators, then it is worthwhile knowing that the 'memory access' can be within the READ statement. For example:

```
FORI%=0 TO 99:READ I%?memoryarea:NEXT
```

where 'memoryarea' is the address of an area in memory that we have previously assigned to this variable.



BLANK LOOKS - N. Silver

Blanking the screen while a picture is being drawn is usually done by redefining all the colours to zero using a series of VDU19 commands. A much shorter method is to set the horizontal 'sync' frequency to zero using VDU23;0;0;0;0;. To get back to normal when the picture is drawn use VDU23;n;0;0;0; where n is 127 for modes 0 to 3 and 63 for modes 4 to 7.

A SPREADSHEET PROGRAM (Part 1)

This month we present the first part of our very own Spreadsheet program. This is complete as it stands, and provides most of the basic features needed for this popular application. Next month we shall add some of the more advanced features.

INTRODUCTION.

Spreadsheet programs are among the most popular of the more practical applications of micros and are particularly useful for all kinds of forecasting, particularly financial, and answering questions of the "What if....?" variety.

Previous articles on spreadsheets have appeared in BEEBUG, notably Vol.3 Nos. 2 & 3, and readers should refer to these for an introduction to this subject. These articles also reviewed Ultracalc and Viewsheet, two of the most comprehensive spreadsheet packages currently available for the Beeb. For ordinary householders, and those not yet convinced, the cost of such programs may rule them out. The BEEBUG spreadsheet program will satisfy the needs of many users, while those who feel they may require a more powerful package, may gain valuable experience with this one first.

GENERAL INFORMATION

The spreadsheet program SPREADX is written entirely in Basic, in mode 7 (though it is also possible to run it in mode 3 with fewer rows and columns). With a maximum size of 26 columns by 26 rows, wide scope is present for many applications. The program is presented in two parts. This month's instalment is nearly 2/3 of the whole program, and caters for basic requirements. It is a working program in its own right. Next month will give the balance by adding the 'goodies', and allowing storage and retrieval of ALL items of a model (data, headings, calculation specifications, etc.). User specified row and column headings are limited to three characters while the maximum data size is 5 digits plus decimal point.

PROGRAM USES

You can use SPREADX for your family's annual budget, and then for recording actual expenditure (automatically you get your bank and savings balances). Thus you know what monies you have 'spare'. Income Tax assessments (courtesy of Which), house contents insurance, cricket club accounts, etc., etc., once set up are quickly and easily used over and over, and can be inspected and amended. Alternatives can be tried to assess their effect, AND, when satisfied, printed out. Errors can be corrected and changes made in seconds.

A SIMPLE EXAMPLE

The program presents the spreadsheet to you with letters above each column and a letter at the side of each row. Your own headings (a maximum of 3 characters) are also shown. Suppose, by way of example, we are doing the accounts for the two village halls for 1985. We might want to show something like this:

		1ST	2ND	TTL	84	DIF
		A	B	C	D	E
RTS	A	100	50		120	
HTG	B	200	50		200	
WTR	C	50	10		50	
RPR	D	25	50		10	
NEW	E	30	10		20	
TTL	F					
INC	G	-400	-300		-600	
NET	H					

Example Spreadsheet

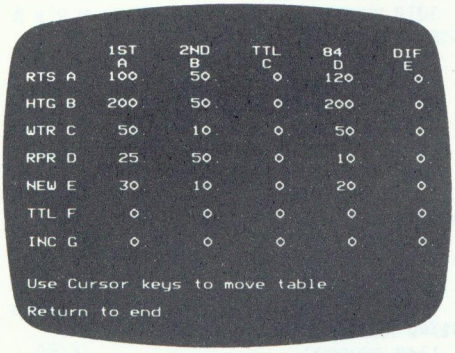
Let us see how we can create this spreadsheet. Assuming the program listed this month is typed in and saved already, running the program will ask for the number of columns (enter 5) and rows (enter 9) required. The main menu now appears.

OPTION 1 allows us to enter our column headings, one by one. Anything we enter will over-write previous entries, but a nul entry (Return only), leaves that entry unchanged. To exit

enter '@' followed by Return.

OPTION 2 is similar for rows.

OPTION 3 is for data which is entered row by row. A '/' will pass us to the next row, and '@' will exit as before. Existing contents are shown. Any entry over-writes, but a nul entry (Return only) leaves entries unchanged. In our example we will be prompted AA?, so we enter 100 (for row A column A), then for AB? enter 50, AC? enter 0 (Return only is alright leaving the existing 0), AD? 120, and so on. If the row and column headings and data have been entered correctly, the spreadsheet should appear as shown (you should be able to deduce the meanings of the row and column headings yourself).



	1ST A	2ND B	TTL C	84 D	DIF E
RTS A	100	50	0	120	0
HTG B	200	50	0	200	0
WTR C	50	10	0	50	0
RPR D	25	50	0	10	0
NEW E	30	10	0	20	0
TTL F	0	0	0	0	0
INC G	0	0	0	0	0

Use Cursor keys to move table
Return to end

OPTION 4 is to specify the calculations we require. You have to be careful here to do things in the right order. For example it is no use adding up column C until it has been filled with the sum of columns A + B. Entering S will SHOW existing 'Specs'. The Amend (and the FULL Delete) will be given next month, but for now you must delete and re-enter (this month's delete procedure covers simple usage only). Each 'Spec' is entered as a numbered 'index', with the first numbered zero. This is for subsequent reference.

A 'Spec' involves specifying the 'box' (i.e. position) involved, and then what goes in to it. A simple example is AB=123 (row A column B to contain 123). Any Basic expression can be used with any of the 'boxes' as desired, e.g:

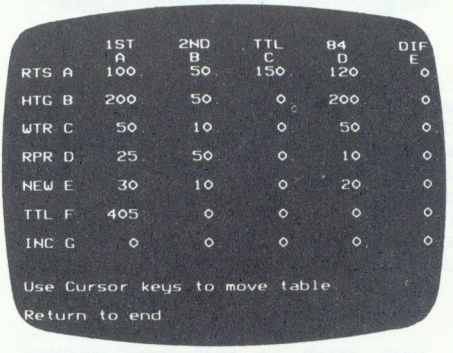
```
AA=AB*BB/(CB-DB)+36
```

(but actual numbers are best inserted as data using option 3). In our spreadsheet example we could use AC=AA+AB (row A column C to contain the sum of row A columns A + B).

Now to make such an entry for every row of column C, and every column of rows F and H could be tedious, so a facility has been built into the program to do this with a single entry. We specify the calculation for the first box, but use ';' instead of '+' to separate the first and last of the columns to be added. In our example AC=AA;AB only adds two columns so it doesn't save much, but in adding up the rows we can see the benefit: FA=AA;EA does the work of FA=AA+BA+CA+DA+EA. Now select menu option 4, I for Insert, and enter these two 'Specs':

```
Index 0 AC=AA;AB
Index 1 FA=AA;EA
```

If you now press 'M' to return to the main menu, and then select option 5, you can see the results.



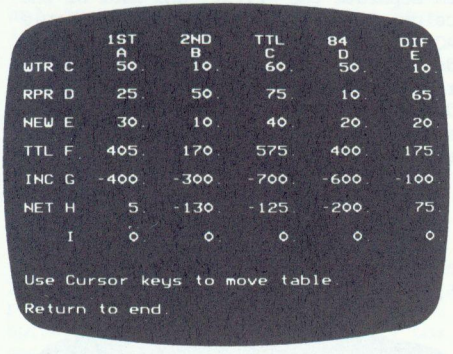
	1ST A	2ND B	TTL C	84 D	DIF E
RTS A	100	50	150	120	0
HTG B	200	50	0	200	0
WTR C	50	10	0	50	0
RPR D	25	50	0	10	0
NEW E	30	10	0	20	0
TTL F	405	0	0	0	0
INC G	0	0	0	0	0

Use Cursor keys to move table
Return to end

Return now to 'Specs' with menu option 4. To add entries as above for all rows of column C would still be tedious, but we can go further still. If we specify how many times to repeat, we can do the lot in one entry. We do this by adding @ on the end, followed by the LAST positions to be filled. We already have AC=AA;AB, but we want to go on adding column A to column B down to row G. So we use AC=AA;AB@G for the rows and FA=AA;EA@D for the columns. Now replace 'Specs' Index 0 and 1 by the above (you will have to delete and reinsert - simply entering would move

the existing Index 0 and 1 to positions 2 and 3).

The facilities for multiplication, division and also subtraction applied to groups of rows or columns will be given next month. In the meantime, if you want to continue, it will be necessary to specify column E for each row e.g. AE=AC-AD, BE=BC-BD, and so on. If the values in row G are entered as negative amounts (using option 3 again), then HA=FA;GA@E will give the net figures (see below for results).



OPTION 5 Always does the calculations before displaying your model, and the FULL program ensures that changes to 'Specs' do not leave in any 'old' figures. You can also use the cursor control keys to move the display on the screen up/down/left/right, which is particularly useful if your spreadsheet is larger than the capacity of the screen display.

OPTIONS 6/7/8/9 - see next month.

Next month we will complete the program by providing all the additional features mentioned. Full instructions will also be supplied, and there will be program notes for the technically minded. In the meantime you have a fully working program which you may like to try out with some of your own applications.

NOTE: Please keep strictly to the line numbering if your program is to match up with part 2 next month.

```
  10 REM Program SPREADX
  20 REM Author A.BEEBER
  30 REM Version 1.9E/1
  40 REM BEEBUG MARCH 1985
  50 REM Program subject to Copyright
  60 :
 100 MODE 7:ON ERROR GOTO 1820
 110 PROCinit
 120 REPEAT
 130 PROCmenu
 140 IFoption%=1 THEN PROCcolrowin(x%,
maxcol$,"column")
 150 IFoption%=2 THEN PROCcolrowin(y%,
maxrow$,"row")
 160 IFoption%=3 THEN PROCdatain
 170 IFoption%=4 THEN PROCspecs
 180 IFoption%=5 THEN PROCscreen
 230 UNTILoption%=0
 240 VDU26:CLS
 250 END
 260 :
1000 DEFPROCmenu
1010 VDU26:CLS:PRINTTAB(7,3)"S P R E A
 D X   M E N U"
1020 PRINTTAB(7,5)"1. Column headings"
1030 PRINTTAB(7,6)"2. Row headings"
1040 PRINTTAB(7,7)"3. Data entry"
1050 PRINTTAB(7,8)"4. Specifications"
1060 PRINTTAB(7,9)"5. Display"
1070 PRINTTAB(7,10)"6. Load data"
1080 PRINTTAB(7,11)"7. Save data"
1090 PRINTTAB(7,12)"8. Hard copy"
1100 PRINTTAB(7,13)"9. Edit @%"
1110 PRINTTAB(7,14)"0. End"
1120 PRINTTAB(4,15)"Which Option ?"
1130 REPEAT:option%=GET-48:UNTILoption
%>=0 AND option%<=9
1140 ENDPROC
1150 :
1160 DEFPROCspecs
1170 CLS:PRINT"Specifications"
1180 VDU28,0,24,39,3:@%=&3
1190 REPEAT:CLS
1200 PRINT"S=Show.I=Insert.A=Amend.D=D
elete.M=Menu"
1210 G$=GET$:IFG$="S" THEN PROCshow EL
SE IFG$="I" THEN PROCin ELSE IFG$="A" T
HEN PROCam ELSE IFG$="D" THEN PROCde
1220 UNTIL G$="M"
1230 ENDPROC
1240 :
1250 DEFPROCscreen
1260 PRINT"Please wait."'"Calculations
being made"
1270 PROCevaluate
1280 PROCprint(c%,7,r%)
1290 ENDPROC
1300 :
1310 DEFFNconvert(A$)
1320 J%=(LEN(A$)-1)
```

```
1330 FORI%=3TOJ%
1340 IFMID$(A$,I%+3,1)="("THENI%=I%+4
1350 IF(MID$(A$,I%+1,1)>"@"ANDMID$(A$,
I%+1,1)<"[")AND(MID$(A$,I%,1)>"@"ANDMID
$(A$,I%,1)<"[")THENA$=FNconv2(A$,I%)
1360 NEXT
1370 IF LEN(A$)>J%+1 THEN1320
1380 =A$
1390 :
1400 DEFFNconv2(B$,I%)
1410 A1%=ASC(MID$(B$,I%,1))-64
1420 A2%=ASC(MID$(B$,I%+1,1))-64
1430 IFI%=1THENST$="" ELSEST$=MID$(B$,
1,I%-1)
1440 B$=ST$+"mat("+STR$(A1%-1)+","+STR
$(A2%-1)+")"+MID$(B$,I%+2,(LEN(A$)-I%-1
))
1450 =B$
1460 :
1470 DEFFNcont(xpos%,ypos%,msg$,char$)
1480 X%=POS:Y%=VPOS
1490 PRINTTAB(xpos%,ypos%);msg$;:*FX15
,1
1500 REPEAT:G$=GET$:UNTIL INSTR(char$,
G$)
1510 PRINTTAB(xpos%,ypos%);SPC40
1520 PRINTTAB(X%,Y%);
1530 =G$
1540 :
1550 DEFPROCshow
1560 VDU14,12
1570 PRINT"Index";SPC8;"Specifications"
1580 FOR I%=0 TO M%-1
1590 IFSpec$(I%)=""THEN1610
1600 PRINTI%;TAB(13);Spec$(I%)
1610 NEXT:VDU15
1620 G$=FNcont(0,20,"Return to end",CH
R$13)
1630 ENDPROC
1640 :
1650 DEFPROCin
1660 CLS:PRINT"Index";SPC6;"Specificat
ions"
1670 REPEAT
1680 REPEAT:I%=FNinput(POS,VPOS,3,"I")
:UNTIL I%<=M%
1690 IF NUL% THEN I%=M%:PRINTTAB(0,VPO
S);I%;
1700 sp$=FNinput(10,VPOS,19,"S")
1710 IF NUL% THEN1780
1720 IF Spec$(I%)=""THEN1760
1730 FORJ%=M% TO I%+1 STEP-1
1740 Spec$(J%)=Spec$(J%-1)
1750 NEXT
1760 REMPROCvalidate(sp$)
1770 Spec$(I%)=sp$:PRINT:M%=M%+1
1780 G$=FNcont(0,19,"Space to continue
- Return to exit",CHR$32+CHR$13)
1790 UNTIL G$=CHR$13
1800 ENDPROC
1810 :
1820 ONERROR OFF:REPORT:PRINT" at line
";ERL:*FX4,0
1830 IF ERR=18 PRINTTAB(10,23)"W A I T
":G%=INKEY(500):GOTO120
1840 PRINT"To recover, try 'GOTO 120'"
:PRINT"and press Return":VDU26:END
1850 :
1860 DEFPROCprint(co%,st%,ro%)
1870 REPEAT
1880 VDU26,12:*FX15,1
1890 FOR I%=6 TO co%*st% STEP st%
1900 PRINTTAB(I%+2,0);col$(col%+(I%DIV
st%))
1910 PRINTTAB(I%+3,1);CHR$(col%+(I%DIV
st%)+B%)
1920 NEXT:R%=row%
1930 FOR J%=2 TO 2*ro% STEP 2
1940 C%=col%
1950 PRINTTAB(0,J%);row$(row%+(J%DIV2)
-1)
1960 PRINTTAB(4,J%);CHR$(row%+(J%DIV2)
+B%-1)
1970 FORI%=6 TO co%*st% STEP st%
1980 @%=edit%(C%):PRINTTAB(I%,J%),mat(
R%,C%):C%=C%+1
1990 NEXT:R%=R%+1
2000 NEXT
2010 PRINT'"Use Cursor keys to move ta
ble."'"Return to end."
2020 UNTIL FNreturn
2030 ENDPROC
2040 :
2050 DEF FNreturn
2060 nodisplay%=FALSE:*FX4,1
2070 REPEAT:G$=GET$:U%=INSTR(cursor$,G
$):UNTIL U%
2080 IFU%=1ANDcol%<(x%-5)THENcol%=col%
+1
2090 IFU%=2ANDcol%>0THENcol%=col%-1
2100 IFU%=3ANDrow%>0THENrow%=row%-1
2110 IFU%=4ANDrow%<(y%-7)THENrow%=row%
+1
2120 IFU%=5THENnodisplay%=TRUE
2130 *FX4,0
2140 =nodisplay%
2150 :
2160 DEF PROCcolrowin(cr%,maxcr$,cr$)
2170 CLS:PRINT CHR$(ASC(cr$)AND&DF)+MI
D$(cr$,2)+" Headings"
2180 PRINT"Enter ";cr$;" headings from
A to ";maxcr$'"(@ to end)"'
2190 VDU28,0,24,39,4
2200 FOR I%=0 TO cr%-1
2210 PRINT'CHR$(I%+B%);:A$=FNinput(POS
+1,VPOS,3,"S")
2220 IFA$="" ELSE IFA$="@"THENI%=cr% E
LSE IFcr$="row"THENrow$(I%)=A$ ELSE col
$(I%)=A$
2230 NEXT
2240 ENDPROC
2250 :
```

```
 2260 DEFPROCdatain
 2270 CLS:PRINT"Data Entry"'"(/ for nex
t row - @ to end)"
 2280 VDU28,0,24,39,4
 2290 FOR I%=0 TO y%-1
 2300 FOR J%=0 TO x%-1:@%=edit%(J%)
 2310 PRINT CHR$(65+I%)+CHR$(65+J%)+"="
;mat(I%,J%);SPC(5);
 2320 INPUT A$
 2330 IF A$="@" THEN I%=y%:J%=x% ELSE I
F A$="/" THEN J%=x% ELSE IF A$<>"" THEN
 mat(I%,J%)=VAL(A$)
 2340 NEXT J%,I%:VDU26
 2350 ENDPROC
 2360 :
 2370 DEFPROCevaluate
 2380 FORK%=0TO M%-1
 2390 IFSpec$(K%)=""THEN2490
 2400 IFMID$(Spec$(K%),9,1)="@" THENPRO
Cplursum(Spec$(K%)):GOTO2490
 2410 IFMID$(Spec$(K%),6,1)=";"THENPROC
summate(Spec$(K%)):GOTO2490
 2420 IFMID$(Spec$(K%),3,1)="." THENPRO
Cmultiply(Spec$(K%)):GOTO2490
 2430 IFMID$(Spec$(K%),3,1)="_" THENPRO
Cmultwo(Spec$(K%)):GOTO2490
 2440 B$=FNconvert(Spec$(K%))
 2450 A$=RIGHT$(B$,(LEN(B$)-3))
 2460 V%=(ASC(LEFT$(Spec$(K%),1)))-65
 2470 Z%=(ASC(MID$(Spec$(K%),2,1)))-65
 2480 mat(V%,Z%)=EVAL(A$)
 2490 NEXT
 2500 ENDPROC
 2510 :
 2520 DEFPROCsummate(A$)
 2530 result=0
 2540 AA%=ASC(MID$(A$,4,1))-65:AB%=ASC(
MID$(A$,5,1))-65
 2550 AC%=ASC(MID$(A$,7,1))-65:AD%=ASC(
MID$(A$,8,1))-65
 2560 A1%=ASC(LEFT$(A$,1))-65:A2%=ASC(M
ID$(A$,2,1))-65
 2570 IFAA%=AC%THEN2650
 2580 IFAB%<>AD%THEN PROCerror:ENDPROC
 2590 IFAC%<AA%THEN PROCerror:ENDPROC
 2600 A$=LEFT$(A$,3)
 2610 FORI%=AA%TOAC%
 2620 result=result+mat(I%,AB%)
 2630 NEXT
 2640 GOTO2700
 2650 IFAD%<=AB%THEN PROCerror:ENDPROC
 2660 A$=LEFT$(A$,3)
 2670 FORI%=AB%TOAD%
 2680 result=result+mat(AA%,I%)
 2690 NEXT
 2700 mat(A1%,A2%)=result
 2710 ENDPROC
 2720 :
 2730 DEFPROCerror
 2740 PRINT'"Error in - ";K%;SPC1;Spec
$(K%):PRINT"Space to continue"
 2750 G%=GET
 2760 ENDPROC
 2770 :
 2780 DEFPROCplursum(A$)
 2790 LOCALB$,C$,D$,DD%,E$,i%
 2800 IFMID$(A$,6,1)<>";"THENPROCerror:
ENDPROC
 2810 B$=LEFT$(A$,8):C$=RIGHT$(A$,1):D$
=LEFT$(A$,1)
 2820 IFMID$(A$,1,1)=MID$(A$,4,1) AND M
ID$(A$,1,1)=MID$(A$,7,1) THEN2910
 2830 IFMID$(A$,2,1)=MID$(A$,5,1) AND M
ID$(A$,2,1)=MID$(A$,8,1) THEN2850
 2840 PROCerror:ENDPROC
 2850 D$=MID$(A$,2,1)
 2860 FORi%=0TO(ASC(C$)-ASC(D$))
 2870 DD%=ASC(D$)+i%:E$=CHR$(DD%)
 2880 B$=LEFT$(B$,1)+E$+MID$(B$,3,2)+E$
+MID$(B$,6,2)+E$
 2890 PROCsummate(B$)
 2900 NEXT:ENDPROC
 2910 FORi%=0TO(ASC(C$)-ASC(D$))
 2920 DD%=ASC(D$)+i%:E$=CHR$(DD%)
 2930 B$=E$+MID$(B$,2,2)+E$+MID$(B$,5,2
)+E$+MID$(B$,8,1)
 2940 PROCsummate(B$)
 2950 NEXT:ENDPROC
 2960 :
 2970 DEF FNinput(X%,Y%,maxlen%,type$)
 2980 LOCALcode%,input$,length%,OK%,pnt%
 2990 input$="":length%=0:pnt%=FALSE
 3000 PRINTTAB(X%,Y%);SPC(maxlen%):PRIN
TTAB(X%,Y%);
 3010 REPEAT
 3020 OK%=FALSE:code%=GET
 3030 IF code%=127 THEN PROCdel
 3040 IF type$="S" THEN PROCalpha ELSE
PROCnum
 3050 IF length%=maxlen% THEN OK%=FALSE
 3060 IFOK%THENinput$=input$+CHR$(code%
):length%=length%+1:VDU code% ELSE VDU7
 3070 UNTIL code%=13
 3080 VDU13:NUL%=(length%=0):IF type$="
S" THEN =input$ ELSE =VAL(input$)
 3090 :
 3100 DEF PROCdel
 3110 IF length%>0 THEN pnt%=NOT(RIGHT$
(input$,1)="."):length%=length%-1:input
$=LEFT$(input$,length%):VDU code%
 3120 ENDPROC
 3130 :
 3140 DEF PROCalpha
 3150 IF code%>31 AND code%<127 THEN OK
%=TRUE
 3160 ENDPROC
 3170 :
 3180 DEF PROCnum
 3190 IF (length%=0 AND code%=45) OR (c
ode%>47 AND code%<58) OR (code%=46 AND
NOT pnt%) THEN OK%=TRUE
```

Continued on page 26



ADVENTURE GAMES

Well I made it back! It was touch and go for a while, and to be honest, at one point I thought the Troll had got me. Perhaps telling you about it will calm my nerves. It's all those adventures from EPIC Software that did it...

WHEEL OF FORTUNE (£9.95)

This is one of the better games currently available, as it incorporates nearly every feature you would expect from a modern adventure. There are teletext graphics for every location, of which there are over 200! An ample supply of accompanying text blends with the graphics to produce a traditional adventure-land of caves, trolls and magic. The land is also populated by wandering characters who can help or hinder you in your quest to recover the magic Wheel of Fortune. A further neat feature is that at some points, the game will progress without you. While you sit pondering, you may suddenly be informed of a happening elsewhere.

The game understands multiple commands such as GO NORTH AND PICK UP EVERYTHING. It also permits you to question some of the more intelligent game inhabitants (e.g. "WHERE IS THE WHEEL?"). The game also has unusually "friendly" SAVE and LOAD commands which permit you to save the game under different file names. This may seem inconsequential, but it is all these small features which point to the quality of the overall game. Additionally, commands such as CONTINUE are a nice feature, which when abbreviated to C saves you repeatedly typing the same movement commands. All together this is a highly recommended game which rightly deserves your consideration.

CASTLE FRANKENSTEIN (£7.95)

This text-only game is one of three earlier Epic games, which have been re-released with increased text. In this game your village is being terrorised by the noises coming from the nearby ruin of Castle Frankenstein. You being a brave citizen, volunteer to go and have a look - stupid boy!

Having taken half an hour to figure out how to light my lamp I was in just the right mood to settle any monster's hash! Without a pause for breath, I set off towards the castle, ready to smash ten bells out of any trouble-maker! On entering the wood I found an elderly gentleman with his neck snapped, and his violin smashed to matchwood - I retired to the village to have a nice sit down!

From all the tunnels I have found beneath the village and castle, I suspect the Baron has created a mad, six-foot rabbit! I never did find the monster as I decided that with two more games to review, and as I have a very slim neck, I would leave the village by the east road, very quietly! The game has many good points and the extra text has been useful in creating the right atmosphere of menace.

THE QUEST FOR THE HOLY GRAIL (£7.95)

With an eye to the title of this game I did a "FUNNY-WALK EAST", and set off hoping to find the Monty Python team. Deep in the forest, under the shadow of Camelot itself, I found the first of the bad-tempered knights which infest this game. Unlike the Python team, these characters don't have a sense of humour. They appear to have one purpose, and that is to separate your head from your shoulders.

While searching the forests, swamps and castles for the elusive Grail, you will encounter dragons, witches, multi-coloured knights and distressed

damsels. Magical fruits are to be found which, if eaten at the correct time, can help you in your quest to become a Knight of the Round Table. I found the majority of the game relatively easy but there are a few nasties which required a sneaky peek at my Hints and Spells book. The game permits you to be resurrected a number of times, should your sword fail to save you from the perils of the forest.

THE KINGDOM OF KLEIN (£7.95)

An aura of doom surrounds the Klein Kingdom. The Magic bottle was removed from the pedestal in the King's palace and stolen by the Wicked Witch of the Mountain. She swore that she would put a terrible curse on anybody foolish enough to try to recover it. Enter foolish hero stage left! You must destroy the Witch and recover the magic Klein bottle.

Full sentence, as well as the customary two word commands are accepted in this 230 location, machine code adventure. The tale ranges over mountains, lakes and caves. I spent some considerable time getting a particular door to open, and as I stood congratulating myself, it closed again with the message "TOO SLOW". I didn't laugh! I dont know what other time dependant traps there are, but next time I'll leap and then look. I have played an earlier version of Klein which again had less text and was more cryptic and deadly. This new version is an improvement, and the additional clues for the dimwitted helped me immensely.

I tried fighting the giant using my rolled up umbrella, and he almost died laughing - but not quite! I think I'll leave you to find and kill the Witch as my sword arm and brain are both becoming very numb.

MAGIC SPELL

As I mentioned in an earlier article, it is possible to remove those magic characters from commercial programs which blank the screen when you attempt to list them. This month's spell is a technique to insert these characters in your own programs. By this means you may prevent players looking at the data statements within your adventures. The trick involves scattering REM statements which contain the characters &0C and &15, throughout your program (See User Guide page 507 for an explanation of these).

By placing the characters within a REM statement they will have no effect on your program, but they will effect any attempt to list them. The problem is how to insert these characters, as there is no key on the keyboard which can be used. The method is to insert a REM statement of the correct size (e.g. REM##) and then run a small procedure to change the hashes to the desired codes.

As an example type in the following program and then run it. The procedure MAGIC will step through the program from the beginning (PAGE) to the end (TOP). Wherever REM## is found, the hash characters will be changed to &0C followed by &15.

```
10REM##
20PROCMAGIC
30END
40DEFPROCMAGIC
50M=PAGE
60FOR X=1 TO TOP-PAGE
70IF M?X<>&F4 GOTO 100
80A%=X+1:B%=X+2
90IF M?A%=&23 AND M?B%=&23 THEN
  ?(M+A%)=&0C:?(M+B%)=&15
100NEXT
110ENDPROC
```

Should you now attempt to list the program, you will find that the screen will blank and the keyboard lock up. Pressing Break and typing OLD will recover the situation. To change the nasty characters back to hashes, list line 90 and change it to:

```
90IF M?A%=&0C AND M?B%=&15 THEN
  ?(M+A%)=&23:?(M+B%)=&23
```

Now re-running the program will reverse the procedure. Should you now wish to incorporate this technique into your own game, insert the procedure DEFPROCMAGIC at the end of your program then scatter REM## statements throughout the remainder. The command PROCMAGIC should now be given, and upon its completion procedure MAGIC may be deleted.

COMMUNICATE WITH YOUR BEEB

Five Modems reviewed by Ben Miller-Smith

Using your Beeb to communicate with Prestel, Bulletin Boards, or even just your like-minded friends, is a fast growing interest of many micro users. Ben Miller-Smith, a consultant in computer hardware and software systems, looks at some of the versatile modems now on the market and reports on his findings.

The data communication world abounds in abbreviations and jargon which cannot be avoided. For useful background information see "Modems and Bulletin Boards" in Beebug Vol. 3 No. 5 (Oct.'84) which explains much of the terminology. For simplicity in this review all references to bulletin board operation imply operation at 300/300 baud to V21 standards. All the modems under consideration have been tested as fully as possible under typical working conditions. In particular I would like to thank Beebug contributor Peter Rochford for his help at the 'far end' of several lengthy data-transfer sessions.

BEFORE YOU BUY

Before installing a modem you must have, or provide:

a) A modern (Type 600) British Telecom telephone socket near the computer.

b) In most cases, at least one spare sideways ROM socket in the computer.

c) For accessing Prestel/Micronet, Telecom Gold, etc, etc, a subscription to that service.

THE MODEMS

Five modems have been reviewed for this report, as listed below. The price range shown indicates the approximate cost for a basic modem only, up to a full configuration system which includes all useful options, software, cables, etc. In practice, it is highly desirable to have appropriate software in a sideways ROM, and where specific software is not available from the modem supplier the installation of a general-purpose communications ROM (e.g. Pace's Commstar) is assumed, at a cost of some £34.

Modem	Price range	Facilities
Watford Electronics 'Modem 84'	£78.20 - £111.55	Pr/Rs
Pace 'Nightingale'	£136.85 - £159.85	Pr Bb OA Bl/Rc
Miracle Technology 'WS 2000'	£148.35 - £241.44	Pr Bb OA Bl/Dt+Rc AA ADa Cc
Loco Systems 'SCM-100'	£149.95 - £155.90	Pr Bb OA Bl Rs Cc AA AD
Tandata 'TM 200'	£198.95 - £218.50	Pr Cc ADa OA Bb/Dt

Items after a '/' are at extra cost. Prices include VAT.

Key to facilities:

AA	Auto answer facility	Dt	Special disc/tape s/ware available
AD	Auto dial (ADa - with audio output)	OA	Originate/Answer control
Bb	Bulletin board (V21) operation	Pr	Prestel (V23) operation
Bl	Bell (USA)/CCITT (UK) control	Rc	Use of 'Commstar' ROM assumed
Cc	Full computer control of operation	Rs	Special ROM S/ware available

Note: With the exception of the Tandata TM 200 all the modems provide a telephone socket for your 'displaced' phone - the phone will remain operable at all times, except when the modem itself is actually switched 'on-line'.

WATFORD ELECTRONICS MODEM 84

A basic and low cost Prestel-only modem.

The basic modem (£78.20 including cable to BBC) is housed in a cheap but neat black plastic box with the front panel having three LED's (Power, Data Carrier Detect, and On-line) and two switches (on-line, and mode - Prestel or user to user), and is mains powered. The rear panel provides a telephone socket, a DIN socket for the computer interface cable, a 2.5 metre cable terminated in a telephone plug, and the mains lead entry point. The internal and external construction and appearance is satisfactory, which is rather more than can be said for the accompanying documentation.



The Modem 84 manual is written presupposing that the Watford Electronics Modem 84 (Prestel) ROM has been purchased - the total package cost is then some £94.30. The installation instructions curiously refer to connecting a non-existent ribbon cable to the BBC user port, while the manual is full of references to using the £ sign as the equivalent to # (a character much used in accessing Prestel) - it doesn't work, use Return instead. Most of the manual is dedicated to instructions and explanations on using the modem and ROM software for accessing Prestel - to this end the modem, Prestel ROM software and manual all proved satisfactory. The function keys provide easy selection of a variety of options, including mailbox handling, downloading programs (e.g. from Micronet), dumping screens to an Epson (or compatible) printer, saving and recalling pages to or from disc (or tape), and all the standard Prestel facilities.

At extra cost a second ROM is available to provide user-to-user half-duplex communication at 1200 bps - the Prestel ROM has to be present, so the total package cost is then £115.55. This ROM is primarily intended to be used to transfer data files at reasonable speed to (or from) a second computer, which must be fitted with the same Watford ROM in order to exploit the facilities to the full. This facility would mainly be of interest if you have a need for such communication between two sites, and can ensure that both are similarly equipped.

VERDICT

A cheap and simple V23 modem for use on Prestel. Recommended configuration should include the Prestel ROM. The user-to-user ROM option will be of limited value to most people.

THE PACE NIGHTINGALE

A flexible general purpose unit with manual controls only.

The Nightingale modem provides both V21 and V23 communication facilities with front panel option switches to control selection of the mode, originate/answer, and CCITT/Bell standards. It is housed in a dark brown plastic case, with a yellow front panel having two LED's (Power/Data - dual colour, and Carrier Detect) and four push-button switches to control the mode of operation. The labelling of the switches is a bit 'busy', and it is rather difficult to tell at a glance



what modes have been selected. The rear panel provides a telephone socket, a computer interface DIN socket, a self-test push button, a lengthy (3 metre) telephone cable and plug, and entry for the mains lead. The modem's construction appears neat, tidy and satisfactory.

Not surprisingly, Pace recommends the use of its 'Commstar' ROM with the modem, and offers a discount on the price if both are purchased together. The modem instructions (13 typed A4 pages) are comprehensive and cover all aspects of installation and use. However, very little detail on actual data communication formats is included (e.g. on configuring the BBC serial port to operate on Prestel), and it is in this area that Commstar (and its manual) shine. The Nightingale + Commstar combination proved excellent in operation and in the facilities provided, as the modem can be configured to meet all reasonable requirements for operating in full-duplex modes, while Commstar provides comprehensive menu and function key driven software to take care of the details. The combination thus provides all of the data communication facilities you could reasonably wish for, provided you don't mind making (and answering) your own telephone calls and have no ambitions to set up your own bulletin board or other automatic call-handling facility.

VERDICT

The Nightingale modem provides manually selected operating modes compatible with all the common systems in use. When used in conjunction with Commstar (recommended) the system provides very satisfactory data communication facilities at a reasonable price, although without the possibility of future enhancements for automatic call handling.

MIRACLE TECHNOLOGY WS 2000M

Versatile hardware, with many extra-cost enhancement options.

The WS 2000 is housed in a remarkably small plastic case, coloured and textured to match the BBC computer. The black front panel contains five LED indicators and three rotary switches, all clearly labelled. The rear panel



provides a 25-way D-socket for the computer interface, a 3 metre cord and plug for the telephone connection, a telephone socket, a ribbon cable connector, the mains lead entry and an on/off switch.

The basic modem (at £156.40 including BBC interface cable) provides manually-controlled full-duplex Prestel (normal and inverse data rates) and bulletin board operation, and additionally supports half-duplex operation at 600 and 1200 bps with Transmit/Receive switching controlled either manually or by the RTS (request to send) signal on the RS423 interface. These many options could make the unit tricky to set up, but the well-labelled front panel and comprehensively illustrated manual define all the options clearly. The manual covers installation, operation, self-testing and fault finding very thoroughly, and provides technical details on the hardware, user port and serial interface signals for those wishing to write their own data communication software.

Several extra-cost options are available for the modem, as listed below:

Auto Dial Card + Cable.	£39.10
Auto Answer Card (not tested).	£34.50
Disc software for Auto-dial. **	£11.44
Remote Control via user port.	£16.10

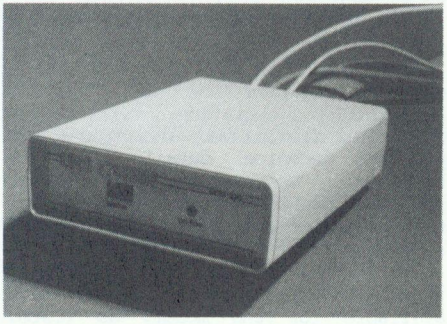
** 'Softdrive' assumes Commstar ROM fitted (about £34 extra).

The documentation of the Auto Dial and Remote Control options provides detailed installation instructions, and technical descriptions of the interfaces and dialling operations, but little else, while the software

contained some errors which Miracle Technology have now corrected. Although it would be possible (but tricky) to write your own Auto Dialling and Remote Control software from the information provided, it is strongly recommended that the 'Softdrive' option is used as it provides simple and effective menu driven facilities for accessing Prestel (including automatic calling, logging on and page selection if you wish), while the bulletin board section allows the creation of a disc database of useful numbers (and their associated operating modes), with automatic calling of a selected number. The system assumes that a Commstar ROM is fitted to the computer, and 'Softdrive' initialises and calls Commstar to handle the actual data communications when a call has been established.

VERDICT

The basic modem with manual control is the most flexible and versatile of those reviewed, and has facilities to add extra-cost options for full automatic call handling in both directions.



LOCO SYSTEMS SCM-10
A full-specification modem at a remarkable price.

The small grey plastic box with no control and only one ('on-line') LED conceals a wealth of facilities, including V21 and V23 operation, Auto Dial, Auto answer, and originate/answer and CCITT/Bell switching, all in the basic price (£149.95). Four cables emerge form the rear: a 2 metre telephone cable and plug, a 1 metre serial interface cable and DIN plug, a 1 metre ribbon cable with user port connector, and a 1 metre power lead and plug to fit the BBC auxiliary power socket - a Modem Controller ROM ('Modrom') is provided for insertion in a BBC sideways ROM socket, while a telephone may be plugged into a socket on the front panel.

The only extra-cost (£5.95) option available (or necessary) is an adaptor for the BBC power outlet, essential if you are already powering discs or other equipment from the computer. All the modem's facilities and options are selected and controlled by the Modrom software (there are no manual controls at all), and these are all implemented as extra * commands, usable from Basic or other languages - for example, *MODEM A 300 UK would configure the modem to operate in Answer mode at 300 bps to United Kingdom (V21) standards, while *AA A 1200 UK would prime the system to Auto Answer any incoming call in Prestel mode - if the command parameters are omitted then the modem will automatically cycle between possible modes when answering (or initiating) a call. If a valid carrier is received the SCM-100 will settle into an appropriate mode according to the type of incoming carrier and the *STATUS command will tell you what it is, if necessary. The modem is constructed to reasonable standards (although strain relief on the cables could be improved), while the documentation covers installation (including the ROM), testing, all the available commands and their functions, and the use of the default Prestel and Terminal modes to access Prestel and bulletin boards - in these modes the function keys provide a useful range of commands, but you will have to make your own function key strip as none is provided.

VERDICT

Good value for money if you want a wide range of facilities: the Modem plus Modrom combination offers simple operation for those that want it, while data communication experts will appreciate the many extra features and will exploit them to the full. For Telesoftware and other file transfer applications you will need Commstar (or equivalent) as well.

TANDATA MARKETING TM 200
Mainly designed for Prestel, V21 operation tricky.

Housed in a wide but slim-line black plastic case, this modem comes complete with Auto Dial facilities and an internal battery-backed memory that can store up to eight telephone numbers and, if you wish, Prestel or other customer indentities and passwords. An optional (but almost essential) BBC Micropack is available which provides a modem to BBC serial interface lead, and modem-control software ('Tango') on tape (transferable to disc) - however, Tango primarily supports Prestel operation only, and is of little help for general purpose communication with V21 bulletin boards. The modem has only two LEDs on the front panel ('Power' and 'On-Line'), while the rear panel provides a 3 metre line connection with plug, a DIN serial interface connection, the power lead (from a separate plug-in power module), and two unlabelled switches. A serious and inconvenient omission is the absence of a 'replacement' telephone socket - at the very least you will need a double-outlet adaptor if you wish to keep your telephone operational while the modem is plugged in.

All modem functions are controlled by special codes sent over the serial link, rather than by a separate user port connection. Using the facilities in the Tango software it proved easy to set up the Auto Dial function and access Prestel, and modem performance was satisfactory. Tango is function key driven (also using the Copy, Delete, Control and cursor keys) and provides all the facilities you could reasonably wish for while accessing Prestel: key f0 provides a 'Help' display though its clarity and format leaves a lot to be desired - the Micropack manual optimistically states 'The user will soon become familiar with the layout of the display - for 'soon' read 'eventually'. However, the manual is clear and comprehensive, and covers all aspects of Tango's operation when accessing Prestel. Unfortunately the same cannot be said of the other, more general, modem documentation when it comes to V21 (300/300, bulletin board) operation. Nine typed A4 pages define the control sequences necessary for communication with the modem (which has its own internal microprocessor): not only is the operation somewhat devious, but various timing restrictions apply on the rate at which characters can be sent. Selection of V21 operating mode is by the rear panel switches (whose layout bears little resemblance to the sketch in the documentation) and only competent programmers and data communication enthusiasts are likely to be able to write satisfactory software to control and use the V21 mode. Because of the control codes used it is difficult to use Commstar (it intercepts and gets confused by some codes), and this reviewer was unable to use the modem in V21 mode for other than simple tests. It is to be hoped that Tandata will publish extended versions of Tango that permit more general purpose use.

VERDICT

A satisfactory (but expensive) Auto Dial modem that operates well on Prestel, provided the BBC Micropack is used. V21 operation is tricky and complex, and not suitable for beginners. Bear in mind the lack of a telephone socket for your 'displaced' telephone.

BRITISH TELECOM (BTAB) APPROVALS

All the (basic) modems carry BTAB Approval with the exception of the Loco Systems SCM-100 for which approval has been requested. The WS 2000 is not at present approved if the enhancements (Auto Dial, etc.) are fitted.

Addresses:

Watford Electronics Limited,
250 High Street, Watford,
Herts. Tel: 0923 40588/37774.

Pace Software Limited,
92 New Cross Street,
Bradford, W.Yorks.
Tel: 0274 729306.

Miracle Technology (UK) Limited,
10-12 St.Peters Street, Ipswich,
Suffolk. Tel: 0473 50304/5.

Loco Systems, P.O.Box 9,
Twickenham, Middx. Tel: 048 624 480.

Tandata Marketing Limited,
Albert Road North, Malvern,
Worcs. Tel: 06845 68421.

PRETTY LIST

STRUCTURED LISTINGS OF BASIC PROGRAMS

The LISTO7 command in Basic is very helpful for producing program listings formatted to show the structure of the program, but it does have its limitations. This utility by Peter Hayes goes much further, including the complete reformatting of muti-statement lines.

One of the features of BBC Basic is the extent to which it allows the programmer to write well structured programs, more so than most other versions of Basic. However, the Beeb's limited memory often forces a program to be written in a style which obscures this structure, resulting in programs which are difficult to read. Although the LISTO7 command can be used when listing programs to provide a more structured result, it is quite limited in what it can achieve. 'Pretty List' is a useful utility for producing much better structured listings than LISTO7 alone, and will enhance the appearance of your programs considerably.

The features of this program are:

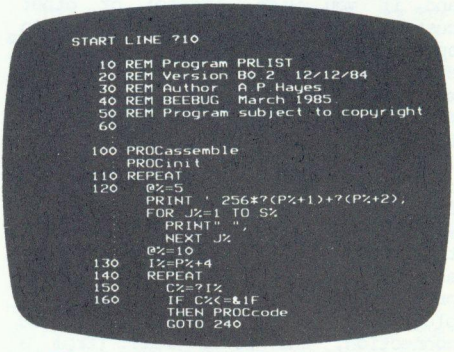
1. Indentation of statements in the same way as LISTO7.
2. Splitting of multi-statement lines.
3. Splitting of IF-THEN-ELSE onto separate lines.
4. Displaying the hex codes of non-printable characters.

To start with, you will need to type the program in as listed and save it to cassette or disc (be careful when typing in the machine code section from lines 1430 to 1620). The PRLIST program can sit anywhere in memory, but for simplicity we shall refer to one method of using this program. Load the Basic program to be listed, and then type:

```
PRINT ~PAGE  <Return>
PAGE=TOP+&100  <Return>
LOAD "PRLIST"  <Return>
```

and then run the program PRLIST. This will ask you four questions. The first question is about the PAGE value of your Basic program. This can be obtained from the first command that you typed above (normally &E00 for cassette and &1900 for disc systems).

The computer will then ask you for a start line. This is the line number that you wish to start the formatted listing from. You are then asked if you wish to output codes as numbers. This refers to all teletext control codes



```
START LINE ?10

  10 REM Program PRLIST
  20 REM Version B0.2  12/12/84
  30 REM Author  A.P.Hayes
  40 REM BEEBUG  March 1985
  50 REM Program subject to copyright
  60
 100 PROCassemble
     PROCinit
 110 REPEAT
 120   @%=5
       PRINT ' 256*?(P%+1)+?(P%+2),
       FOR J%=1 TO S%
         PRINT" ",
         NEXT J%
       @%=10
 130   I%=P%+4
 140   REPEAT
 150     C%=?I%
 160     IF C%<=&1F
         THEN PROCcode
         GOTO 240
```

and VDU codes below 32. If you wish to display (or print) the codes as spaces then answer 'N' to this question. If you wish to print the codes as condensed characters (Epson based printers) or display them as 2 digit hex bytes on the screen then answer 'Y'.

The final question will ask if you require output to the printer or to the screen alone. If you wish to display the program only on the screen, then the computer enters 'paged mode' which requires you to press the Shift key to continue listing the program.

Using this utility will enable you to produce program listings with a well structured format even though your working version may be highly condensed. It won't, of course, put back spaces that may have been 'compacted' out, but it will make your programs much easier to read and understand.

Program Notes

The program is designed to produce a printout to an Epson (or compatible) printer, using condensed mode for any control characters as described above. The printer control codes for this are contained in lines 1210 and 1260. They can be readily changed to suit other printers (or omitted altogether, though this may well produce a slightly less readable program).

The outer loop of the main program runs from line 130 to 300 and controls the processing of a full line. The second loop, starting at line 160, scans each character of a line and takes appropriate action.

The procedures called are:

PROCassemble. This sets up the machine code to deal with tokenised Basic keywords. This procedure searches the Basic keyword table for a token to match the number stored in A% and then prints the keyword.

PROCinit. This sets up the global variables.

PROCnum. This prints out the line number.

PROCstr. This copies strings allowing for special characters.

PROCcode. This outputs the hexadecimal value of any special codes. If you have answered yes to both the printer and display code options it will set condensed printing to print the codes.

```
   10 REM Program PRLIST
   20 REM Version B0.4
   30 REM Author  A.P.Hayes
   40 REM BEEBUG  March 1985.
   50 REM Program subject to copyright
   60 :
  100 CLS:VDU15
  110 ON ERROR GOTO 1640
  120 PROCassemble:PROCinit
  130 REPEAT
  140 @%=5:PRINT ' 256*?(P%+1)+?(P%+2);
:FOR J%=1 TO S%:PRINT" ";:NEXT J%:@%=10
  150 I%=P%+4
  160 REPEAT
  170 C%=?I%
  180 IF C%<=&1F THEN PROCcode:GOTO 270
  190 IF C%=&22 THEN PROCstr:GOTO 270
  200 IF C%=&3A THEN PRINT'"     :";:FOR
 J%=1 TO S%:PRINT" ";:NEXT J%:GOTO 270
  210 IF C%<&80 THEN PRINT CHR$(C%);:GO
TO 270
  220 IF (C%=&8B) OR (C%=&8C) THEN PRIN
T '"     ";:FOR J%=1 TO S%:PRINT" ";:NE
XT J%:GOTO 260
  230 IF C%=&8D THEN PROCnum:GOTO 270
  240 IF (C%=&E3) OR (C%=&F5) THEN S%=S
%+2
  250 IF (C%=&ED) OR (C%=&FD) THEN S%=S
%-2
  260 A%=C%:CALL token
  270 I%=I%+1
  280 UNTIL I%>P%+?(P%+3)-1
  290 P%=P%+?(P%+3)
  300 UNTIL ?(P%+1)=&FF
  310 PRINT:VDU3
  320 END
  330 :
 1000 DEF PROCnum
 1010 LOCAL n%
 1020 n%=(?(I%+3) AND &3F)*256+(?(I%+2)
 AND &3F)
 1030 IF (?(I%+1) AND &20)=&20 THEN n%=
n%+128
 1040 IF (?(I%+1) AND &10)=0 THEN n%=n%
+64
 1050 IF (?(I%+1) AND &4)=0 THEN n%=n%+
16384
 1060 PRINT STR$(n%);
 1070 I%=I%+3
 1080 ENDPROC
 1090 :
 1100 DEF PROCstr
 1110 REPEAT
 1120 IF (C%<=&1F) OR (C%>=&7F) THEN PR
OCcode ELSE PRINT CHR$(C%);
 1130 I%=I%+1:C%=?I%
 1140 UNTIL (C%=&22 AND ?(I%+1)<>&22) O
R C%=&0D
 1150 IF C%=&22 THEN PRINT """";
 1160 ENDPROC
 1170 :
 1180 DEF PROCcode
 1190 IF OC%=0 VDU32:ENDPROC
 1200 LOCAL i%,c%,c$,h%:c%=C%:c$=""
 1210 PRINT CHR$(1)CHR$(15);
 1220 FOR i%=1 TO 2
 1230 h%=48+c% MOD 16:IF h%>57 THEN h%=
h%+7
 1240 c%=c% DIV 16:c$=CHR$(h%)+c$
 1250 NEXT
 1260 PRINT c$;CHR$(1)CHR$(18);
 1270 ENDPROC
 1280 :
 1290 DEF PROCinit
 1300 LOCAL l%,f%
 1310 S%=1
 1320 INPUT'"PAGE value of program =&"P
G$
 1330 P%=EVAL("&"+PG$)
 1340 INPUT'"Start line ",f%
 1350 l%=256*?(P%+1)+?(P%+2)
 1360 IF l%<f% THEN P%=P%+?(P%+3):GOTO
1350
 1370 INPUT'"Output codes as numbers ",
C$
 1380 IF LEFT$(C$,1)="Y" OC%=1 ELSE OC%
=0
 1390 INPUT'"Output to printer ",P$
 1400 IF LEFT$(P$,1)="Y" VDU2 ELSE VDU14
 1410 ENDPROC
 1420 :
 1430 DEFPROCassemble
```

```
 1440 IF ?&806D=65 offset=&D ELSE offse
t=&11
 1450 DIM SP 100:FORZ=0 TO2 STEP2
 1460 P%=SP:[OPTZ
 1470 .token
 1480 STA &70:LDA#&61:STA&71:LDA#&80:ST
A&72
 1490 LDY#offset:.loop:LDA(&71),Y
 1500 CMP&70:BEQ found
 1510 CLC:LDA&71:ADC#1:STA&71
 1520 LDA&72:ADC#0:STA&72:CLC
 1530 JMP loop
 1540 .found:DEY:LDA(&71),Y
 1550 CLC:CMP#&80:BCC found:CLC
 1560 CMP#&8A:BEQ print:INY
 1570 .print:INY:LDA(&71),Y
 1580 CLC:CMP#&7F:BCS end
 1590 JSR&FFEE:JMP print
 1600 .end:RTS:]
 1610 NEXT
 1620 ENDPROC
 1630 :
 1640 ON ERROR OFF
 1650 IF ERR<>17 THEN MODE 7
 1660 REPORT:PRINT" at line ";ERL
```

Continued from page 16

```
 3200 IF code%=46 AND OK% THEN pnt%=TRUE
 3210 IF type$="I" AND pnt% THEN pnt%=F
ALSE:OK%=FALSE
 3220 ENDPROC
 3230 :
 3240 DEFPROCinit
 3250 B%=65:M%=0:Dfile$=""
 3260 cursor$=CHR$136+CHR$137+CHR$138+C
HR$139+CHR$13
 3270 DIMmat(25,25),col$(25),row$(25),S
pec$(200),edit%(25)
 3290 INPUT"Enter No.cols(max 26)- "x%:
IFx%>26 THEN 3290
 3300 INPUT"Enter No.rows(max 26)- "y%:
IFy%>26 THEN 3300
 3310 z%=x%*y%:col%=0:row%=0
 3320 FORI%=0TOx%-1:edit%(I%)=&020006:N
EXT
 3330 maxrow$=CHR$(y%+64):maxcol$=CHR$(
x%+64)
 3340 IF y%>7 THEN r%=7 ELSE r%=y%
 3350 IF x%>5 THEN c%=5 ELSE c%=x%
 3360 ENDPROC
 3370 :
 3380 DEFPROCde
 3390 CLS:PRINT"Index"
 3400 REPEAT
 3410 I%=FNinput(POS,VPOS,3,"I")
 3420 IF NUL% OR I%>M% THEN 3450
 3430 PRINTTAB(POS+10,VPOS);Spec$(I%)
 3440 FORJ%=I% TO M%-1:Spec$(J%)=Spec$(
J%+1):NEXT:M%=M%-1
 3450 G$=FNcont(0,20,"Space to continue
 - Return to exit",CHR$32+CHR$13)
 3460 UNTIL G$=CHR$13
 3470 ENDPROC
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

BUG IN USE OF STRING INDIRECTION OPERATOR - D. Morgan

When using the Basic indirection operator '$' any attempt to access a string in memory of over 255 characters in length will result in Basic substituting a null string as the result (but the string in memory is left untouched).

OVERCOMMING CASSETTE DATA ERRORS - T. Thomson

When using *EXEC, INPUT#, or BGET# some cassette recorders can overrun into the next block of data when the motor is switched off after each block. This shows up as 'Block?' and 'Data?' errors. If your tape recorder is susceptible to this you can increase the inter-block gap using *OPT 3,n when recording the data, where n is the time required between blocks in tenths of second. You will have to experiment a little to find the value best suited to your recorder. See the User Guide page 398 for more details.

*RUN

Although not mentioned in the user guide, the abbreviation for *RUN is */. You cannot *RUN a Basic program, although *RUN (filename) will load it into memory. You have to OLD it first. If you *RUN a file create with *SPOOL it will operate as though you had *EXECed it.

MAKING MUSIC ON THE BEEB (Part 2)

Following on from last month's article, Ian Waugh looks at some of the less-common aspects of the SOUND parameters as a prelude to the programming of multi-part tunes.

If you've dabbled with the SOUND command even a little (and read the relevant sections in the User Guide) you will know that it is normally followed by four parameters:

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

C is the Channel number (Ø to 3), A is the Amplitude (-15 to Ø, or 1 to 16 under envelope control), P is the Pitch (Ø to 255) and D is the Duration (Ø to 255). The User Guide explains this quite well. It also states that the first parameter, C, can be expanded like this:

```
SOUND &HSFC,A,P,D
```

These parameters are not so readily understood and their purpose not always as obvious as the usual C, A, P and D parameters. Their ranges are as follows:

H - Hold:	Ø or 1
S - Synchronisation:	Ø to 3
F - Flush:	Ø or 1
C - Channel:	Ø to 3
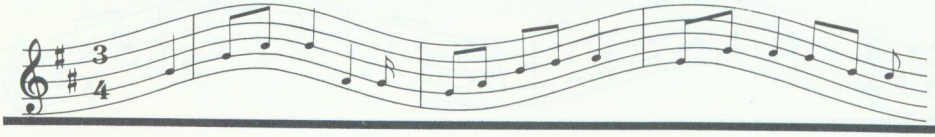
Let's have a look at them and see what they do. The first thing to notice is that they are preceded by an ampersand (&). This is to inform the Beeb that the following numbers are hexadecimal (see page 71 of the User Guide). As the values used are only in the range Ø to 3, we can forget about this for practical purposes as long as we don't forget to include the ampersand.

H FOR HOLD

This is described in the User Guide on pages 187, 350 and 352 and is probably the least-understood of all the SOUND parameters, which is hardly surprising as it is described inaccurately on the first two occasions and all explanations tell you what can be done with the instruction, not what the instruction actually does. The default value is Ø which allows the SOUND command to operate as normal. With H set to 1, the amplitude and pitch parameters are ignored and only the duration is obeyed. What use is it? By itself, it will create a silence or a rest, but if it follows a note on the same channel which is under envelope control then the previous note may be extended. To understand this we really need to understand the ADSR (Attack, Decay, Sustain, Release) phases of the ENVELOPE command (see pages 180 to 187 of the User Guide).

Without envelope control (i.e. when A lies between -15 and Ø) the duration parameter 'D' controls the exact length of each note. Under envelope control, D only determines the length of the Attack, Decay and Sustain phases. If no other commands follow on the same channel, the sound will continue into the Release phase. Normally, when playing a tune, another note will follow immediately so the release phase is rarely heard. A SOUND command with H set to 1 produces a 'dummy note', as it is called in the User Guide, for the duration of D. This will extend the previous note by the value of D and allow its release phase to occur. Note, setting H to 1 does not automatically allow the release phase to be completed but simply allows it to continue.

Misunderstandings have arisen because the User Guide states, incorrectly, on page 187 that the amplitude, pitch and duration are ignored, leading the reader to believe that this command calculates the release time on the previous note and



allows it to play out. The description on page 350 is ambiguous and other writers have drawn the wrong conclusion. This short program illustrates the Hold option:

```
  10 REM PROGRAM 4.1
  20 REM HOLD PARAMETER DEMONSTRATION
  30:
  40 ENVELOPE1,4,0,0,0,0,0,0,126,-1,0,
-1,126,60
  50 TIME=0
  60:
  70 SOUND1,1,53,40
  80 PROCTime
  90 SOUND1,1,69,20
 100 PROCTime
 110 END
 120:
 130 DEF PROCTime
 140 REPEAT
 150 IF TIME/100=INT(TIME/100) PRINT T
IME/100
 160 UNTIL TIME>800
 170 STOP
 180 ENDPROC
```

This is merely for demonstration purposes so I hope the purists will forgive the STOP in the middle of a procedure. The intention was to clarify exactly the main program which lies between lines 70 and 100. PROCTime begins after the SOUND commands have been issued and prints to the screen approximately every second. Run the program. You should find that the sound produced by line 70 lasts four seconds. Its D value is only 40 which means that the attack, sustain and decay lasts for two seconds (40 twentieths), the other two seconds being the release phase. Remove line 80 and you will hear the note produced by line 90 after two seconds indicating that the release phase of the first note was cut short. Enter another line 80:

```
  80 SOUND&1001,0,0,40
```

The first note will now last four seconds - its own two seconds and the extra two seconds afforded it by the dummy note in line 80. If you increase the duration in line 80 you will hear the ADS and R phases complete and then there will be a silence while the dummy note runs its time before the note at line 90 sounds. Experimenting with this short program should enable the Hold parameter to be correctly understood.

S FOR SYNCHRONISATION

This parameter allows two or more notes to sound at exactly the same time. With S equal to 0, the default value, the notes are queued as usual and are sounded as soon as they arrive at the front of the queue. If S is 1, 2 or 3 then that note does not sound until another note or notes with the same S value appear at the front of the queue on other channels. If S is 1, the computer waits for one more note, with S set to 3 it waits for notes on all four channels. This is quite well documented in the User Guide but it is not always obvious how to use it. As a means of sounding notes at the same instant is this:

```
10 SOUND1,-15,73,30
20 SOUND2,-15,89,30
30 SOUND3,-15,101,30
```
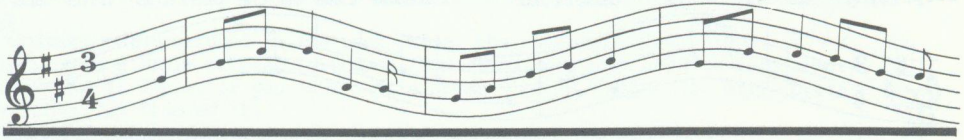
so different from this?

```
10 SOUND&0201,-15,73,30
20 SOUND&0202,-15,89,30
30 SOUND&0203,-15,101,30
```

The value of this command, however, lies in the fact that it allows us to execute other statements between sounds without throwing our sound out of synchronisation or 'sync' as it is often referred to. Insert this line in the two previous examples and observe the result:

```
15 FOR delay=1 TO 400:NEXT
```

When programming multi-part tunes I normally use 'sync' to keep the channels together. It is not always necessary, especially in short pieces, but it enables the computer to do other



things while the music is playing - a sort of 'interrupt from Basic' if you like, but very effective. We'll look at such a program in the next article.

F FOR FLUSH

With F set to 1, the SOUND statement in which it occurs is immediately executed. It flushes the buffer of any notes already in the queue and stops any note which may be sounding at the time, as the next program demonstrates:

```
10 REM PROGRAM 4.2
20 REM CHANNEL FLUSHING DEMONSTRATION
30 SOUND3,-10,5,245
40 SOUND1,-15,53,60:SOUND2,-12,5,60
50 SOUND1,-15,81,60:SOUND2,-12,33,60
60 SOUND1,-15,101,60:SOUND2,-12,53,60
70 SOUND1,-15,117,5:SOUND2,-12,69,5
80 SOUND1,-15,113,60:SOUND2,-12,65,60
90 PRINT"PRESS ANY KEY TO STOP"
100 IF INKEY(400)<>-1 THEN SOUND&11,0
,0,0:SOUND&12,0,0,0:SOUND&13,0,0,0 ELSE
PRINT"TOO LATE"
110 SOUND1,-15,149,2:SOUND1,-15,137,2
120 SOUND1,-15,129,2:SOUND1,-15,137,2
130 SOUND1,-15,145,2:SOUND1,-15,137,2
140 SOUND1,-15,145,2:SOUND1,-15,149,4
```

The INKEY delay in line 100 gives you four seconds to hit a key. If you do so it will flush all channels and proceed with the sounds at line 110. If you do not press a key, the second routine will be queued in the normal way.

A FOR AMPLITUDE

A positive amplitude value of 1 to 4 will put the SOUND command under envelope control. If you do not use the RS423 and cassette output buffers then up to 16 envelopes can be defined and A can take any value between 1 and 16.

On a musical note (figuratively speaking), many factors affect the perceived volume of a sound. Reverberation, echo, vibrato and duration all tend to increase volume, as does the addition of harmonics. A sound lasting 1/100th or even 1/10th of a second will not seem as loud as a sound lasting one second. Volume also tends to alter with pitch.

P FOR PITCH

P alters the pitch in 'quarter semitone' steps, and if we want to work in conventional western notation we must work in increments of four. The diagram in last month's article (BEEBUG vol.3 no.8) shows how the values of P correspond to printed notes and the piano keyboard.

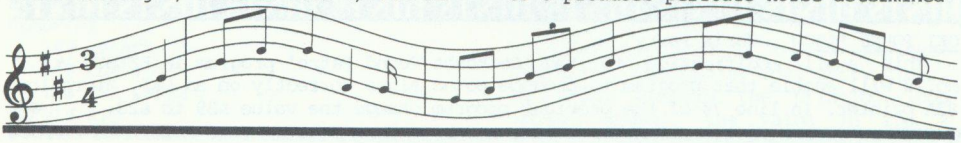
The User Guide states that a pitch value of 0 produces an A# note. This is not true as can be heard from this program.

```
10 FOR Pitch=13 TO 1 STEP-4
20 SOUND1,-15,Pitch,8
30 NEXT Pitch
40 SOUND1,-15,0,8
```

You should hear that the difference between pitch values 1 and 0 is not as great as between pitch values separated by a factor of four. It produces the same interval as that between any two adjacent pitch values, i.e. a quarter of a semitone, which is what you would expect. This being so, it would appear more sensible to use a pitch value of 0 for the lowest note and, in fact, if you measure the actual frequencies produced by the sound chip you will find that a pitch value of 0 is closer to the concert pitch of B0 than a pitch value of 1. As the 'official' pitch values are fairly well known and in common use it would only serve to confuse to use different ones. In these articles (and in my book) I use the User Guide values.

NOISE CHANNEL

The P parameter on channel 0, the noise channel, operates in a different way to the other three channels. It only takes values between 0 and 7 and produces various noises. It is explained quite well in the User Guide but an interesting use for the noise channel is in producing otherwise unobtainable low notes. Setting P equal to 3 produces periodic noise determined



by the pitch setting of channel 1. Try this:

```
10 FOR Pitch=200 TO 1 STEP-4
20 SOUND1,0,Pitch,10
30 SOUND0,-15,3,10
40 NEXT Pitch
```

That SHOULD sound like a reasonable semitone scale, but the relationship between the pitch on channel 0 and the pitch on channel 1 does not seem to be quite musically accurate. The diagram indicates the pitch values for channel 1 which produce the indicated pitch on channel 0. These continue downwards from the octaves listed in last month's article.

Figure 7.2

	C	C♯	D	D♯	E	F	F♯	G	G♯	A	A♯	B
0	144	148	152	156	160	164	168	172	176	180	183	188
-1	96	100	104	108	112	116	120	124	128	132	136	140

The relationships between these two pitches may not be the same on your computer but they should only deviate by one figure. The only anomaly appears to be A#0 but check this on your own computer. The notes so produced may not be exact octaves but they should be near enough for use in a composition. You will hear that as the pitch drops it begins to sound more like a buzz although it is arguably useful right down to octave C-1.

D FOR DURATION

This sets the length of time a note sounds in twentieths of a second. With a value of 255 the sound will continue indefinitely until stopped by pressing Escape or by flushing the channel. With a value of 0, the note is given a duration of one two-hundredth of a second. This can be very useful for separating notes of the same pitch. For example:

```
10 SOUND1,-15,53,20
20 SOUND1,-15,53,20
```

will sound as one note with a total duration of two seconds. If you insert:

```
15 SOUND1,0,0,0
```
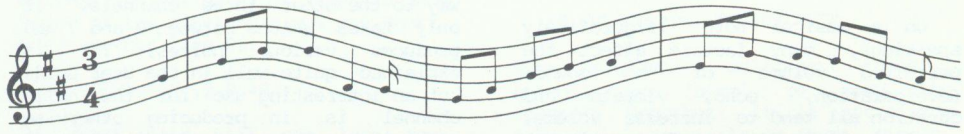
this will cause a slight separation between the notes.

IMPROVING THE SOUND OUTPUT

The sound output from the Beeb's speaker leaves a lot to be desired although it is substantially better than the output of most other micros. It is an easy matter to take an extension from the internal speaker. It is connected to the main board via plug PL15 and you need to remove the case and keyboard PCB to reach it. One such method was given in BEEBUG Vol.2 No.7 p.19, and there are a few commercial plug-in-and-go modules available. I can not recommend some type of external speaker too strongly - it really is a substantial improvement over the internal speaker and you will hear sounds and music as you have never heard them before. Tampering with the insides of your computer, however, may invalidate your guarantee so seek expert advice if in doubt.

Finally, if it all becomes a bit too much you can switch off the sound with *FX210,1 and on again with *FX210,0.

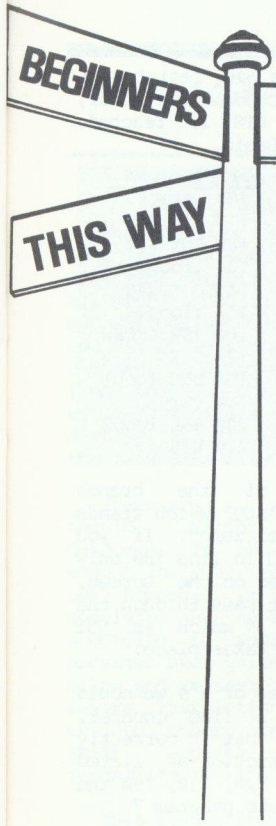
The figure and programs are from Making Music on the BBC Computer by Ian Waugh, published by Sunshine Books at £5.95 and used with kind permission of the publishers.



HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

OKI POUND PATCH - Chris Parry

This small modification to the 'Seikosha Pound Patch' program in BEEBUG Vol.2 No.10 will enable that program to produce pound signs correctly on an OKI Microline 82A printer. In line 70 of the previous program change the value &B9 to &23.



INTRODUCING MACHINE CODE (Part 2)

Gordon Weston continues his series on machine code for beginners by looking at the programming of simple loops.

Before we look at any new ideas this month, we will set up a useful skeleton of a program that we can use in all our future examples. All assembly language programs need to be embedded between a few lines of Basic, and we'll use a standard version as shown in program 5. This is very similar to program 4 from the last article, except that the assembly language lines have been removed. From now on, lines 10 to 50 and lines 500 to 530 will be the same for each program and I will only give you the assembly language instructions to insert between these two sections.

Program 5

```
 10 MODE7
 20 DIM code 100
 30 FOR I%=0 TO 3 STEP 3:P%=code
 40 [
 50 OPT I%

500 ]
510 NEXT
520 CLS:CALL start
530 END
```

Now we will restore the lines that were removed, but with different line numbers:

```
100 .start
110 LDA #65
160 JSR &FFEE
210 RTS
```

As explained in the last article, this program will display a single "A" on the screen. The problem comes when we want to print a full line of A's on the screen. Obviously, we need some form of loop arrangement containing the statement JSR &FFEE, a loop counter and then a decision on whether to leave the loop.

There is an important check to make first, since we are using a jump to an operating system routine which may affect the contents of the microprocessor's registers. The User Guide page 457 under OSWRCH informs us that the contents of the A,X and Y registers are preserved by this routine. This means that we can leave the ASCII code for "A", 65, in the accumulator, and we are free to use the X and Y registers which can both be used as counters.

For reasons which will become clear later, the fastest method of loop counting is to count downwards which is called decrementing a counter. The assembly language instruction for this is 'DEX' for the X register, and 'DEY' for the Y register, which reduces the value in the corresponding register by one. We will use the X register and enter DEX in line 170. Because we want to print a full line on the screen, which is 40 characters, we want to repeat the printing loop forty times. We need to load the X register with 40 before the printing loop starts and this is done by entering LDX #40 in line 140, which in assembly language means LoaD the X register directly with 40. (LDY #40 would similarly load the Y register with 40). The new lines are:

```
140 LDX #40
170 DEX
```

After the value in the X register has been reduced by one, forty times, by going through the printing loop forty times, the value in the X

register is now ZERO, and we need a method of detecting this so that we can exit from the loop.

In the last article, I described three of the microprocessor's 8-bit registers (called A, X and Y). There is another 8-bit register called the status register, but instead of its contents being treated as one 8-bit number, each of the eight bits is treated individually as one of 8 flags showing what has happened in the microprocessor. One of these is called the 'zero' flag and indicates whether the result of the last instruction was zero, which is the condition we want to detect, to exit from the loop.

There are a set of assembly language instructions, called BRANCH instructions, which according to the flags in the status register, either allow the program to continue, or divert it to the address (or label) specified in the branch instruction.

The branch instruction we will use is 'BNE' standing for "Branch if Not Equal to zero" followed by a label to tell it where to branch to. We then insert the label in the program to show where to rejoin the program. The new lines are:

```
150 .loop1
180 BNE loop1
```

A label can be any name you choose (here we use 'loop1') and is always preceeded in assembler by a full stop. Note that the full stop should not be used when the label is part of an instruction. Enter the new lines 140,150,170 and 180 to make program 6 and run this (remember to include the skeleton program 5 as well).

The accumulator is loaded with the ASCII code for "A", 65, and then the X register is loaded with 40. "A" is printed on the screen as a result of JSR &FFEE and the X register is decremented at line 170 making the contents of X equal to 39 at the end of the first loop. Because the result was NOT zero, the branch instruction makes the microprocessor jump to the location identified by the label 'loop1' which means in effect that the program continues at line 150. On loop number forty, the contents of register X become zero, the branch does not occur, and the instruction RTS is reached, returning control to Basic.

Part Program 6

```
100 .start
110 LDA #65
140 LDX #40
150 .loop1
160 JSR &FFEE
170 DEX
180 BNE loop1
210 RTS
```

Part Program 7

```
100 .start
110 LDA #65
120 LDY #24
130 .loop2
140 LDX #40
150 .loop1
160 JSR &FFEE
170 DEX
180 BNE loop1
190 DEY
200 BNE loop2
210 RTS
```

The opposite of the branch instruction 'BNE' is 'BEQ' which stands for "Branch if EQual to zero". If you replace BNE with BEQ in line 180 only one "A" will be printed on the screen, because on the first pass through the print loop, X becomes 39 which is NOT zero and so no branch takes place.

To print 24 lines of A's we could use the Y register as a line counter. Restore line 180 so that it correctly contains the BNE instruction as listed above and add lines 120, 130, 190 and 200 to program 6 to make program 7.

The line counter Y has been loaded with 24 and also counts down to zero.

We can now use program 7 to test the use of another operating system routine. If you look at 'OSRDCH' in the summary table on page 452 of the User Guide, it gives you the entry address &FFE0 and says that this routine reads a character from the keyboard and stores it in the accumulator. On page 456 you are told again that X and Y are preserved.

Alter line 110 from LDA #65 to JSR &FFE0 and run the program. When you now press a key, the screen is filled with the character just entered.

As a little exercise, alter line 140 to LDX #1 and predict what will happen before you run the program and press a key. Alter line 140 to LDX #0 and 120 to LDY #1 and work out what has happened.

DATABASE PACKAGES REVIEWED

David Janda has selected four of the more popular database packages on the market and now reports on his findings in this all important area.

When deciding to buy a database it's best to consider what you need it for. Accessing data is a common task that we all wish to do and can with this type of package, but the ease with which you can get at your precious data is not so uniform. On a friendly database you may find that you can simply load the data and go. On others you may have to load display frames (called masks) before the main data file can be accessed.

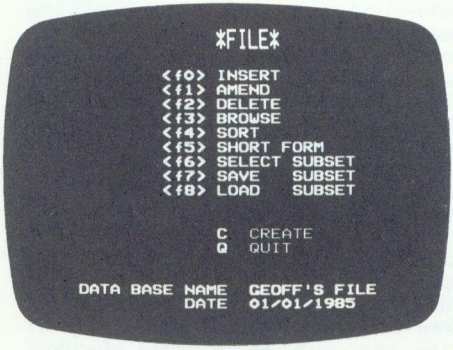
There are a two points that you should consider before buying a database. Is the package flexible enough for your needs and is it easy to use? Remember, it is your data, and you should be able to store and retrieve it according to your own criteria. Some packages are quite limited and only allow you to search and sort with quite severe restrictions. These are best avoided. Others are quite complex in their application but are near impossible to live with.

This review looks at four of the more popular databases for the BBC micro. As you will see, each offers different facilities, and different levels of friendliness.

```
Product : StarBASE
Format  : ROM
Supplier: GCC
          66, High Street, Sawson,
          Cambridge, CB2 4BG.
          0223 - 835330
Price   : £69
```

StarBASE from GCC of Cambridge was one of the first ROM database packages available for the Beeb. The package is supplied in two parts: the EPROM containing the main database code, and a utilities disc which contains several menu driven utilities that are used for file maintenance.

Unlike most other databases, it is necessary to dedicate at least one whole side of a disc for a single



StarBASE file. The reason for this is that the whole disc is specially formatted by Starbase, and a few special files are placed on the disc when formatting is complete. This gives you 395K of storage for a double sided disc file. This means that the physical position of each record is stored in a bit-map file and records can be quickly retrieved, but small files are unnecessarily wasteful of disc space.

StarBASE also requires the disc to be 'date stamped'. This involves the user entering the date, the user's name and an optional password when the disc is first formatted.

Common to many Beeb databases, StarBase allows a freeform input mask to be created. The mask is the size of one mode 7 screen, with text in any colour. A big plus for StarBASE is that the field name you select for use by the system can be up to eight characters in length.

The mask is designed by moving the cursor around the screen, placing text prompts as you go and inserting the field name, data type, and length in the required position. A panel at the bottom of the screen reminds you which function key does what during this operation.

Entering records is then simply a matter of typing in the required data as your own mask prompts for it. Again labelled function keys look after all the commands.

Amending, deleting and browsing through records is possible only on a subset of the original file. This may seem a bit of a bother if you wish to interrogate the whole file, but the subset defaults to include the whole file. Otherwise records are selected for the subset from any field according to several search criteria, - greater than, less than, and so on - applying to string as well as numeric fields.

Only one field can be searched at a time. Searching for records with specified items in two or more fields cannot be done in a single operation. Once found, a subset of the original file can then be searched further, sorted, saved as a separate file or printed out.

The utilities disc contains several routines that allow the user to design a print form, configure the printer, print address labels and so on. These routines cope well with printer codes, page breaks, and even the dreaded pound sign.

The whole package is menu driven. Each different stage has its own concise menu and selection of function keys, labelled on the screen. The package is reasonably fast (because of the special disc arrangement) and friendly to use.

Product : Beta-Base
Format : Disc
Supplier: Clares
98 Middlewich Rd., Northwich,
Cheshire, CW9 7DA.
0606 - 48511
Price : £25

Beta-Base from Clares proves that good software does not only exist on ROM. Beta-Base is completely menu driven, and like many databases, allows subsets to be created from a main file. Your file can be any length up to the size of your disc. The number of records allowed depends on the number of fields in each record - it is a swings and roundabouts system. However you can fit about 500 average ten field records onto a disc.

Creating a file mask with Beta-Base is not as easy to do as with some packages. There is no on screen editor to design a mask. Instead you just specify the name, length, and data type (string, integer, or real) of each field, but these can be easily edited at any time.

Formatting the output is a little tricky too. Complex control codes are used to print the fields that you want in the positions you want. However, once you have got used to this system it is very flexible.

Beta-Base really comes into its own when searching through a file. This package does not ask the user to select a key field - there isn't one! Instead, records within a file can be interrogated by up to any five selected fields each with a particular search criteria. The same applies when sorting a file, it can be done on any field desired. The search criteria include the usual 'more than', 'less than', etc. and also the all-too-rare 'not equal to' and the search for the occurrence of a sub-string within a field. In other words, you can be very selective.

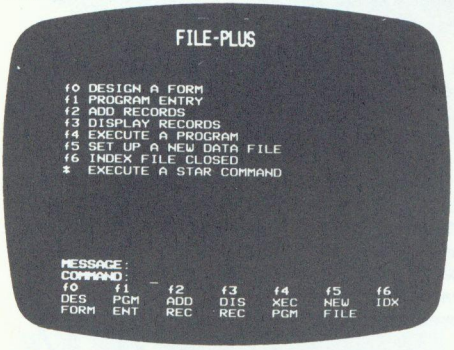
It is also very easy to create or sort a subset which is the same size as the original file and, if necessary, save it under a different name - not so easy with other databases. Indeed there is a facility to transfer complete or part files to another file and to merge files together.

A particularly friendly aspect of Beta-Base is the method employed to browse through a file - much as you might browse through a card index file. When in the browse mode, you can flick through the records, at two speeds, using the cursor keys. The feeling is that of using a word processor, flicking through text. Any record can be edited in this mode, or deleted altogether.

Beta-Base is a joy to use. At all stages of file creation and record

entry, the user is informed of what is happening and steps taken to prevent destructive mistakes. My only grumble is that there is no free format in designing an input mask, instead a record will only be displayed 'as is'. However this gripe is easily offset by the thoughtful approach to Beta-Base's design and its power in use.

Product : File-Plus
Format : ROM
Supplier: Watford Electronics
Cardiff Road, Watford,
Herts.
0923 - 40588
Price : £49.50



My first impression of File-Plus was very good. The reason is because File-Plus has an in-built interactive query language (IQL). An IQL is used to write small programs that can be used to search your files on a variety of criteria. In a simple form an IQL might allow you to say 'IF AGE>21 THEN PRINT', which would print the record.

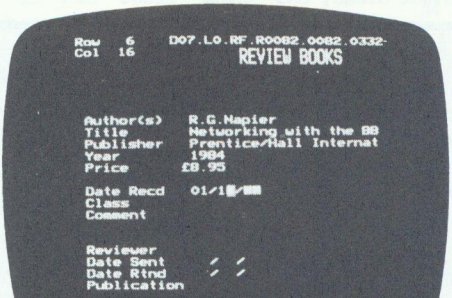
File-Plus allows individual files to span across up to three disc sides if required, giving more than enough storage space for most needs. A screen editor is provided to allow you to design your input mask. This is rather crude to use but it will allow a mask of up to three mode 7 screens to be created. Each field is assigned only a two letter identifier. As a result, it can get difficult to find suitable different combinations of letter pairs to adequately describe a field.

File-Plus is referenced by an index file. So after entering data it is best to sort the file. That sounds okay, but you have to use a separate program on a utilities disc to do the sorting. Once this is done, a separate work file is created which contains the key index file. You then go back to the main menu, load the mask file, load the 'assist' file and open the index file. Only then can you start to search the database!

You could decide not to bother with the sorting and decide to search the file sequentially. That's alright if the field you are interrogating is of the string type, but sequential searching of a numeric field is not allowed.

There are some good points to File-Plus. The IQL is very useful and, like the other features, is well described in the manual with plenty of examples. Output formatting for printouts is reasonably flexible using embedded commands. All in all, File-Plus has the elements of being a good database, but its present implementation is very tatty. Extracting your data is tedious and time consuming.

Product : Datagem
Format : ROM
Supplier: Gemini
18a Littleham Road, Exmouth,
Devon, EX8 2QG.
0395 - 265165
Price : £129.95



When DataGem was first announced it caused quite a stir of excitement. Not only is it ROM based with 24K of code, but it is pretty costly as well.

For your money you get a couple of EPROMs on a large carrier board, a hefty manual and a demo disc. The manual and the demo disc contain several examples on using DataGem and you would be wise to go through them thoroughly to get to grips with this difficult package.

Files must be created to their maximum size at the start. Datagem can handle files with up to 5100 records. An input mask for your file is set up with a flexible but unfriendly screen editor that allows you to design input forms of up to four mode 7 screens worth in different colours. Data types include the expected numeric and string, along with a 'date' type.

Output is also very flexible. A separate form can be designed for printed output, and it is possible to use the output with Wordwise and View to produce complete documents. In all cases there are sensible defaults.

DataGem is let down by its search facilities. The criteria can be very complex - multiple field searches are provided for and complex equations within searches can be made. However, you cannot assign names to fields, not even simple two letter names like File-Plus uses, so to search the first field you must specify it as 'F1' not as 'Item sold', or whatever. The records found can be held in up to eight areas, and these areas can be sorted, merged and so on to extend your search. Although very complex searches are possible, it is not a simple process and you'll need time to learn.

The worst feature of DataGem is the user interface. DataGem requires a lot of Ctrl-key stokes to perform operations and uses several long menus. No use is made of the function keys and the net result is confusing and occasionally even destructive of data in the hands of the inexperienced.

In a package of this size and price this is really inexcusable. It is not unreasonable to expect a more friendly approach to both the mask designing and the searching. Why no Interactive Query Language here?

Make no mistake about it, DataGem is a very powerful package that does the job and well,but I would not recommend it to the first time user or anyone who requires rapid access to their data.

With such widely varying packages it is impossible to give a 'best buy'. This must depend on your own particular needs. Do, however, try to select a package that is flexible enough to cater for your future as well as your present requirements.

In addition, you may be surprised that we have not included Masterfile in this review even though it is one of the more popular databases. We have taken this course of action, as explained in the editorial, in order to maintain an objective stance. Instead we have listed its features for readers to make their own comparisons.

MASTERFILE II – BEEBUGSOFT'S DATABASE PACKAGE

MASTERFILE II is the new random access disc database package from BEEBUGSOFT. It is extremely easy to use, being menu driven, and is supplied with a comprehensive 50 page manual. MASTERFILE II replaces the original MASTERFILE program and offers many new features, including the ability to combine with Wordwise or View to produce standard letters. Some of the main features are listed below:

Field lengths of up to 255 characters.

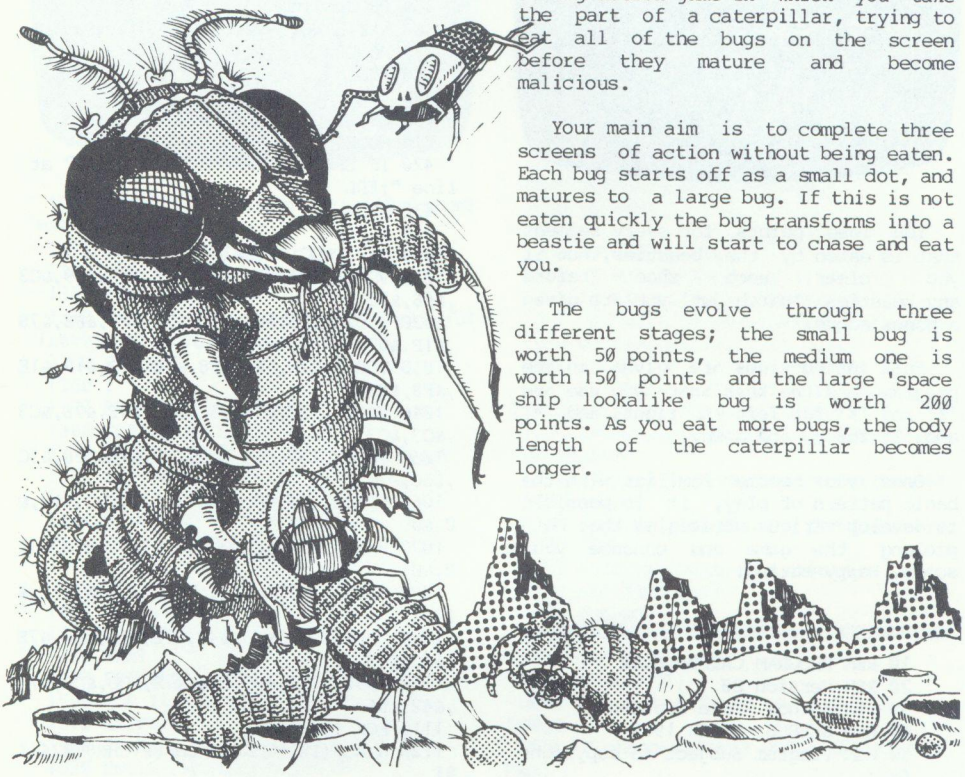
Up to 18 fields per record.

Field types of string, integer, numeric, decimal and date.

File size limited only by disc capacity.

CATERPILLAR

Caterpillar is a simple yet delightfully addictive action game, featuring some really fast and smooth graphics. Dave Robinson tells you what it's all about.



Caterpillar is a one player, fast moving action game in which you take the part of a caterpillar, trying to eat all of the bugs on the screen before they mature and become malicious.

Your main aim is to complete three screens of action without being eaten. Each bug starts off as a small dot, and matures to a large bug. If this is not eaten quickly the bug transforms into a beastie and will start to chase and eat you.

The bugs evolve through three different stages; the small bug is worth 50 points, the medium one is worth 150 points and the large 'space ship lookalike' bug is worth 200 points. As you eat more bugs, the body length of the caterpillar becomes longer.

Fast sorting and searching using a system of 'Tag' files which may be performed on most combinations of criteria (by using logical operators).

Instant update of individual records as required.

Global update of integer, numeric or decimal fields.

Some spreadsheet facilities.

Records may be printed in a format of your own design.

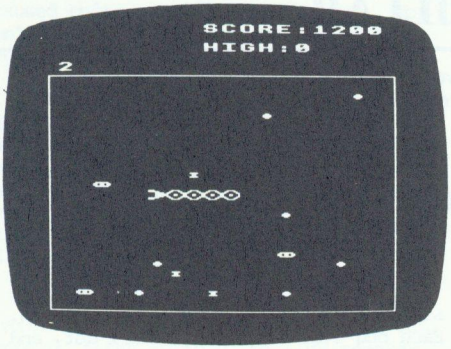
Printing may include your own text in any format.

Output from MASTERFILE II may be used with Wordwise or View.

Flexible multi-label printing.

6502 second processor compatible.

MASTERFILE II is priced at £19.00 (before members' discount). For further details please contact The Software Manager, BEEBUGSOFT, PO Box 50, St. Albans, Herts.

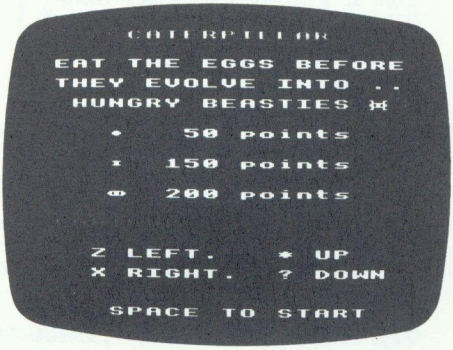


You lose points for every segment that is eaten by the beasties, but if you clear each sheet before any beasties 'hatch out' you are given a bonus score.

Full instructions are given in the program, and the keys that you use are 'Z' and 'X' for left and right, and '*' and '?' for up and down.

Once you become familiar with the basic pattern of play, it is possible to develop various strategies that both prolong the game and enhance your score. Happy eating!

```
  10 REM PROGRAM CATERPILLAR
  20 REM Version B0.1
  30 REM Author  Dave Robinson
  40 REM BEEBUG  MARCH 1985
  50 REM Program Subject to copyright
  60 :
 100 ON ERROR GOTO 400
 110 DIM X%(21),Y%(21)
 120 DIM head$(4),beast$(3)
 130 DIM egg$(3),egg%(20)
 140 DIM change%(20)
 150 DIM BX%(20),BY%(20)
 160 PROCinitial
 170 MODE5
 180 PROCinstruct
 190 REPEAT
 200 screen%=1
 210 REPEAT
 220 PROCsetup
 230 PROCdrawscreen
 240 REPEAT
 250 PROCworm
 260 IF move%>4 PROCchange:move%=0
 270 IF aim%=target% won%=TRUE
 280 PROCbeast
 290 beast%=beast%MOD(BN%+1)+1
 300 UNTIL won% OR lost%
 310 IF BN%=0 PROCbonus
 320 screen%=screen%+1
 330 UNTIL lost% OR screen%=4
 340 PROCpause(100)
 350 PROCending
 360 UNTIL NOT again%
 370 MODE6
 380 END
 390 :
 400 ON ERROR OFF
 410 MODE7
 420 IF ERR<>17 THEN REPORT:PRINT" at
line ";ERL
 430 END
 440 :
1000 DEF PROCinitial
1010 VDU23,224,&3C,&66,&C3,&99,&99,&C3
,&66,&3C
1020 VDU23,225,&07,&1F,&78,&F8,&F8,&78
,&1F,&07
1030 VDU23,226,&E0,&F8,&1E,&1F,&1F,&1E
,&F8,&E0
1040 VDU23,227,&18,&3C,&3C,&7E,&7E,&C3
,&C3,&C3
1050 VDU23,228,&C3,&C3,&C3,&7E,&7E,&3C
,&3C,&3C
1060 VDU23,229,&0,&0,&18,&3C,&3C,&18,&
0,&0
1070 VDU23,230,&0,&0,&3C,&18,&18,&3C,&
0,&0
1080 VDU23,231,&0,&0,&7E,&DB,&DB,&7E,&
0,&0
1090 VDU23,232,&C3,&42,&7E,&DB,&DB,&7E
,&42,&C3
1100 VDU23,233,&66,&42,&7E,&DB,&DB,&7E
,&42,&66
1110 FOR I%=1 TO 4
1120 head$(I%)=CHR$17+CHR$2+CHR$(224+I
%)
1130 NEXT
1140 body$=CHR$17+CHR$2+CHR$224
1150 beast$(1)=CHR$17+CHR$1+CHR$232
1160 beast$(2)=CHR$17+CHR$1+CHR$233
1170 beast$(3)=CHR$17+CHR$2+CHR$233
1180 FORI%=1 TO 3
1190 egg$(I%)=CHR$17+CHR$3+CHR$(228+I%)
1200 NEXT
1210 hiscore%=0:score%=0
1220 ENDPROC
1230 :
1240 DEF PROCsetup
1250 BN%=0
1260 lost%=FALSE:won%=FALSE
1270 FOR I%=0 TO 5
1280 X%(I%)=I%+1:Y%(I%)=10
1290 NEXT
```




```
 1300 OX%=5:OY%=10:head%=4:tail%=0:aim%
=-1:B%=1
 1310 FOR I%=0 TO 20
 1320 BX%(I%)=0:BY%(I%)=0
 1330 NEXT
 1340 target%=4*screen%+4:WXdir%=+1:WYd
ir%=0:H%=1:move%=0:beast%=0
 1350 ENDPROC
 1360 :
 1370 DEF PROCdrawscreen
 1380 CLS:VDU23,1,0;0;0;0;
 1390 VDU19,2,6,0,0,0:VDU19,1,3,0,0,0
 1400 PRINTTAB(9,1)"SCORE:"
 1410 PRINTTAB(9,3)"HIGH:"
 1420 PRINTTAB(1,5);screen%
 1430 GCOL0,1:MOVE32,48
 1440 DRAW1248,48:DRAW1248,816
 1450 DRAW32,816:DRAW32,48
 1460 FOR I%=1 TO 4
 1470 PRINTTAB(I%,10)body$
 1480 NEXT
 1490 PRINTTAB(5,10)head$(1)
 1500 FOR I%=0 TO target%
 1510 REPEAT
 1520 J%=RND(16)+1:K%=RND(22)+7
 1530 UNTIL FNpoint(J%,K%)=0
 1540 PRINTTAB(J%,K%)egg$(1)
 1550 egg%(I%)=J%*&100+K%
 1560 change%(I%)=1
 1570 NEXT
 1580 FOR I%=target%+1 TO 20
 1590 egg%(I%)=0:change%(I%)=4
 1600 NEXT
 1610 COLOUR2:PRINTTAB(15,1);score%
 1620 PRINTTAB(14,3);hiscore%
 1630 ENDPROC
 1640 :
 1650 DEF PROCworm
 1660 move%=move%+1:X%=OX%:Y%=OY%
 1670 IF INKEY(-67) WXdir%=+1:WYdir%=0:
H%=1
 1680 IF INKEY(-98) WXdir%=-1:WYdir%=0:
H%=2
 1690 IF INKEY(-105)WYdir%=+1:WXdir%=0:
H%=3
 1700 IF INKEY(-73) WYdir%=-1:WXdir%=0:
H%=4
 1710 X%=X%+WXdir%:Y%=Y%+WYdir%
 1720 C%=FNpoint(X%,Y%)
 1730 IF C%=3 PROCswallow:ENDPROC
 1740 IF C%<>0 ENDPROC
 1750 PRINTTAB(X%,Y%)head$(H%)
 1760 PRINTTAB(X%(head%),Y%(head%))body$
 1770 PRINTTAB(X%(tail%),Y%(tail%))CHR$
32
 1780 OX%=X%:OY%=Y%
 1790 head%=(head%+1)MOD21
 1800 tail%=(tail%+1)MOD21
 1810 X%(head%)=X%:Y%(head%)=Y%
 1820 SOUND1,-15,(X%+10)*Y%,1
 1830 ENDPROC
 1840 :
 1850 DEF PROCbeast
 1860 IF BN%=0 PROCpause(3)
 1870 IF BX%(beast%)=0 ENDPROC
 1880 OBX%=BX%(beast%):OBY%=BY%(beast%)
 1890 Xdir%=SGN(X%(tail%)-OBX%)
 1900 Ydir%=SGN(Y%(tail%)-OBY%)
 1910 IF OBX%+Xdir%=X%(tail%) AND OBY%+
Ydir%=Y%(tail%)PROCeat:ENDPROC
 1920 IF FNpoint(OBX%+Xdir%,OBY%+Ydir%)
<>0 ENDPROC
 1930 BX%(beast%)=OBX%+Xdir%:BY%(beast%
)=OBY%+Ydir%
 1940 PRINTTAB(BX%(beast%),BY%(beast%))
beast$(B%)
 1950 PRINTTAB(OBX%,OBY%)CHR$32
 1960 B%=(B%MOD2)+1
 1970 ENDPROC
 1980 :
 1990 DEF PROCchange
 2000 Z%=RND(21)-1
 2010 IF change%(Z%)=4 OR egg%(Z%)=0 EN
DPROC
 2020 IF change%(Z%)=3 PROCnewbeast:END
PROC
 2030 PRINTTAB(egg%(Z%)DIV&100,egg%(Z%)
MOD&100)egg$(change%(Z%)+1)
 2040 IF change%(Z%)<3 change%(Z%)=chan
ge%(Z%)+1
 2050 ENDPROC
 2060 :
 2070 DEF PROCnewbeast
 2080 PRINTTAB(egg%(Z%)DIV&100,egg%(Z%)
MOD&100)beast$(1)
 2090 change%(Z%)=4
 2100 target%=target%-1:BN%=BN%+1
 2110 BX%(BN%)=egg%(Z%)DIV&100:BY%(BN%)
=egg%(Z%)MOD&100
 2120 ENDPROC
 2130 :
 2140 DEF PROCbonus
 2150 COLOUR1:bonus%=screen%*500
```

```
 2160 PRINTTAB(1,2)"*BONUS*"
 2170 PRINTTAB(3,4);bonus%
 2180 PROCpause(50)
 2190 score%=score%+bonus%
 2200 COLOUR2:PRINTTAB(15,1);score%
 2210 PROCpause(100)
 2220 ENDPROC
 2230 :
 2240 DEF PROCending
 2250 *FX15,1
 2260 again%=TRUE:CLS:COLOUR1
 2270 IF lost% PRINTTAB(6,6)"OH DEAR!",
TAB(6,8)"YOU LOST"
 2280 IF won% PRINTTAB(5,8)"WELL DONE!"
 2290 PRINTTAB(3,11)STRING$(14,"-")
 2300 COLOUR2
 2310 PRINTTAB(5,15)"SCORE:";score%
 2320 IF score%>hiscore% THEN hiscore%=
score%
 2330 PRINTTAB(5,18)"HIGH:";hiscore%
 2340 COLOUR3
 2350 PRINTTAB(4,24)"AGAIN (Y/N)?"
 2360 REPEAT:UNTIL INKEY-69 OR INKEY-86
 2370 IF GET$="N"again%=FALSE
 2380 score%=0
 2390 ENDPROC
 2400 :
 2410 DEF PROCswallow
 2420 IF head%<tail% top%=head%+21 ELSE
 top%=head%
 2430 PRINTTAB(X%,Y%)head$(H%)
 2440 S%=-1
 2450 REPEAT
 2460 S%=S%+1
 2470 UNTIL egg%(S%)DIV&100=X% AND egg%
(S%)MOD&100=Y%
 2480 egg%(S%)=0
 2490 PRINTTAB(X%(head%),Y%(head%))body$
 2500 FOR I%=top% TO tail% STEP-1
 2510 PRINTTAB(X%(I%MOD21),Y%(I%MOD21))
egg$(1)
 2520 SOUND1,-15,1*I%*3,1
 2530 PROCpause(2)
 2540 PRINTTAB(X%(I%MOD21),Y%(I%MOD21))
body$
 2550 NEXT
 2560 OX%=X%:OY%=Y%
 2570 head%=(head%+1)MOD21
 2580 X%(head%)=OX%:Y%(head%)=OY%
 2590 aim%=aim%+1
 2600 score%=score%+change%(S%)*50
 2610 COLOUR2:PRINTTAB(15,1);score%
 2620 PRINTTAB(15,1);score%
 2630 IF aim%=target% won%=TRUE
 2640 ENDPROC
 2650 :
 2660 DEF PROCeat
 2670 PRINTTAB(OBX%+Xdir%,OBY%+Ydir%)be
ast$(3)
 2680 tail%=(tail%+1)MOD21
 2690 aim%=aim%-1
 2700 IF aim%=-4 THEN lost%=TRUE
 2710 PRINTTAB(OBX%,OBY%)CHR$32
 2720 SOUND1,-15,10,2
 2730 PROCpause(10)
 2740 SOUND1,-15,60,1
 2750 PRINTTAB(OBX%,OBY%)egg$(1)
 2760 S%=-1
 2770 REPEAT:S%=S%+1:UNTIL egg%(S%)=0
 2780 egg%(S%)=OBX%*&100+OBY%
 2790 change%(S%)=1
 2800 PRINTTAB(OBX%+Xdir%,OBY%+Ydir%)be
ast$(B%)
 2810 BX%(beast%)=BX%(beast%)+Xdir%:BY%
(beast%)=BY%(beast%)+Ydir%
 2820 score%=score%-50
 2830 COLOUR2:PRINTTAB(15,1);score%
 2840 ENDPROC
 2850 :
 2860 DEFFNpoint(S%,T%)
 2870 =POINT(S%*64+32,1008-T%*32)
 2880 :
 2890 DEF PROCpause(time%)
 2900 TIME=0:REPEAT:UNTIL TIME=time%
 2910 ENDPROC
 2920 :
 2930 DEF PROCinstruct
 2940 VDU19,1,6,0,0,0
 2950 VDU23,1,0;0;0;0;
 2960 COLOUR1
 2970 PRINTTAB(4,2)"CATERPILLAR"
 2980 PRINTTAB(4,3)STRING$(11,"-")
 2990 COLOUR3
 3000 PRINT'"EAT THE EGGS BEFORE"
 3010 PRINT'"THEY EVOLVE INTO .."
 3020 COLOUR2
 3030 PRINT'TAB(1)"HUNGRY BEASTIES! ";C
HR$232
 3040 COLOUR3
 3050 PRINTTAB(3,12);CHR$229;SPC3;"50 p
oints"
 3060 PRINTTAB(3,15);CHR$230;SPC2"150 p
oints"
 3070 PRINTTAB(3,18);CHR$231;SPC2"200 p
oints"
 3080 COLOUR1:PRINT''"CONTROLS:"
 3090 COLOUR3
 3100 PRINT''TAB(2)"Z LEFT.   * UP"
 3110 PRINT'TAB(2)"X RIGHT.  ? DOWN"
 3120 COLOUR2
 3130 PRINTTAB(3,30)"SPACE TO START"
 3140 REPEAT
 3150 PRINTTAB(17,9)CHR$232,TAB(16,9)CH
R$32
 3160 IF INKEY(-99) ENDPROC
 3170 PROCpause(30)
 3180 PRINTTAB(16,9)CHR$233,TAB(17,9)CH
R$32
 3190 IF INKEY(-99) ENDPROC
 3200 PROCpause(30)
 3210 UNTIL FALSE
 3220 ENDPROC
```

BEEBUG Workshop

USING ERROR TRAPPING

This month Surac starts a new workshop on the subject of error trapping, and explains various methods for recovering from errors.

One advantage of Basic, above machine code for example, is its ability to inform us of errors when they occur, allowing us to react to them in an intelligent manner without the loss of either our program or data (apart from very exceptional circumstances).

THE BASIC KEYWORDS

Error trapping ('catching' an error once it's happened and diverting the execution of the Basic program to a specific point as a result) is accomplished in Basic with the use of some code of the form of either:

ON ERROR GOTO <line number>

or more generally:

ON ERROR <Basic statement(s)>

The key to making a program react to errors in a sensible or useful manner is the use of this 'ON ERROR' construct and some carefully planned code. Remember too, that although this is referred to as error trapping, the occurrence of such an 'error' does not necessarily imply a bug in the program but a situation which Basic is just unable to handle by itself.

Whenever an error occurs, there is an associated error number and error message that can be accessed although the messages may be of zero length. Basic provides access to the error number with the ERR function, and the error text may be printed using the REPORT statement. Basic also provides a further aid to error handling; ERL is the 'error line', which tells us the line which Basic was executing when the error occurred. Once executed an 'ON ERROR' piece of code remains active until then end of the program, or another ON ERROR is encountered. We can also return to the default error handler (it's the one that normally prints up error messages before we introduce an error trap of our own), using ON ERROR OFF.

As a very simple example of error trapping, consider the short program below, which inputs a mathematical function (line 30) and uses this to evaluate a function in terms of the variable X (line 50). If an error occurs, then it is trapped and the error routine (lines 90 to 120) is called.

```
 10 REM EXAMPLE 1
 20 ON ERROR GOTO 90
 30 INPUT"Formula :"A$
 40 X=-1
 50 PRINT EVAL A$
 60 X=X+1
 70 IF X=2 THEN 30 ELSE 50
 80 :
 90 IF ERR=17 REPORT:END
100 REPORT
110 PRINT" when X=";X;" for ";A$'
120 GOTO 60
```

Error number 17 (ERR=17 in line 90) is produced by the user pressing Escape while running the program. Many programs use this as a legitimate way to terminate (or escape from) a program as this example shows, although Basic traps this as an 'error'.

When an error occurs in a program, Basic tends to 'forget' what it was just doing, partly in case it has just been wrapping itself in some horrendous loop. Although this saves us from some problems (if a procedure was calling itself repeatedly for example), it does create a few others because Basic will forget about any FOR-NEXT and REPEAT-UNTIL loops, and any nested procedures or functions that had been entered (or called) at the time of the error.

PROGRAMS THAT RECOVER FROM ERRORS

Overall, to make a program recover from an error is quite simple: it needs to 'know' what it was doing immediately prior to the error, what has been lost as a result of the error (variables, loops, etc.), and what it should do to recover from the error. By taking ERR, ERL and a few user-defined 'flags', a lot of information can be gleaned by the program about its position just prior to the error, and appropriate action then taken.

Errors also clear the Basic stack of any references to local variables active at the time of the error. If an error occurs whilst a local variable is in use, its correct global value is lost, and it reverts to its current local value. This will all too often prove catastrophic.

There is no easy solution to this problem, apart from doing your best to ensure that an error such as this never occurs by some careful forward thinking. This can generally be catered for by making extra tests to see if any actions are likely to cause an error (e.g. make the program test for division by zero rather than using error trapping). One cause of 'errors', even in a debugged program, is from pressing the Escape key. The addition of *FX229,1 towards the start of your program will help to eliminate this problem by causing the Escape key to 'lose' its error generating ability.

THE SERIOUSNESS OF ERRORS

It is normally worthwile to distinguish between errors from which a recovery is possible (termed 'non-fatal'), and those from which it isn't (termed 'fatal'). For example, a program may rely upon a data file being present on disc to work at all; if a 'No such file' error were to occur when attempting to initiate the program, then the effect of the error would be fatal because the program would not be able to continue. If, however, an 'Accuracy lost' error were to be generated when performing a lengthy calculation, the program could substitute a 'default' value, and carry on without too much difficulty.

ERRORS DEEP DOWN

Without going into great detail (see the User Guide page 464 for this) we will look briefly at how 'errors' are stored in memory. Working in terms of bytes from a given address, the byte at the address will hold a zero; the next byte the error number (ERR); the next few (up to about 250) bytes the error message (as printed by REPORT); and the last byte a zero. It is quite permissable to have an error message of no length (in which case a zero would immediately follow the ERR byte). Note also that Basic applies a small amount of compression to error messages; any Basic keywords are tokenised. The delightful thing about the arrangement of errors in memory is that, although it needs to be executed as machine code, it is very easy to take a given message and error number, and to generate that error artificially from within a Basic program. For example, try the following:

```
10 ON ERROR REPORT:PRINT ERR:END
20 INPUT "Error number "E%
30 INPUT "Error text "E$
40 DIM A% LEN E$+2
50 ?A%=0 : REM ERROR INSTRUCTION
60 A%?1=E% : REM 'ERR'
70 $(A%+2)=E$ : REM 'REPORT'
80 ?(A%+LENE$+2)=0 : REM STOP
90 CALL A% : REM MAKE ERROR
```

Although rather crude as it stands, the program above, whch can be easily modified, illustrates how easy it is to generate your own errors.

ZERO ERRORS

With some errors, 'No room' for example, trying to call an error routine may simply make matters worse (if more memory is required), the result potentially being a continuous loop that achieves nothing. To prevent this happening, Basic treats any error with an error code of zero differently, and uses its own error handler regardless of the error trapping set up by the user.

You will have gathered by now that although error trapping has its undoubted uses it also has its pitfalls. It is nearly always worthwhile incorporating some error trapping in your own programs and there are many examples in BEEBUG programs that you can follow.

TURBO PASCAL

Following his recent review of Acornsoft's P-system Pascal, John Maher now turns his attention to another Pascal language system for the Beeb. Here he reports on Turbo Pascal. Is this the Pascal for you?

Product : Turbo Pascal
Supplier : Altor Computer Software,
801 Govan Road, Glasgow.
Tel 041-445-5130
Price : £63.19

Turbo Pascal was developed by Borland, a California based software company for internal use on the IBM PC, and since its introduction just over a year ago it has received 'rave' reviews in the States. It will run under CP/M, CP/M-86, and MS-DOS with its own simple operating system: an editor, a compiler, and some disc and program utilities. The whole system can now be run on a BBC micro using a Z-80 Acorn second processor. The Turbo Pascal system occupies about 30k leaving about 23K for programs and text. Turbo Pascal can also be used with the Torch Z-80 system and gives 28K of space since CP/N (MCP) is in ROM. The compiler can compile to memory or to disc to produce .COM files, source programs can have include files, and the system allows code overlays, so that very large programs can be constructed.

The Turbo compiler is very fast, compilation to memory is blindingly fast. A 1250 line program compiled to disc in 1min 21secs. Compiled programs also run fast since compilation is to Z80 code, not to an intermediate code as in ISO-Pascal or UCSD Pascal. Benchmark comparisons were given in the previous review of the Acornsoft/TDI P-system (BEEBUG Vol.3 No.8). During compilation, if a syntax error is discovered, an error message is given, the editor is called and the relevant text area is displayed, with the cursor pointing to the error. Run time errors can also be found easily. The very high compilation speed, together with the ability to run programs from memory, and the ease in finding errors, makes Turbo Pascal almost like Basic (perish the thought!) in its immediate responses during program development.

The Turbo system does not provide for libraries like the P-system. However, the order of declarations, which is normally rigidly enforced in Pascal, has been relaxed, so that a separate portion of source text, with its own declarations, can be incorporated as necessary at the head of a program.

Turbo Pascal follows the standard set by Jensen & Wirth in their 'Pascal User Manual and Report'. However, there are significant differences such that at least one other reviewer has said that 'this is not Pascal'. The differences derive largely from the language extensions, and are those which might be wished for by any working programmer. Amongst these differences GET and PUT have not been implemented, and instead READ and WRITE have been extended considerably. I found this easier to use.

The extensions to Pascal include absolute address variables, bit/byte manipulation, direct access to the CPU memory and to the data ports, dynamic strings and various logical operations on integers. In-line machine code can be included in Pascal but there is no assembler like that for BBC Basic, you have to pre-assemble and enter the code as hexadecimal numbers (between square brackets). In addition, there is program chaining with common variables, random access data files, structured constants and type conversions. The structured constants are an interesting development, since they allow the initialisation of variables at the head of a program - like INIT in PL/1. Added to this list are over one hundred standard identifiers, with string handling facilities, blockreads, blockwrites, and fillchar like UCSD Pascal, and facilities for converting STRINGs to and from INTEGER and REAL numbers. For scientific calculations Turbo Pascal has considerable

attractions over most other Pascals without double precision, as the REALs use six bytes, to give an accuracy of eleven significant digits.

Turbo Pascal does not yet have an accompanying assembler, though EXTERNAL subprograms can be called. These can be set up with any CP/M assembler, and an absolute area of memory can be set aside from the operating system.

The Turbo Reference Manual, whilst containing some text errors, is excellent and very comprehensive; it is well indexed and clear. Some program examples are provided, both on disc and in the manual.

The Turbo editor is a screen editor specifically designed for program text. It is also a good word processor. If you are familiar with MicroPro's Wordstar, then since the standard commands are exactly the same, you will have no problems. It is possible to reconfigure the system to your own taste, via an installation program TINST.COM. This allows you to set screen formats and editing commands. However, this does not use the BBC cursor keys.

SHOULD I BUY TURBO PASCAL?

If you have a Torch or Acorn Z80 processor this is by far the best version of Pascal to use, either for the novice or the experienced programmer. I prefer it to either ISO Pascal or Acornsoft/TDI UCSD Pascal on the BBC micro. It is also very inexpensive in itself.

[Despite the comments above, ISO-Pascal from Acornsoft (reviewed BEEBUG Vol.3 No.6) will prove quite satisfactory for many, and without the necessity of paying for a costly second processor as well. Ed.]

POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG POSTBAG

BEEBUG FOR BEGINNERS

I find BEEBUG magazine great value for money, but I can't help but believe that the vast majority of your membership, like me, often ends up struggling as a very interested computer user, rather than an out and out computer boffin. I appreciate how tedious it must be for seasoned experts to have every detail explained but I feel this is essential if you are not to lose members through frustration.

Ian Jester

We would expect that BEEBUG members cover the whole range from complete beginner to seasoned expert. For this very reason we always try to present as much variety as possible in the magazine to cater for all tastes. Every issue contains some articles aimed specifically at beginners and others with detailed explanations. We also try to present all programs in such a way that they may be used by anyone, even those with no programming knowledge at all. However, it would be impractical to provide detailed yet simple explanations of all programs, especially the more complex ones. If you too have a view on this subject then we would like to hear from you.

ASTAAD AND PRINTMASTER

Those who find graphics printing exasperatingly slow, particularly when doing fairly small drawings such as logos, with ASTADD (see BEEBUG Vol.2 Nos. 7 & 9) and have Printmaster fitted may like to make the following modification to ASTAAd2 so that the WINDOW facility comes into play and you don't have to wait while areas of blank screen are being copied!

Change line 2190 to read:

```
2190 INPUTTAB(6,0)"Use Arrows, Tab & Re
turn to set WINDOW, else Return"
```

Add line 2194 *WINDOW

Add line 2196 *Gdump 0 3 1 1 5

Delete '12' from line 2220

Delete lines 2210 & 2220

Insert '12' in line 2250 to give

```
2250 VDU12,5:ENDPROC
```

F.Duerden

Thanks for the idea.

ELITISM WITH WATFORD DFS

I, like many others, like to match the excellence of the Beeb with the best that is available in support systems and have bought a DFS that gives better facilities than the Beeb product - in my case Watford's DFS. Unfortunately, the disc version of Elite does not run under the Watford

DFS - nor, I believe, under Pace AMCOM.

On phoning Acornsoft, I received only the advice that the game had been written with the Acorn DFS in mind, not others.

In my view, Acorn are going to lose out. Given the choice of a good game or a good DFS, the latter takes preference. If game sales are to be maximised, all Acornsoft have to do is to be a little less clever in their programming routines - we could all enjoy the game as well as our choice of DFS.

B.G.Hulatt

First of all, as a matter of fact, Elite will work with later versions of the Watford DFS, certainly version 1.4 or later (and some versions 1.3). Whether Acornsoft should take positive steps to ensure compatibility of their products with other than the Acorn DFS is more debatable. As the original manufacturer Acorn's DFS sets the standard that others should surely follow. The fact that both Watford and Pace DFS have progressed through several versions would indicate that these suppliers recognize that situation. That being so, there is then a strong obligation on Acorn to ensure that all relevant information is readily available to third party suppliers. After all, strong software and hardware support will ensure the continuing success of the Beeb for Acorn for some time to come.

FROM AMCOM DFS TO SECOND PROCESSOR

I have recently purchased a 6502 second processor which did not at first function correctly. I found that the AMCOM DFS (used before) works with link S9 closed while the second processor will only work with link S9 open, but there was no mention of this in any of the documentation.

N.J.Garrod

Thanks for the warning.

BEEB FAULTS AGAIN

Halfway through reading the letter from Mr Walker in the October issue of BEEBUG (Vol.3 No.5), I was compelled to check whether I had written it myself! The fault described bore an uncanny resemblance to a problem I had had with my own machine.

Again the fault first appeared after fitting an ATPL board, but also persisted when this was removed. The power leads to the Beeb's main board were suspected and the connectors resoldered, but the problem recurred when the ATPL board was refitted.

Finally, with some expert help, the problem was eventually traced to IC14 which was replaced, and all has been well for some six months now.

So if you have similar problems, do check the integrity of the power leads, and look at this chip if you have similar problems.

Gareth Hughes

This confirms the experiences described by Mr Walker in his original letter referred to above.

BASIC II AND 6502C CHIPS

Is there any chance of BEEBUG supplying the Basic II chip like you did with the O.S. 1.2 ROM. I'm sure quite a few members would purchase them if available. Also, is there any chance of a report on the new enhanced 6502 chip advertised at around £17.50? Is it compatible with the BBC micro and will the new chip work with the standard software?

R.H.Nightingale

We are not able at present to offer Basic II chips as these have not generally been made available by Acorn. The CMOS 6502 chip used in the Acorn second processor could be installed in the Beeb in place of its standard 6502. However, there would appear to be little advantage in doing this for most users, though there should be no problems of compatibility. The extra speed (3Mhz instead of 2Mhz) will have no effect beacause of the basic clock rate used while the additional machine code instructions will not be supported by the Beeb's standard assembler and would need to be pre-assembled and poked directly to memory. For more advanced users we hope to publish shortly an article on using the 6502C.

SPIDER MAN

Our game of Spider Man may at first sight seem to be a typical arcade game, but closer examination will reveal that this most original game by R. Lewis presents a bewildering challenge of tactics and ingenuity, the like of which is not often seen.

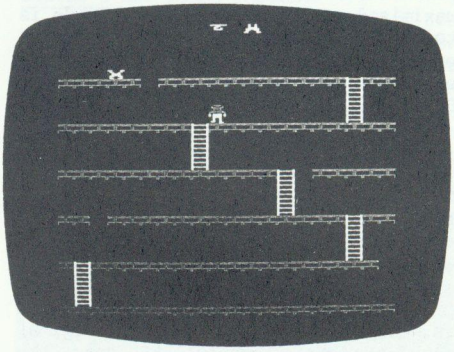
Your mission is to capture the mutant spiders that are patrolling the underground systems. To touch a spider means instant death, but you can jump and grab a floor tile to capture a spider immediately above you. Thus temporarily trapped, the spider can be transported to one of the hooks at the top of the screen to kill it.

The game sounds easy but it isn't, especially as there are gaps in the walkways, down which you can fall. This also means that if you want to 'hang' a spider, you must make sure that the walkways on your route to the hooks at the top are complete. Floor sections can be removed and inserted in the gaps at will to achieve this.

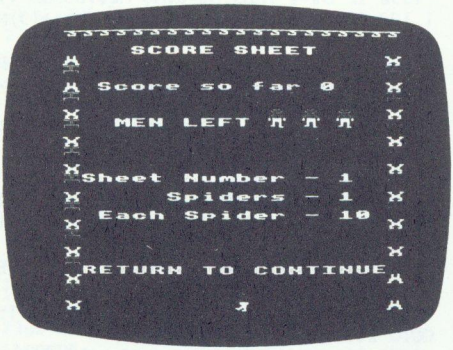
All the relevant information is displayed on the title screen of the program and the keys that you use are 'Z', 'X', '*', '?' for left, right, up and down, and the spacebar to jump up and remove a tile or jump up and hang the spider on a hook.

You can walk off the side of each walkway, reappearing on the other side of the screen. This is useful for escaping from the spiders but any floor tiles that you are carrying will disintegrate in the process.

This game requires a lot of cunning and strategic planning if you are to be successful in your self-appointed task. Don't be misled either by the arcade style of screen. Good thinking will succeed much better than fast action here.



```
  10 REM PROGRAM SPIDER
  20 REM VERSION B0.2
  30 REM AUTHOR  R.LEWIS
  40 REM BEEBUG  MARCH 1985
  50 REM PROGRAM SUBJECT TO COPYRIGHT
  60 :
 100 ON ERROR GOTO 220
 110 MODE5:VDU23,1,0;0;0;0;:VDU19,3,5,
0,0,0
 120 DIM BX%(6),BY%(6),BXI%(6),BYI%(6)
,A%300
 130 ?&D0=2:?A%=0:A%=A%+1:A%?147=10:NS
%=1:NH%=1:HSC%=0:VDU23;8202;0;0;0;:VDU2
3,242,255,255,24,24,24,255,129,129:VDU2
3,244,255,24,12,36,60,0,0,0
 140 PROCINST
 150 PROCSCORE:PROCINIT:REPEAT:TM%=TM%
+1:IF (TM%MOD256)<245 THEN C%=3:S%=148
ELSE C%=1:S%=S%-4:SOUND1,-15,S%,4
 160 PROCMAN:PROCBUG:UNTIL FIN%
 170 NS%=NS%+1:IF NS%<4 AND MEN%>0 THE
N GOTO 150
 180 IF NS%>3 THEN NH%=NH%+1:NS%=1
 190 IF MEN%<1 THEN PROCSCORE:NH%=1:ME
N%=3:NS%=1:SC%=0
 200 GOTO150
 210 :
 220 ON ERROR OFF
 230 MODE 7:?&D0=0:IF ERR=17 END
 240 REPORT:PRINT " at line ";ERL
 250 END
 260 :
1000 DEF PROCINIT
1010 CLS:COLOUR1:FORJ%=1TO6:FOR I%=0TO
19
1020 PRINTTAB(I%,J%*5+1);CHR$242;
1030 K%=I%+1+J%*21+A%:?K%=0:IF J%=1 TH
EN K%?-21=12:NEXT ELSE NEXT
1040 K%=J%*21+A%:?K%=10:NEXT
1050 FIN%=FALSE:IX%=0:TM%=0
1060 VDU23,255,255,255,255,255,255,255
,255,255
```




```
1070 VDU23,243,129,255,129,129,129,255
,129,129
1080 COLOUR2:FOR J%=1 TO 5:FOR H%=1 TO
NH%+1
1090 I%=RND(17)+1:K%=I%+1+J%*21+A%
1100 IF ?K%>7 OR K%?-1>7 OR K%?1>7 THE
N GOTO 1090
1110 IF H%=1 THEN K%?21=8:?K%=9:PROCLA
DDER(I%,J%*5+6) ELSE ?K%=7:PRINTTAB(I%,
J%*5+1);" ";
1120 NEXT
1130 NEXT
1140 COLOUR3:VDU23,240,36,36,36,60,126
,102,66,195
1150 FOR II%=1 TO NS%
1160 COLOUR2:PRINTTAB(8-NS%+II%*2,0)CH
R$244;:K%=9-NS%+II%*2+A%:?K%=11
1170 I%=RND(17)+1:J%=RND(5):K%=I%+1+J%
*21+A%:IF ?K%>0 THEN GOTO 1170
1180 BX%(II%)=I%:BY%(II%)=J%*5:BXI%(II
%)=1:BYI%(II%)=0:?K%=II%
1190 COLOUR3:PRINTTAB(I%,J%*5)CHR$240
1200 NEXT
1210 M%=0:IM%=0:IB%=0:X%=0:Y%=30:V%=0:
B%=0:D%=0:II%=0:PROCMANCH:PROCMAKE:E$=E
2$:M$=M2$
1220 ENDPROC
1230 :
1240 DEF PROCPMAN
1250 VDU5:MOVE X%*64,(32-Y%)*32-4:GCOL
0,3:PRINT CHR$230
1260 MOVE X%*64,(33-Y%)*32-4:GCOL 0,1:
PRINT CHR$231
1270 IF B%>0 THEN MOVE X%*64,(34-Y%)*3
2-4:GCOL 0,1:PRINT CHR$242:ELSE GOTO 12
90
1280 IF D%>0 THEN MOVE X%*64,(35-Y%)*3
2-4:GCOL 0,3:PRINT CHR$240
1290 VDU 4:ENDPROC
1300 :
1310 DEF PROCBUGCH
1320 IF IB%=1 THEN VDU23,240,195,66,36
,60,60,102,66,102:ENDPROC
1330 VDU23,240,36,36,36,60,126,102,66,
195:ENDPROC
1340 :
1350 DEF PROCPBUG
1360 VDU5:MOVE BX%(II%)*64,(32-BY%(II%
))*32-4:GCOL 0,C%:PRINT CHR$240:VDU4
1370 ENDPROC
1380 :
1390 DEF PROCLADDER(P%,Q%)
1400 LD$=CHR$243+CHR$8+CHR$11
1410 COLOUR 2:PRINTTAB(P%,Q%-1);STRING
$(5,LD$)
1420 ENDPROC
1430 :
1440 DEFPROCMANCH
1450 ON 2*M%+IM%+1 GOTO 1460,1470,1480
,1490,1500,1510,1460,1470
1460 IF B%=0 THEN VDU23,230,189,189,60
,36,36,36,100,6:GOTO 1520 ELSE VDU23,23
0,60,60,60,36,36,36,100,6:GOTO 1520
1470 IF B%=0 THEN VDU23,230,189,189,60
,36,36,36,38,96:GOTO1520 ELSE VDU23,230
,60,60,60,36,36,36,38,96:GOTO1520
1480 VDU23,230,60,60,24,24,28,22,19,50
:GOTO 1520
1490 VDU23,230,60,60,24,24,56,232,72,2
4:GOTO 1520
1500 VDU23,230,60,60,24,24,56,104,200,
76:GOTO 1520
1510 VDU23,230,60,60,24,24,28,23,18,24
:GOTO 1520
1520 ON M%+B%+1 GOTO 1530,1540,1550,15
60,1570,1580,1590,1600
1530 VDU23,231,60,90,126,60,36,126,255
,189:GOTO 1610
1540 VDU23,231,28,6,62,20,28,60,126,18
9:GOTO 1610
1550 VDU23,231,28,48,62,20,28,60,126,1
89:GOTO 1610
1560 VDU23,231,60,126,126,60,60,126,25
5,189:GOTO 1610
1570 VDU23,231,189,219,255,189,165,126
,60,60:GOTO 1610
1580 VDU23,231,157,135,191,149,157,126
,60,60:GOTO 1610
1590 VDU23,231,157,177,191,149,157,126
,60,60:GOTO 1610
1600 VDU23,231,189,255,255,189,189,126
,60,60:GOTO 1610
1610 ENDPROC
1620 :
1630 DEF PROCMAKE
1640 M2$=CHR$17+CHR$3+CHR$230+CHR$8+CH
R$11+CHR$17+CHR$1+CHR$231
1650 M3$=M2$+CHR$8+CHR$11+CHR$242
1660 M4$=M3$+CHR$8+CHR$11+CHR$17+CHR$3
+CHR$240
1670 E2$=CHR$32+CHR$8+CHR$11+CHR$32
1680 E3$=E2$+CHR$8+CHR$11+CHR$32
1690 E4$=E3$+CHR$8+CHR$11+CHR$32
```

```
 1700 ENDPROC
 1710 :
 1720 DEF PROCMAN
 1730 OM%=M%:IF IM%=0 THEN IM%=1 ELSE I
M%=0
 1740 K%=X%+1+(Y%DIV5)*21+A%:IF IX%<>0
OR Y%MOD5=0 THEN M%=0:IX%=0
 1750 IF INKEY-67 AND V%=0 AND Y%MOD5=0
 THEN IX%=1:M%=2
 1760 IF INKEY-98 AND V%=0 AND Y%MOD5=0
 THEN IX%=-1:M%=1
 1770 IF ?K%=7 AND V%=0 THEN V%=1:IX%=0
 1780 IF ((?K%>0 AND ?K%<7) OR (K%?IX%>
0 AND K%?IX%<7)) AND V%=0 THEN FIN%=TRU
E:MEN%=MEN%-1
 1790 IF ?K%=8 AND V%=0 THEN M%=3
 1800 IY%=0:L%=X%+1+((Y%+4)DIV5)*21+A%
 1810 T%=X%+1+((Y%-3)DIV5)*21+A%
 1820 IF INKEY-105 AND ?K%=9 AND IX%=0
THEN IY%=1:M%=3
 1830 IF INKEY-73 AND ?L%=8 AND IX%=0 T
HEN IY%=-1:M%=3
 1840 IF INKEY-99 AND V%=0 AND M%<>3 AN
D (B%=0 OR (B%=4 AND K%?-21>6 AND K%?-2
1<13 AND K%?-21<>8)) AND IX%=0 THEN V%=
-1:IY%=-1:SOUND1,-15,120,2:GOTO 1950
 1850 IF V%=0 THEN GOTO 1940 ELSE IF V%
>0 THEN V%=V%+1:IY%=1:M%=0:SOUND1,-15,(
7-V%)*20,4
 1860 IF V%>5 THEN V%=0
 1870 IF V%=-1 THEN IY%=-1:V%=-2:SOUND1
,-15,150,2:GOTO1950
 1880 IF V%=-3 THEN V%=0:IY%=1
 1890 IF V%=-2 AND B%=0 AND (?T%=0 OR (
?T%<7 AND C%=3)) THEN V%=-3:IY%=1:B%=4:
D%=?T%:?T%=7:BY%(D%)=0:SOUND1,1,100,5:G
OTO 1940
 1900 IF V%=-2 AND B%=4 AND ?T%=11 AND
D%>0 THEN V%=-3:IY%=1:?T%=13:SC%=SC%+10
*NH%:BY%(D%)=1:D%=0:SOUND1,1,100,20:GOT
O 1940
 1910 IF V%=-2 AND B%=4 AND ?T%=7 THEN
V%=-3:IY%=1:B%=0:?T%=D%:BY%(D%)=Y%-3:BX
%(D%)=X%:BXI%(D%)=1:D%=0:SOUND1,1,100,5
:GOTO 1940
 1920 IF V%=-2 THEN V%=-3:IY%=1:SOUND1,
-15,180,5
 1930 IF V%=5 AND B%=4 THEN B%=0:BX%(D%
)=X%:BY%(D%)=Y%-3:?T%=D%:BXI%(D%)=1
 1940 IF B%=0 THEN M$=M2$:E$=E2$ ELSE I
F D%=0 THEN M$=M3$:E$=E3$ ELSE M$=M4$:E
$=E4$
 1950 PROCMANCH:IF IY%=0 AND IX%=0 AND
OM%=M% GOTO 1980
 1960 PRINTTAB(X%,Y%);E$;
 1970 IF M%=3 PROCLADDER(X%,((Y%+5+(IY%
<>1))DIV5)*5+1):Y%=Y%+IY%:X%=X%+IX%:PRO
CPMAN:ENDPROC
 1980 IF M%=3 THEN ENDPROC ELSE Y%=Y%+I
Y%:X%=X%+IX%:IF X%<0 THEN X%=19 ELSE IF
 X%>19 THEN X%=0 ELSE GOTO 2000
 1990 IF B%=4 THEN B%=0:M$=M2$:SOUND 1,
1,100,10:IF D%<>0 THEN BX%(D%)=(X%+19)M
OD 38:BY%(D%)=Y%
 2000 IF K%?IX%=8 AND V%=0 THEN M%=3:PR
OCMANCH:PROCPMAN ELSE PRINTTAB(X%,Y%);M
$;
 2010 ENDPROC
 2020 :
 2030 DEF PROCBUG
 2040 IF II%<NS% THEN II%=II%+1 ELSE II
%=1:BC%=0:PROCBUGCH:IF IB%=0 THEN IB%=1
 ELSE IB%=0
 2050 IF BY%(II%)=0 THEN GOTO 2250 ELSE
 IF BY%(II%)<>1 THEN GOTO 2070
 2060 BC%=BC%+1:IF BC%=NS% THEN FIN%=1:
GOTO 2250 ELSE GOTO 2250
 2070 K%=BX%(II%)+1+(BY%(II%)DIV5)*21+A%
 2080 IF BXI%(II%)>-1 AND (K%?1=8 OR K%
?1=9) THEN GOTO 2100 ELSE IF BXI%(II%)>
-1 AND K%?1<>0 THEN BXI%(II%)=-1
 2090 IF BXI%(II%)<1 AND (K%?-1=8 OR K%
?-1=9) THEN GOTO 2100 ELSE IF BXI%(II%)
<1 AND K%?-1<>0 THEN BXI%(II%)=1
 2100 IF C%=3 GOTO 2130
 2110 IF K%?1=7 THEN BXI%(II%)=1:K%?1=0
:GOTO 2120 ELSE IF K%?-1=7 THEN BXI%(II
%)=-1:K%?-1=0:GOTO 2120 ELSE GOTO 2130
 2120 COLOUR1:PRINTTAB(BX%(II%)+BXI%(II
%),BY%(II%)+1)CHR$242
 2130 IF K%?1=0 OR K%?1=8 OR K%?1=9 OR
K%?-1=0 OR K%?-1=8 OR K%?-1=9 THEN GOTO
 2140 ELSE BXI%(II%)=0
 2140 IF ?K%=8 OR (?K%=9 AND BY%(II%)MO
D5<>0) THEN E1$=CHR$17+CHR$2+CHR$243 EL
SE E1$=" "
 2150 IF BYI%(II%)=0 OR BY%(II%)MOD5<>0
 THEN GOTO 2160 ELSE BYI%(II%)=0:IF K%?
1<>7 THEN BXI%(II%)=1 ELSE IF K%?-1<>7
THEN BXI%(II%)=-1
 2160 IF ?K%=9 AND BY%(II%)<Y% THEN BYI
%(II%)=1:BXI%(II%)=0 ELSE IF ?K%=8 AND
BY%(II%)>Y% THEN BYI%(II%)=-1:BXI%(II%)
=0
 2170 IF ?K%<>8 AND ?K%<>9 AND BXI%(II%
)<>0 THEN ?K%=0
 2180 IF BXI%(II%)=0 AND BYI%(II%)=0 TH
EN GOTO 2240 ELSE IF BXI%(II%)<>0 AND B
YI%(II%)<>0 THEN BXI%(II%)=0
 2190 IF BXI%(II%)=1 AND K%?1=0 THEN K%
?1=II%
 2200 IF BXI%(II%)=-1 AND K%?-1=0 THEN
K%?-1=II%
 2210 PRINTTAB(BX%(II%),BY%(II%));E1$;
 2220 BX%(II%)=BX%(II%)+BXI%(II%)
 2230 BY%(II%)=BY%(II%)+BYI%(II%):IF BX
%(II%)=X% AND (BY%(II%)=Y% OR BY%(II%)=
Y%-1) THEN FIN%=TRUE:MEN%=MEN%-1
 2240 IF BYI%(II%)<>0 OR K%?BXI%(II%)=8
 THEN PROCPBUG ELSE COLOUR C%:PRINTTAB(
BX%(II%),BY%(II%))CHR$240;
 2250 ENDPROC
```

```
 2260 :
 2270 DEF PROCINST
 2280 CLS:COLOUR3:PRINTTAB(4,2)"SPIDER
CHASE";
 2290 PRINTTAB(0,6)"Catch the Spider";
 2300 M%=0:IM%=0:B%=4:PROCMANCH:PROCMAK
E:PRINTTAB(17,9)M4$;
 2310 PRINTTAB(0,12)"Place it on the ho
ok";
 2320 COLOUR2:PRINTTAB(16,14);CHR$244;"
";CHR$244;
 2330 COLOUR3:PRINTTAB(3,16)"CONTROLS";
 2340 COLOUR1:PRINTTAB(3,18)"Z  -  LEFT
";
 2350 PRINTTAB(3,20)"X  -  RIGHT";
 2360 COLOUR2:PRINTTAB(3,22)"*  -UP THE
LADDER";
 2370 PRINTTAB(3,24)"?  -DOWN LADDER";
 2380 COLOUR1:PRINTTAB(1,26)"SPACE-  JU
MP";
 2390 COLOUR3:PRINTTAB(1,30)"RETURN TO
CONTINUE";
 2400 REPEAT UNTIL INKEY-74
 2410 SC%=0:MEN%=3:ENVELOPE 1,1,6,0,-6,
200,100,200,100,2,0,-1,120,110
 2420 ENDPROC
 2430 :
 2440 DEF PROCSCORE
 2450 M%=0:B%=0:CLS:COLOUR3:PRINTTAB(4,
2)"SCORE SHEET";
 2460 IF MEN%<1 THEN  COLOUR3:PRINTTAB(
4,6)"GAME OVER";:GOTO  2490
 2470 PRINTTAB(3,10)"MEN LEFT";
 2480 PROCMANCH:FOR I%=1 TO MEN%:PRINTT
AB(10+2*I%,10)M2$:NEXT:GOTO 2520
 2490 PRINTTAB(2,10)"FINAL SCORE ";SC%;
 2500 IF SC%>HSC% THEN HSC%=SC%:COLOUR2
:PRINTTAB(3,14)"NEW HIGH SCORE";:PRINTT
AB(10,16);HSC%;
 2510 COLOUR1:PRINTTAB(4,20)"PRESS RETU
RN";:PRINTTAB(3,22)"FOR A NEW GAME";:GO
TO 2550
 2520 COLOUR2:PRINTTAB(2,6)"Score so fa
r ";SC%;
 2530 COLOUR3:PRINTTAB(1,16)"Sheet Numb
er - ";NH%;:PRINTTAB(6,18)"Spiders - ";
NS%;
 2540 PRINTTAB(2,20)"Each Spider - ";10
*NH%;:COLOUR2:PRINTTAB(1,26)"RETURN TO
CONTINUE";
 2550 BX%=1:BI%=1:M%=2:FOR I%=0 TO 19
 2560 COLOUR 2:PRINTTAB(I%,0)CHR$244:CO
LOUR1:PRINTTAB(I%,31)CHR$242;:NEXT
 2570 FOR I%=1 TO 10
 2580 COLOUR1:PRINTTAB(0,1+I%*3)CHR$242
;:PRINTTAB(19,1+I%*3)CHR$242;:NEXT
 2590 REPEAT
 2600 COLOUR3:FOR I%=1 TO 10:PRINTTAB(0
,I%*3)CHR$240;:PRINTTAB(19,(11-I%)*3)CH
R$240;:NEXT
 2610 PRINTTAB(BX%,30)E2$
 2620 BX%=BX%+BI%:IF BX%=0 THEN BX%=1:B
I%=1:M%=2 ELSE IF BX%=19 THEN BX%=18:BI
%=-1:M%=1
 2630 PRINTTAB(BX%,30)M2$;
 2640 IF IM%=0 THEN IM%=1:IB%=1 ELSE IM
%=0:IB%=0
 2650 PROCMANCH:PROCBUGCH:UNTIL INKEY-74
 2660 ENDPROC
```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

ALTERNATIVE REM

As an alternative to the REM statement you can use *|. The Escape character (|) is taken by the Operating System to terminate the * command, so anything following it is ignored. For example:

```
10 REM this is a remark
20 *| so is this
```

PRINT FORMATTING

Unlike most Basics, BBC Basic does not leave the cursor on the same line following a comma after the last item in a PRINT statement. For example:

```
FOR I%=1 TO 4:PRINT I%,;:NEXT
```

Both the comma and semicolon have to be used to ensure that the numbers are not only spaced out, one to each field, but also that no new line is started with each PRINT.

IRRETRIEVABLE CRASH - Andrew Reynolds

*FX247,76 will cause a total system crash when either Break or Ctrl-Break is pressed. Only switching off the machine will restore control. *FX247 (and *FX248 and 249) redirect the Break vector. 76 is the machine code JMP instruction so after executing this call Break causes a jump to a totally irrelevant location.

IF YOU WRITE TO US

BACK ISSUES (Members only)

All back issues are kept in print (from April 1982) priced as follows:

Individual copies:
- Volume 1 - £0.80
- Volume 2 - £0.90
- Volume 3 - £1.00

Volume 1 set (10 issues) £7
Volume 2 set (10 issues) £8

Please add cost of post and packing as shown:

No of copies	DESTINATION UK	Europe	Elsewhere
1	0.30	0.70	1.50
2 - 5	0.50	1.50	4.70
6 - 10	1.00	3.00	5.50
11 - 20	1.50	4.00	7.00

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

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

SUBSCRIPTIONS

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

MEMBERSHIP COSTS:

U.K.
£6.40 for 6 months (5 issues)
£11.90 for 1 year (10 issues)

Eire and Europe
Membership £18 for 1 year.

Middle East £21
Americas and Africa £23
Elsewhere £25

Payment in Sterling essential.

PROGRAMS AND ARTICLES

All programs and articles used are paid for at up to £40 per page, but please give us warning of anything substantial that you intend to write. In the case of material longer than a page, we would prefer this to be submitted on cassette or disc in machine readable form using "Wordwise", "View", "Minitext Editor" or other means. If you use cassette, please include a backup copy at 300 baud.

HINTS

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

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

Editorial Address

BEEBUG
PO Box 50
St Albans
Herts

Subscriptions & Software Address

BEEBUG
PO BOX 109
High Wycombe
Bucks HP10 8HQ

Hotline for queries and software orders

St.Albans (0727) 60263
Manned Mon-Fri 9am-4.30pm

24hr Answerphone Service for Access orders, and subscriptions

Penn (049481) 6666

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

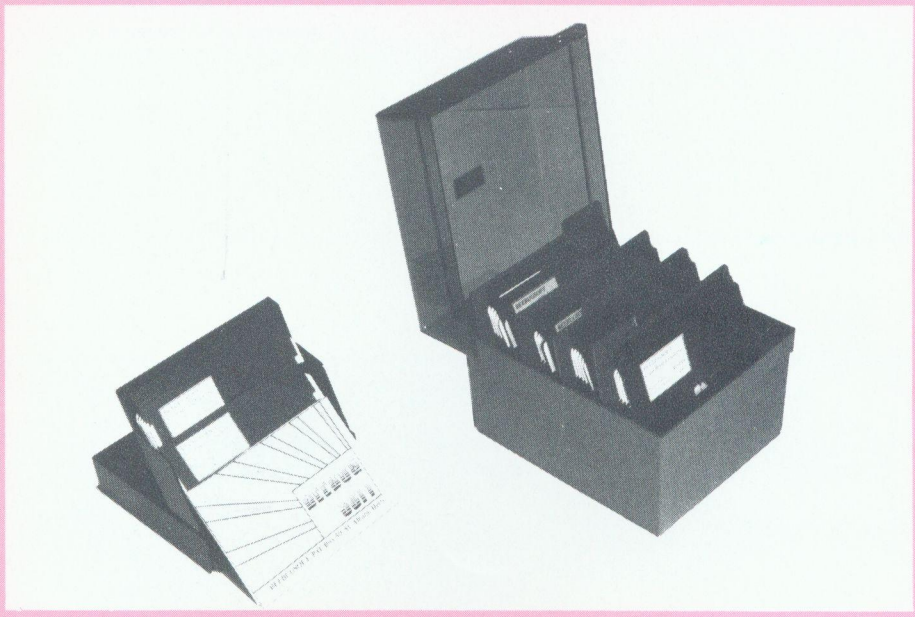
BEEBUG MAGAZINE is produced by BEEBUG Publications Ltd.
Editor: Mike Williams.
Assistant Editor: Geoff Bains. Production Editor: Phyllida Vanstone.
Technical Assistant: Alan Webster.
Secretary: Debbie Sinfield
Managing Editor: Lee Calcraft.
Thanks are due to Sheridan Williams, Adrian Calcraft, Matthew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

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

BEEBUG Publications Ltd (c) 1985.

High Quality Low Priced Discs

Backed by The Reputation of BEEBUG



10 S/S D/D Discs – £13.90	10 D/S D/D Discs – £19.40
25 S/S D/D Discs – £33.45	25 D/S D/D Discs – £46.95
50 S/S D/D Discs – £59.30	50 D/S D/D Discs – £87.05

All Prices Include Storage Box, VAT and Delivery to Your Home (UK).

All discs are 100% individually tested, supplied with hub ring as standard, and guaranteed error free. They are ideal for use on the BBC Micro and have performed perfectly in extensive tests at BEEBUG over many months.

Orders for 25 or 50 are delivered in strong plastic storage boxes with four dividers. Orders for 10 are sent in smaller hinged plastic library cases.

We are also able to offer the empty storage container, which holds up to 50 discs for £10 including VAT and post.

Please use the order form enclosed
or order directly from:
BEEBUGSOFT, P.O. Box 109,
High Wycombe, Bucks HP10 8HQ.

BEEBUG SOFT

THE BEEBUG MAGAZINE ON DISC AND CASSETTE

The programs featured each month in the BEEBUG magazine are now available to members on disc and cassette.

Each month we will produce a disc and cassette containing all of the programs included in that month's issue of BEEBUG. Both the disc and the cassette will display a full menu allowing the selection of individual programs and the disc will incorporate a special program allowing it to be read by both 40 and 80 track disc drives. Details of the programs included in this month's magazine cassette and disc are given below.

Magazine cassettes are priced at £3.00 and discs at £4.75.
SEE BELOW FOR FULL ORDERING INFORMATION.

This Month's Programs Include:

Body Popping, a fascinating program displaying 3D representations of the human body, the first part of BEEBUG's own Spreadsheet program (a complete working program as it stands), a utility to produce well structured listings of Basic programs, two more programs from our series on Making Music on the Beeb, a fast moving and addictive game called Caterpillar, two demonstrations of error trapping from this month's BEEBUG Workshop, a very original game called Spider Man requiring both fast reactions and much thought to succeed, and the winning program from the last Brainteaser competition 'Sum Squares'.

MAGAZINE DISC/CASSETTE SUBSCRIPTION

Subscription to the magazine cassette and disc is also available to members and offers the added advantage of regularly receiving the programs at the same time as the magazine, but under separate cover.

Subcription is offered either for a period of 6 months (5 issues) or 1 year (10 issues) and may be backdated if required. (The first magazine cassette available is Vol. 1 No. 10; the first disc available is Vol. 3 No. 1.)

MAGAZINE CASSETTE SUBSCRIPTION RATES

6 MONTHS	(5 issues)	UK £17.00 INC... Overseas £20.00 (No VAT payable)
1 YEAR	(10 issues)	UK £33.00 INC... Overseas £39.00 (No VAT payable)

MAGAZINE DISC SUBSCRIPTION RATES

6 MONTHS	(5 discs)	UK £25.50 INC... Overseas £30.00 (No VAT payable)
1 YEAR	(10 discs)	UK £50.00 INC... Overseas £56.00 (No VAT payable)

CASSETTE TO DISC SUBSCRIPTION TRANSFER

If you are currently subscribing to the BEEBUG magazine cassette and would prefer to receive the remainder of your subscription on disc, it is possible to transfer the subscription. Because of the difference between the cassette and disc prices, there will be an extra £1.70 to pay for each remaining issue of the subscription. Please calculate the amount due and enclose with your order.

ORDERING INFORMATION

Please send your order to the address below and include a sterling cheque. Postage is included in subscription rates but please add 50p for the first item and 30p for each subsequent item when ordering individual discs or cassettes in the UK. Overseas orders please send the same amount to include the extra post but not VAT.
SEND TO:

BEEBUGSOFT, PO BOX 109, HIGH WYCOMBE, BUCKS, HP10 8HQ

Printed in England by Staples Printers St Albans Limited at The Priory Press.

ISSN 0263-7561