Vol.12 No.4 August/September 1993

BEEBUG

FOR THE BBC MICRO & MASTER SERIES

- SPYSNATCHER
- THE IDEAS PROCESSOR
- HEARING TEST
- LOAN REPAYMENTS

BEEBUG Vol.12 No.4 August/September 1993

FEATURES

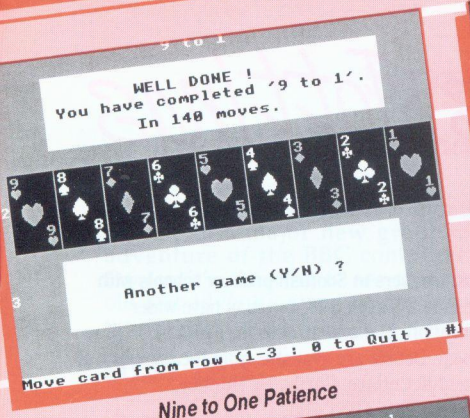
REGULAR ITEMS

HINTS & TIPS

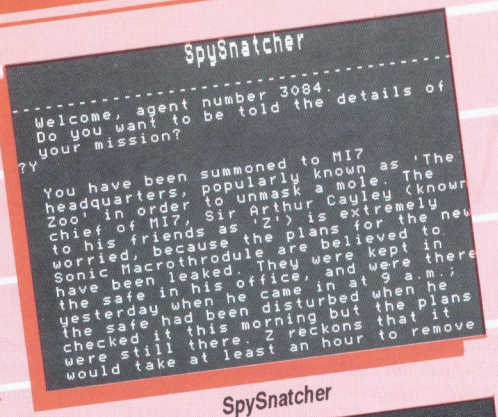
PROGRAM INFORMATION

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

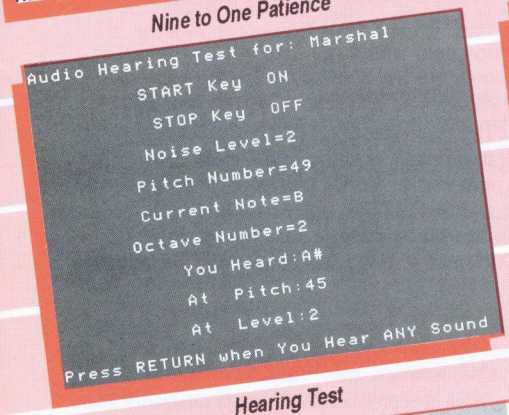
All programs in BEEBUG magazine will run on any BBC micro with Basic II or later, unless otherwise indicated. Members with Basic I are referred to the article on page 44 of BEEBUG Vol.7 No.2 (reprints



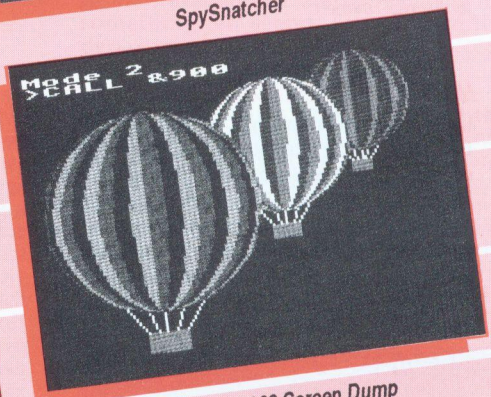
Nine to One Patience



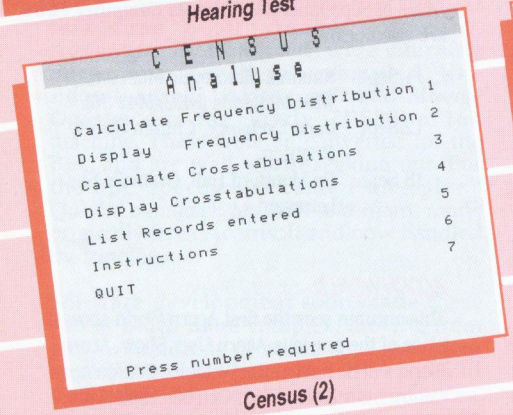
SpySnatcher



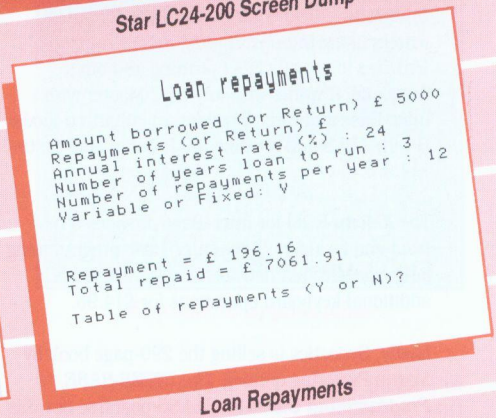
Hearing Test



Star LC24-200 Screen Dump



Census (2)



Loan Repayments

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

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

Program needs at least one bank of sideways RAM.

Program is for Master 128 and Compact only.

News/Editor's Jottings

This month, for a change, we are able to bring you news of a number of new products/developments for the BBC market.

NEW PRODUCTS FOR WORDWISE/INTERBASE

Users of Computer Concepts' products for the BBC micro will be interested in a number of new offerings from Cheshire based Synectics. For letter writers using Wordwise Plus, ***Wordwise-Mail*** will add a wealth of new functions, including the facility to create its own database of names and addresses. At any time, entering the first few characters of a respondent's address enables Wordwise-Mail to pull out all the relevant details. The price is £9.95 on 40 track or 80 track disc.

Inter-Mail provides similar facilities for letter writers using Inter-Word, and includes ***Inter-Utilities*** to provide label printing and other functions. It works with any BBC Master with Inter-Base and Inter-Word (or an enhanced model B with ADFS and shadow RAM). Inter-Mail costs £14.95.

The ***XMenu*** ROM for Inter-Base provides a new front end for users of the Inter-Base programming language with ten new menu options and six additional keyboard operations for £14.95.

Lastly, Synectics is selling the 290-page book by Martin Pickering entitled ***The INTER-BASE Programming Guide*** for £14.95 post free (for this cheques should be payable to the author, M.T.Pickering). For more information on all of these products, please contact Synectics at 10 Bollin Close, Sandbach, Cheshire CW11 9TZ.

PUPIL ASSESSMENT RECORDS

We have received details of a Pupil Assessment Records system from Birkenhill Computing Services Ltd. The system has been designed to assist teachers in Scottish primary schools with access to a Master 128, but may have wider application. The system sells for just £14.95 plus £2 p&p. Details are available from the company at Birkenhill Cottage, Gartly, By Huntley, Aberdeenshire AB54 4RJ.

MAD RABBIT PUBLIC DOMAIN

This is another BBC PD library which has recently moved from Southport to Wakefield. Software is supplied in 80 track DFS format or ADFS format, on 5.25" or 3.5" at prices from £1.25 per title. A catalogue is available from Mad Rabbit PD, P.O.Box 4, Criggestone, Wakefield, West Yorkshire WF4 3XE.

ALL FORMATS COMPUTER FAIRS

Dates of September ***All Formats Computer Fairs*** are given below.

11 Sept	National Motorcycle Museum, West Midlands (NEC/M42 J6).
12 Sept	Corn Exchange, Church Street, Brighton.
18 Sept	De Montford Hall, Granville St., Leicester.

ACORN WORLD

This autumn sees the first Acorn World show in place of the previous Acorn User Show. Acorn World will be held at the Wembley Conference Centre from 29th to 31st October 1993. This show, organised by Acorn itself, promises to be a major event for all users of Acorn machines, and we will bring you the latest details in our October issue. Both BEEBUG and RISC Developments will have adjacent stands where we will be pleased to meet all BEEBUG readers. Make a note of the dates now.

M.W.

SpySnatcher

Marshal Anderson tries not to be economical with the truth.

Having seen, in the last few months, the release of a major new graphic based adventure of the BBC computer, *Blade Dancer*, it's cheering to see that the good old text adventure is not being ignored by writers and publishers.

FIRST THIS

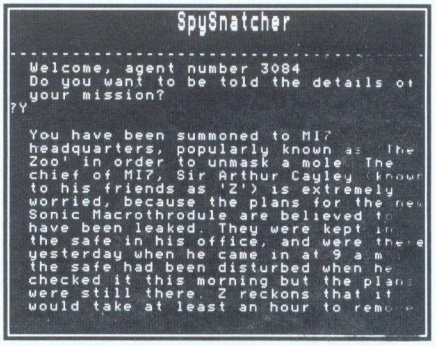
If you're a relative newcomer to computers, i.e. you've bought yours in the last five years or so, it's quite possible that you've not come across this kind of game before so here's a brief explanation. Text adventures grew up in the late seventies on the slow, text based machines of that time, owing a lot to the *Dungeons and Dragons* game produced by TSR about then. The structure is one of 'rooms' which are linked, usually via compass directions, and objects which can be used in some way. The player communicates with the software in something approaching normal English through a bit of the program called a parser. In the early games this was very limited, you could only type two word sentences like 'Go north', 'Take axe' and 'Kill troll'. At this time Scott Adams' company, Adventure International, were churning out some excellent examples like *Adventureland, Voodoo Castle* and *Strange Odyssey* for the Tandy TRS80, a 16k machine. These were all converted for the BBC and are well worth seeking out. For the BBC specifically came *Philosopher's Quest, Countdown to Doom* and many more, originally from Acornsoft and now handled by Topologica.

Software development soon made these games much more complex, moving from tens to hundreds of rooms and providing sophisticated parsers that could understand long and involved sentences. Two doyens of this latter type are Jon Thackray (*Acheton, Hezarin*) and Jonathan Partington (*Kingdom of Hamil, Avon*) and they've just come up with a new one.

SpySnatcher is based, however loosely, on a certain governmentally embarrass-ing but totally boring book published a few years ago. Fortunately you don't have to read the book to play the game.

The bulk of the game take place in what is euphemistically called The Zoo, it is the headquarters of MI7. The night before the our adventure takes place someone broke into the safe in Z's office, Z being the head of MI7, and copied the secret plans for something, er, secret. Everyone in the department is under suspicion, except you, and you must find the Mole.

One of your many problems is that the investigation is unofficial, you must not get caught moving about the building and you have just one hour before the guard comes on duty. Time is calculated on a one minute per move basis so you must make the best of your first 60 actions. Once the guard is around you must constantly watch out for him; discovery will end the investigation.



Welcome to SpySnatcher

The building you are in is quite large. I'm not sure I've been everywhere yet, and the descriptions are very detailed and atmospheric. One of the drawbacks here is that, because of memory limitations, the screen is in mode 7. This means that many of the long descriptions need Shift to be pressed to read them all. To get around this there are two levels of detail - you can switch between verbose and terse, or leave the setting at normal which gives a full description the first time you enter a room and a short one thereafter.

What's nice about the way the game is set up is that the bulk of the detective work takes in a comparatively enclosed environment. There aren't actually hundreds of rooms to map; it's the detail in the rooms that provides the interest. There are all sorts of desks to be opened, safes to be cracked, computers to interrogate, Dictafone tapes to be played and so on.

```
Mr Russell's office                    5
----------------------------------------
?SECRET
  A message on the screen says:

  XUG5001K INVALID LOGON DATA

  The time is about 4 minutes after 6.
:USE MULTIPOCS
  What user name?
?RUSSELL
  What password?
?ACCOUNTS
  A message on the screen says:

  XUG5001K INVALID LOGON DATA

  The time is about 5 minutes after 6.
:
```

Mr Russell's office

The game as a whole is a mixture of several genres. There's some fairly straight forward linking puzzles which hark back to the early days of adventures where you have to assemble a number of objects or find bits of information that lead to a new puzzle. There's plenty of humour. There are also some fairly cryptic, if somewhat contrived, clues which may well have you seeking help away from the computer. The most impressive part of the plot, though, is the sheer methodical detective work.

One of the problems with setting up any kind of investigation in an adventure format is dealing with characters, the software still has problems coping with all the sorts of questions players might want to ask. SpySnatcher gets round this really neatly by not having you question anyone, you must find your evidence through the clues that they have left behind them. Someone has been to the theatre, someone has left the country, but how do you know who they are and how can you verify these facts? It's all there, hidden in filing cabinets, word processors, waste bins - you have to look and take note.

The passage of time comes in to play too. You need to be in the right place before tapes are erased or the cleaner destroys vital evidence. While this is not in 'real time' you have to plan your moves carefully to make sure you don't miss anything.

The parser in this program is excellent which is a great help in the early stages. Apart from the usual two word commands you can do complicated things like 'Take everything but the note pad then go north'. Having said that, I always revert to the two word approach as it is usually the most predictable - if something goes wrong you know exactly what it was. You can, of course, save your position at (almost) any time which helps you try things out.

```
Mr Newton's office                     5
----------------------------------------
  marked PLAY, RECORD, REWIND, FORWARD,
  STOP and EJECT. Currently the tape is
  recording. As there is no input (such
  as a microphone) it is being erased.

  The time is about 17 minutes after 6.
:STOP TAPE
  OK.

  The time is about 18 minutes after 6.
:REWIND TAPE
  The battery seems to be a bit flat and
  the tape recorder won't do that.

  The time is about 19 minutes after 6.
:
```

Mr Newton's office

To save on the frustration level there is on-line help throughout the game, though it can only be reached via the manual to stop you using it at every turn; it is, after all, cheating. The text is held on disc which provides such a wealth of description; it is accessed very quickly and I never found it irritating.

If you're looking for an alternative to all those bright colours and loud noises SpySnatcher has to be for you. It'll keep you busy for weeks and you'll get a real feeling of satisfaction when you complete it.

Product: SpySnatcher
Supplier: Topologica
PO Box 39,
Stilton,
Peterborough PE7 3RL.
Tel/fax 0733 244682
Price: £15 inc. VAT

The Ideas Processor

Gil Swain takes a creative look at content-free software.

In my previous articles (BEEBUG Vol. 11 No. 4 etc.) I have described ways of using any text editor or word processor to perform tasks, normally the province of specially written software. Whilst most computer owners and users try to obtain the best software available with which to perform a given task, it is not always desirable or possible to purchase new software. Given that we have been going through a period of recession the reality for most of us is that it will be a while before we can afford to splash out. This would be as true for owners of BBC or Archimedes machines, or even PC's. One solution would be to explore public domain and shareware, but even this is not entirely cost free, since we are under a moral obligation to register the software and contribute towards the author's expenses.

An alternative strategy is to employ such programs as may already be available. Sometimes software is supplied with the machine. The Master comes with Editor, View and Viewsheet. The Archimedes and its descendants are supplied with a text editor, Draw and Paint programs and many PC's are bundled with software, such as Microsoft Works. Then again we may have added to our original equipment purchases such essential items as database programs and spreadsheets.

In the two previous articles I suggested how a humble text editor or simple word processor like View or Wordwise could be used for 'Freetext' databases and decision making. In this article I hope to demonstrate how a spreadsheet could be employed as an outliner or ideas processor.

To some readers the practice of outlining or planning projects will be second nature. To others the theory and practise will be a new experience. Computer outliners have been available for sometime as PC Shareware, and the idea has already been successfully transported to the Archimedes. Now it is the turn of the Model B and Master machines.

There are many occasions when an exercise, like the compilation of a magazine article, needs to include a great deal of detail, but also requires to be presented in a logical sequence. On such occasions some kind of aid to planning is desirable. You could of course resort to pen and paper, but by utilising some kind of computer software, you are provided with a great deal more flexibility. For instance after grouping some ideas together, it might make more sense to list the information in a different sequence with items added or removed. The ability to move items electronically or to repeat them, or to establish links and cross references could save a lot of time and effort, not to mention ink and paper.

The computerised 'Ideas Processor' can be applied to any activity requiring detailed planning. In this article I will show how it might have been applied in the article's preparation. If we look at the traditional process of planning, the usual practice would be to write a brief precis of the major ideas, then append notes to the precis to ensure that the salient points are covered. Finally the resulting skeleton would be used as a guide to final composition.

When a computer is employed instead of pen and paper, it does not matter how the various elements are assembled, because we can always instruct the computer to move the various elements from one place to another, as many times as required until we are satisfied with

the result. This would be as true for this article as for any other kind of writing, whether a book, a set of instructions or even less complex tasks. With a word processor it is simplicity itself to add or move items around, or even to delete them. We can even grade ideas according to their importance, and it then becomes possible to re-assess the relative importance of different components.

Let me give a brief example: I started this article with the sentence, "In my previous articles I described ways of using a text editor or word processor to perform tasks, normally the province of specially written software".

I could have identified this in my outline as 'Introduction' and under this heading I could have included subheadings identifying the contents of this first sentence as 'Looking Back' and 'Improvisation' I could then have attached comments to these last items, and end up with a list looking like this:

Introduction
  Looking Back
    Previous Articles
  Improvisation
    Avoiding Special Software

With this trivial example I hope I have demonstrated by the use of indentation how ideas can be ranked in importance. If, on reflection, I decided that an idea was more or less important I could change the emphasis by promoting an item up one or more levels. Alternatively an idea can be demoted or removed altogether with the result that our example might now appear:

Looking Back
  Previous Articles
    Title 1
      Title 2
  Introduction
    Improvisation
      Special Software
        Etc.

At this point some readers will be asking about the significance of using a spreadsheet in preference to a word processor. The answer is to do with the potential size of the resulting plan or outline, coupled with the ability to make use of some of the facilities normally found in spreadsheets. In particular the ability to copy, move or replicate the contents of spreadsheet cells. It is also possible to manipulate cell widths temporarily whilst still working on the outline.

ACTUALLY DOING IT

It is probably easier to put this into practice than to write describing the method, and a lot will depend on the nature of the spreadsheet employed. I actually did my research using Ultracalc, the program with which I am most familiar. I then tried with varying success to use Acorn Viewsheet and Gemini Bcalc. Viewsheet came as part of the Master ROM, I believe Gemini can still be purchased from Watford Electronics as part of a suite of programs that they used to bundle with the BBC Master. There is no reason why the following should not work with Intercalc, or other spreadsheet programs, providing they can handle text and manipulate column widths. If other facilities exist then I leave it to the reader to explore the idea further.

PRACTICALITIES

Let's take a close look at our first paragraph again. In planning this paragraph I might have created the following outline:

Introduction
  Previous Articles
  Get the Best
  Recession
  Different Machines
  One Solution
  Registration

I do not suggest that it will always be essential to delve into microscopic detail

but, purely as an exercise, if we had opted to make a more detailed plan then we might have possibly produced something like this:

Introduction
Previous Articles
Editor or WP
Bespoke tasks
Special software

I could then continue with a similar set of ideas for each of the remaining sentences. By this time our electronic page starts to look very cluttered and by the time we have completed the outline of the rest of the article it would be considerably worse. This is where we can use a spreadsheet to advantage. In the illustration I have set out the beginnings of a worked example. Within each column we can be as brief or as detailed as we choose.

	A	B	C THE SPREADSHEET	D OUTLINER
1				
2				
3	Introduction	Previous Articles	Editor or WP	Idea 1
4				Idea 2
5				
6			Bespoke Tasks	Idea 1
7				Idea 2
8				
9		Get the best	Special Software	Note 1
10				Note 2
11				
12		Recession	Best Available	Etc.
13			Desirability	
14			Possibility	
15				
16		Different Machines	Shortage of funds	
17			Delays	
18				
19		One Solution	BBC	
20			Archimedes	
21			PC	
22				
23		Registration	Public Domain	
24			Sharewear	
25			Moral Obligation	
26			Author's Expenses	
27				
28				
29	Copy the contents of cells B19 - D21 Inclusive to a new block			
30	of cells starting at cell A34			
31				
32				
33				
34	BBC	PD	Example 1	
35			Example 2	
36				
37		Shareware	Example 1	
38			Example 2	
39				

Using a spreadsheet as an ideas processor

A spreadsheet can either be regarded as a single very large work surface, or smaller sections can be widened and expanded. Most spreadsheets also allow for the insertion or deletion of rows and columns. Just suppose that during the planning stage I decided that the subject of other computers is deserving of a paragraph to itself then I can promote this topic up a level (in this case one column to the left), alternatively I can leave it where it is and having noted which cells are occupied by this subheading and its associated ideas I can use a formula elsewhere within the spreadsheet to replicate these ideas and expand on them for an additional paragraph if required.

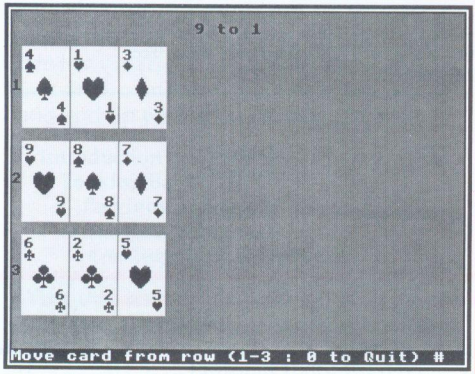
The amount of flexibility that can be exercised will of course depend on firstly having any kind of spreadsheet program, and secondly on the degree of sophistication of which the program is capable. I look at it this way: if you have the software then why not experiment? For instance, using the same tools it might be possible to produce flow charts. Admittedly the graphics might be a slight problem. On the other hand in the spirit of improvisation, all that it will cost is your time and effort.

Nine to One Patience

Leslie Fowl has a simple but effective way of keeping you awake all night.

The game in the listing below is a horizontal version of the patience game 'Tower of Pisa' and uses only nine cards, numbered 9 to 1, hence the name. It also includes the four suits, which are only of cosmetic use during play but are properly alternated upon the correct completion of the 9 to 1 row.

Type in the listing, save and run it. When the game is run the player is presented with a screen containing a three by three grid of cards, with an input message at the bottom. The object of the game is to form a single line of cards in the correct descending order from 9 to 1.



The game in progress

The player may only move one card at a time and only the card at the right hand end of the row. This card can only be placed at the end of another row next to a card of a higher value. An empty row may be filled with a card from either of the two remaining rows. Each entry is checked and rejected if the resulting move is illegal so it is not possible to cheat. The minimum number of moves possible is six, this obviously depends on the shuffling of the cards (I have honestly finished a game in 6 moves). The game usually comes out given a little thought, but it's very hard work until you've learned the tricks, so eyes down!

```
   10 REM Program 9to1
   20 REM version B.01
   30 REM Author  Leslie J Fowl
   40 REM BEEBUG  Aug/Sep 1993
   50 REM Program subject to copyright
   60 :
  100 IF PAGE=&E00 THEN 150
  110 CLS:PRINTTAB(5,10)"Downloading....
 "'TAB(5,12)"Please wait."
  120 VDU21:*KEY0 *TAPE|M FOR I%=0 TO TO
P-PAGE STEP 4:I%!&E00=I%!PAGE:NEXT I%:PA
GE=&E00|M OLD|M RUN|F|M
  130 *FX138,0,128
  140 END
  150 ON ERROR GOTO270
  160 *TV255,1
  170 PROCinit:PROCintro
  180 PROCshuff:MODE1
  190 PROCtable:PROCshuffle
  200 PROCdeal1(0):VDU4
  210 PRINT"Press any key to play....";
  220 IF GET PRINTclear$:PROCdeal2(0)
  230 REPEAT PROCgetnos
  240 UNTIL F%=9
  250 MODE7:PRINTTAB(0,6)"You have had "
;move%;" moves."TAB(0,10);"Would you lik
e another game (Y/N)";:G=INSTR("Nn",GET$
):IF G=0 THEN RUN ELSE VDU22,7,7:END
  260 :
  270 ON ERROR OFF
  280 MODE 7:REPORT
  290 PRINT" at line ";ERL:END
  300 :
 1000 DEF PROCgetnos
 1010 VDU4:PRINTclear$:VDU11:GCOL0,3
 1020 PRINT;"Move card from row (1-3 : 0
 to Quit) #";:from%=(GET-48)
```

```
 1030 IF from%>3 OR from%<0 THEN PRINTcl
ear$:VDU11:GOTO1020
 1040 IF from%=0 THEN F%=9:ENDPROC
 1050 PRINT;from%;
 1060 PRINT'"Move to row (1-3) #";:to%=(
GET-48)
 1070 IF to%=from% THEN PROCno(1):ENDPRO
C
 1080 IF R(from%)=0 THEN PROCno(2):ENDPR
OC
 1090 IF to%=0  THEN F%=9:ENDPROC
 1100 IF to%>3 OR to%<0 THEN PRINTclear$
:VDU11:GOTO1060
 1110 IF NOT FNvalid(to%,from%) THEN PRO
Cno(1):GOTO1020
 1120 PRINT;to%;
 1130 PROCwipe(from%):PROCmove(to%)
 1140 PROCupdate(from%,to%)
 1150 ENDPROC
 1160 :
 1170 DEF FNvalid(to%,from%)
 1180 IF VAL(LEFT$(row$(to%,R(to%)),1))=
0 OR VAL(LEFT$(row$(from%,R(from%)),1))<
 VAL(LEFT$(row$(to%,R(to%)),1)):=TRUE EL
SE =FALSE
 1190 :
 1200 DEF PROCwipe(x%)
 1210 xcord%=136*(R(x%)-1)+34
 1220 ycord%=900-(x%-1)*260
 1230 GCOL0,2:PROCblank(1)
 1240 ENDPROC
 1250 :
 1260 DEF PROCmove(to%)
 1270 row$(to%,R(to%)+1)=row$(from%,R(fr
om%))
 1280 xcord%=(136*R(to%))+34
 1290 ycord%=900-(to%-1)*260
 1300 GCOL0,3:PROCblank(1)
 1310 PROCsuits(from%,R(from%))
 1320 VDU5:SOUND0,-15,4,1
 1330 PROCcard(from%,R(from%),1)
 1340 row$(from%,R(from%))=""
 1350 VDU4:move%=move%+1:ENDPROC
 1360 :
 1370 DEF PROCupdate(f%,t%)
 1380 R(f%)=R(f%)-1:R(t%)=R(t%)+1
 1390 IF R(t%)=9 THEN PROCcheck(t%)
 1400 ENDPROC
 1410 :
 1420 DEF PROCcheck(t%)
 1430 FOR J%=1 TO 8
 1440 IF VAL(LEFT$(row$(t%,J%),1))<VAL(L
EFT$(row$(t%,J%+1),1)) THEN ENDPROC
 1450 NEXT:GCOL0,0
 1460 FOR S=200 TO 0 STEP-8:SOUND1,-10,S
,1:SOUND2,-15,200-S,1:SOUND3,-15,4*S,1:N
EXT
 1470 X%=160:IF R(1)=0 AND R(2)=0 THEN Y
%=900:Y2%=640:GOTO1500
 1480 IF R(2)=0 AND R(3)=0 THEN Y%=640:Y
2%=380:GOTO1500
 1490 IF R(1)=0 AND R(3)=0 THEN Y%=900:Y
2%=380
 1500 MOVEX%,Y%:MOVEX%+964,Y%:PLOT85,X%,
Y%-200:MOVEX%+964,Y%-200:PLOT85,X%+964,Y
%
 1510 MOVEX%+260,Y%-48:VDU5
 1520 GCOL0,3:PRINT"WELL DONE !"
 1530 MOVEX%+32,Y%-90:PRINT"You have com
pleted '9 to 1'."
 1540 MOVEX%+260,Y%-148:PRINT"In ";move%
;" moves."
 1550 GCOL0,0
 1560 MOVEX%,Y2%:MOVEX%+964,Y2%:PLOT85,X
%,Y2%-200:MOVEX%+964,Y2%-200:PLOT85,X%+9
64,Y2%
 1570 GCOL0,3:MOVEX%+160,Y2%-100
 1580 PRINT"Another game (Y/N) ?";:A=INS
TR("YyNn",GET$)
 1590 IF A>0 AND A<3 THEN RUN ELSE VDU22
,7,7:END
 1600 :
 1610 DEF PROCno(n%)
 1620 PRINTclear$:PRINTclear$
 1630 VDU11:COLOUR1
 1640 IF n%=1:PRINT"Illegal move...."
 1650 IF n%=2:PRINT"There are no cards i
n row ";from%;" to move."
 1660 PRINT"Press space bar.";:VDU7:COLO
UR3
 1670 REPEAT UNTIL GET=32
 1680 PRINTclear$:VDU11:ENDPROC
```

```
1690 :
1700 DEF PROCtable
1710 VDU28,0,31,39,30
1720 VDU24,0;70;1279;1023;
1730 VDU19,2,2,0,0,0
1740 GCOL0,130:CLG
1750 CLS:GCOL0,0:VDU5
1760 MOVE512,964:PRINT;"9 to 1"
1770 FOR I%=1 TO 3
1780 MOVE2,1064-(I%*256)
1790 PRINTSTR$(I%):NEXT
1800 GCOL0,2:VDU4
1810 ENDPROC
1820 :
1830 DEF PROCinit
1840 card$="1H2C3D4S5H6C7D8S9H"
1850 DIM row$(3,9),R(3)
1860 clear$=STRING$(38," "):F%=0:move%=
0
1870 VDU23,230,204,204,51,51,204,204,51
,51
1880 VDU23,231,8,28,62,127,62,28,8,0
1890 VDU23,232,54,127,127,127,62,28,8,0
1900 VDU23,233,8,28,28,107,127,107,8,28
1910 VDU23,234,8,28,62,127,127,127,28,6
2
1920 VDU23,235,1,3,3,7,7,15,15,31
1930 VDU23,236,0,128,128,192,192,224,22
4,240
1940 VDU23,237,31,15,15,7,7,3,3,1
1950 VDU23,238,240,224,224,192,192,128,
128,0
1960 VDU23,239,24,124,126,255,255,255,2
55,255
1970 VDU23,240,48,124,252,254,254,254,2
54,254
1980 VDU23,241,255,127,127,63,31,15,7,3
1990 VDU23,242,254,252,252,248,240,224,
192,128
2000 VDU23,243,3,7,15,15,15,7,59,125
2010 VDU23,244,128,192,224,224,224,192,
184,124
2020 VDU23,245,255,255,255,125,57,1,3,7
2030 VDU23,246,254,254,254,124,56,0,128
,192
2040 VDU23,247,1,3,7,15,15,31,31,63
2050 VDU23,248,0,128,192,224,224,240,24
0,248
2060 VDU23,249,63,63,31,31,13,1,3,7
2070 VDU23,250,248,248,240,240,96,0,128
,192
2080 D$=CHR$(231):H$=CHR$(232)
2090 C$=CHR$(233):S$=CHR$(234)
2100 NL$=CHR$(8)+CHR$(8)+CHR$(10)
2110 D2$=CHR$(235)+CHR$(236)+NL$+CHR$(2
37)+CHR$(238)
2120 H2$=CHR$(239)+CHR$(240)+NL$+CHR$(2
41)+CHR$(242)
2130 C2$=CHR$(243)+CHR$(244)+NL$+CHR$(2
45)+CHR$(246)
2140 S2$=CHR$(247)+CHR$(248)+NL$+CHR$(2
49)+CHR$(250)
2150 ENDPROC
2160 :
2170 DEF PROCshuffle
2180 f$="":D%=9:FORI%=1TO9
2190 A%=INT(RND(1)*D%+1)
2200 f$=f$+MID$(card$,2*A%-1,2)
2210 L$=LEFT$(card$,(A%-1)*2)
2220 R$=RIGHT$(card$,(LEN(card$)/2-A%)*
2)
2230 card$=L$+R$:D%=D%-1
2240 NEXT I%:card$=f$
2250 FOR c%=1TO3:FOR r%=1TO3
2260 row$(c%,r%)=LEFT$(card$,2)
2270 card$=RIGHT$(card$,LEN(card$)-2)
2280 NEXT r%:NEXT c%
2290 ENDPROC
2300 :
2310 DEF PROCdeal1(b%)
2320 FOR card%=1 TO 9:SOUND0,-15,5,2
2330 IF card%=1 OR card%=4 OR card%=7 T
HEN xcord%=34 ELSE IF card%=2 OR card%=5
 OR card%=8 THEN xcord%=170 ELSE IF card
%=3 OR card%=6 OR card%=9 THEN xcord%=30
6
2340 IF card%>=1 AND card%<=3 THEN ycor
d%=900 ELSE IF card%>=4 AND card%<=6 THE
N ycord%=640 ELSE IF card%>=7 AND card%<
=9 THEN ycord%=380
2350 R(1)=3:R(2)=3:R(3)=3
2360 GCOL0,3:VDU5:PROCblank(0):NEXT:END
PROC
2370 :
```

```
 2380 DEF PROCblank(b%)
 2390 MOVE xcord%,ycord%:MOVExcord%+128,
ycord%:PLOT85,xcord%,ycord%-218
 2400 MOVE xcord%,ycord%-218:MOVExcord%+
128,ycord%-218:PLOT85,xcord%+128,ycord%
 2410 IFb%=1 THEN ENDPROC
 2420 GCOL0,1:FORx%=0TO2:FORy%=0TO5:MOVE
xcord%+16+x%*32,ycord%-16-y%*32:PRINTCHR
$(230);:NEXT:PRINT:NEXT
 2430 ENDPROC
 2440 :
 2450 DEF PROCintro
 2460 VDU22,7:FOR I%=0 TO 1:PRINTTAB(6,I
%);CHR$(129);CHR$(157);CHR$(135);CHR$(14
1);"    9 to 1      ";CHR$(156):NEXT
 2470 PRINTTAB(6);CHR$(135);CHR$(157);CH
R$(129);"Author L.J.Fowl. ";CHR$(156)
 2480 VDU28,0,24,39,3
 2490 PRINT''"This is a horizontaly adapt
ed version"''"of the patience game 'Tower
 of Pisa'."
 2500 PRINT''"The game makes use of nine
cards only"''"numbered 1 to 9 and uses 4
suits."''"The object of the game is form
the "'"cards into a single line in corre
ct"''"descending order. From left to righ
t."''''"Only the right hand card in any ro
w";
 2510 PRINT" can be moved and it can onl
y be";" placed next to a card of a highe
r value."
 2520 PRINT''"An empty row can be filled
by starting"''"with the end card from eit
her of the"''"other two rows."
 2530 PRINTTAB(6,20);CHR$131;"Space bar
for more info.":IF GET=32 CLS
 2540 PRINT''"Moves are entered as a sin
gle key"''"press.  A 'beep' indicates an
illegal"''"move, which will be rejected b
y the"''"micro,  so you cannot cheat...."
 2550 PRINT''"Suit colours are irrelevant
 during play";" but will be correctly al
ternated in the"''"completed row..."
 2560 PRINT''"An incorrect entry can usua
lly be"''"rectified by entering the same
number"''"for the destination row as for
the"''"source row."
 2570 PRINTTAB(0,21);CHR$(131);"Press an
y key to start....";
 2580 IF GET:ENDPROC
 2590 :
 2600 DEF PROCdeal2(flag):VDU5:GCOL0,0
 2610 FOR r%=1TO3:FOR c%=1TO3
 2620 SOUND0,-15,6,1
 2630 xcord%=136*(c%-1)+32
 2640 ycord%=900-(r%-1)*260
 2650 GCOL0,3:PROCblank(1)
 2660 PROCsuits(r%,c%)
 2670 :
 2680 DEF PROCcard(r%,c%,flag)
 2690 xcord%=xcord%+4:ycord%=ycord%-8
 2700 MOVExcord%,ycord%:PRINTLEFT$(row$(
r%,c%),1)
 2710 MOVExcord%,ycord%-32:PRINTsuit$
 2720 MOVExcord%+88,ycord%-144:PRINTLEFT
$(row$(r%,c%),1)
 2730 MOVExcord%+88,ycord%-176:PRINTsuit
$
 2740 MOVExcord%+28,ycord%-76:PRINTsuit2
$
 2750 IF flag=1 THEN ENDPROC
 2760 NEXT:NEXT:ENDPROC
 2770 :
 2780 DEF PROCsuits(r%,c%)
 2790 IF RIGHT$(row$(r%,c%),1)="D" THEN
suit$=D$:suit2$=D2$:GCOL0,1
 2800 IF RIGHT$(row$(r%,c%),1)="H" THEN
suit$=H$:suit2$=H2$:GCOL0,1
 2810 IF RIGHT$(row$(r%,c%),1)="C" THEN
suit$=C$:suit2$=C2$:GCOL0,0
 2820 IF RIGHT$(row$(r%,c%),1)="S" THEN
suit$=S$:suit2$=S2$:GCOL0,0
 2830 ENDPROC
 2840 :
 2850 DEF PROCshuff
 2860 CLS:M$=CHR$141+CHR$134+"Shuffling.
."
 2870 FOR M%=1 TO LEN(M$)
 2880 FOR Y=12 TO 13
 2890 PRINTTAB(10+M%,Y);MID$(M$,M%,1)
 2900 SOUND0,-8,4,1
 2910 FOR I=1TO100
 2920 NEXT:NEXT:NEXT
 2930 FOR I=1TO500:NEXT:ENDPROC
```

View Printer Driver

David Holton presents the ultimate View printer driver.

Many of the old-timers like Mr Williams and myself actually like *View* as a word-processor. Let the PC users fall about laughing and crying WYSIWYG - their software will be obsolete, too, as the 486 gives way to the 586 and for all I know it'll be the 986 next Tuesday week. View and the Master do everything I want and I understand them; I'm not about to change until the Master finally dies due to the unavailability of parts. There is one big drawback to View, though, and that's the lack of a decent printer-driver. Even the 'View Printer Driver Generator' can't produce anything remotely adequate for the huge range of facilities now found on even the cheapest printers. This is due to the lack of space; View allocates just one page - 256 bytes - for a driver and the official Printer Driver Generator uses even that tiny area rather wastefully.

There are ways around it, and many good drivers have been printed in BEEBUG, though not for quite a while, which is why I'm offering this one. I wrote it just recently on buying a Panasonic KX-P1123; It emulates the Epson LQ-850 and so should work reasonably well with most printers attached to Beebs - you might have to alter one or two of the control codes here and there. I've rejected any fancy tricks like jumps to extra code outside - too much fiddling with extra files to be loaded - so it's pretty compressed and makes no claim to be *structured* (whatever that means) or elegant programming. It works.

It's obviously impractical to make combinations of just two highlights cover all the dozens of effects, and if you use any other symbols the formatting is messed up. This driver gets round that by incrementing a counter each time a code is inserted, and then adding to each subsequent space in that line of text another space to pad out the gap left by the symbol which didn't get printed. This is not a perfect solution, but there are only 256 bytes to play with.

Type in and run the program - be sure to save it under a safe filename first - answer the prompt for a filename and it will save the assembled driver. It is, of course, the driver which you load from within View by PRINTER <filename>. The file must contain no re-definition of the HIGHLIGHT TWO symbol, such as HT 2 130, or it won't work. It copes with 37 control codes. To get a given effect, put in the HIGHLIGHT TWO symbol followed by an upper-case letter A to Z, a numeral 0 to 9 or the character; the remarks in the listing itself tell you which symbol does what. Thus HT 2 followed by an 'A' selects Courier; followed by a W it turns on underlining and so on. The backspace is included purely so I can get a circumflex accent in French - you may have other uses for it. If you change or omit a code, note that '*' on its own will always select the first code in the table, *A will always select the second and so on, whatever those codes may be. Note the following if you need to make changes:

There are three types of control-code:

a) Single-bytes, like an 8 for a backspace or &12 for 'cancel compressed printing'.

b) ESC plus one byte, such as ESC+"P" for Pica or ESC+"E" for emphasized print.

c) ESC plus two bytes, like ESC+"x"+1 for LQ or ESC+"w"+1 for double height.

Believe it or not, the most economical method is to make the driver send out ESC followed by two bytes every time - the slight waste of bytes in the lookup table is more than made up for by the simplification of the code. You should therefore observe the following conventions:

a) Single bytes go into the table as EQUB 0 plus the desired byte. The driver then sends to the printer ESC+0, which does nothing, followed by the relevant byte.

b) ESC plus one byte: put into the table first the desired byte, then EQUB n. I've defined n as &FF at the start of the program, and when the driver reads that byte, which is negative in two's complement, it will divert it from the printer. The only reason for putting 'n' instead of &FF in the table itself is to keep the layout looking pretty.

c) ESC plus two bytes: put both bytes in the table, of course.

Note that EQUB 0 and EQUS "0" are two different animals; the first assembles a zero in the table, the second gives ASC "0", which is &30. EQUS "W" won't do for EQUS "w", either.

If anything doesn't work, peruse your printer's manual with care and amend my table as above. The driver should run on a BBC B and assumes that your printer is set to 'UK' as the default character set. My Master version is set up for USA default, and the savings in space mean that there's room to make the driver beep and print 'View is now free.' when the last byte has been stored by the printer's RAM and the driver shuts off. Maybe there'll be room in MC Corner to discuss variations sometime.

```
 10 REM Program View Printer Driver
 20 REM Version B 1.0
 30 REM Author  David Holton
 40 REM BEEBUG  Aug/Sep 1993
 50 REM Program subject to copyright
 60 :
100 osasci=&FFE3:n=&FF
120 FOR N%=4 TO 6 STEP 2
130 P%=&0400:O%=&7000
150 [ OPT N%
160 JMP print:JMP start
180 JMP stop:RTS
200 .flag
210 BRK
220 .spaces
230 BRK:RTS
250 .Xreg
260 BRK
270 :
280 .stop
290 LDA #3:JMP osasci
310 :
320 .start
330 LDA #2:JSR osasci
350 LDA #&1B:JSR osprint
370 LDA #ASC "@":JMP osprint
390 :
400 .print
410 PHP
420 STX Xreg:CMP #&81
440 BEQ dontPrintIt
450 CMP #&60 \ Pound
460 BNE notPound
470 LDA #&23:JSR osprint
490 JMP dontPrintIt
500 :
510 .notPound
520 CMP #&23:BNE notHash
540 LDX #6
550 :
560 .loop
570 LDA hashData,X:JSR osprint
590 DEX:BPL loop
610 :
620 .dontPrintIt
```

```
 630 STA flag:LDX Xreg
 650 PLP:RTS
 670 :
 680 .notHash
 690 CMP #&0D:BNE notCR
 710 .loop
 720 DEC spaces:BNE loop
 740 :
 750 .notCR
 760 LDX flag:CPX #&81
 780 BEQ gotThingy
 790 CMP #&20:BNE printIt
 810 LDX spaces:BEQ printIt
 830 JSR osprint:DEC spaces
 850 :
 860 .printIt
 870 STA flag:LDX Xreg
 890 PLP:JMP osprint
 910 :
 920 .gotThingy
 930 CMP #ASC "@":BCS sendCode
 950 ADC #&2B \ =1+ASC"Z"-ASC"0"
 960 :
 970 .sendCode
 980 AND #&3F
 990 ASL A:TAX
1010 INC spaces
1020 LDA #&1B
1030 JSR osprint
1040 LDA codes,X
1050 JSR osprint
1060 INX:LDA codes,X
1080 BMI dontPrintIt
1090 JMP printIt
1100 :
1110 .osprint
1120 PHA:LDA #1
1140 JSR osasci
1150 PLA:JMP osasci
1170 :
1180 .codes
1190 EQUB 0  :EQUB 8 \ @ = BACKSPACE
1200 EQUS "k":EQUB 0 \ A = COURIER
1210 EQUS "k":EQUB 3 \ B = PRESTIGE
1220 EQUS "k":EQUB 4 \ C = SCRIPT
1230 EQUS "k":EQUB 6 \ D = BOLD PS
1240 EQUS "P":EQUB n \ E = PICA
1250 EQUS "M":EQUB n \ F = ELITE
1260 EQUS "g":EQUB n \ G = MICRON
1270 EQUB &0F:EQUB n \ H = COMPRESSED
1280 EQUB 0:EQUB &12 \ I = KILL COMP
1290 EQUS "S":EQUB 0 \ J = SUPERSCRIPT
1300 EQUS "S":EQUB 1 \ K = SUBSCRIPT
1310 EQUS "T":EQUB n \ L = KILL SUP/SUB
1320 EQUS "p":EQUB 1 \ M = PROP SPCNG
1330 EQUS "p":EQUB 0 \ N = KILL PS
1340 EQUS "E":EQUB n \ O = EMPHASIZED
1350 EQUS "F":EQUB n \ P = KILL EMPH
1360 EQUS "w":EQUB 1 \ Q = DUBL HEIGHT
1370 EQUS "w":EQUB 0 \ R = KILL DUBL HT
1380 EQUS "W":EQUB 1 \ S = DUBL WDTH
1390 EQUS "W":EQUB 0 \ T = KILL DUBL WD
1400 EQUS "G":EQUB n \ U = DUBL STRIKE
1410 EQUS "H":EQUB n \ V = KILL DUBL ST
1420 EQUS "-":EQUB 1 \ W = UNDERLINE
1430 EQUS "-":EQUB 0 \ X = KILL U/L
1440 EQUS "4":EQUB n \ Y = ITALIC
1450 EQUS "5":EQUB n \ Z = KILL ITALIC
1460 EQUS "x":EQUB 0 \ 0 = DRAUGHT
1470 EQUS "x":EQUB 1 \ 1 = LQ
1480 EQUS "R":EQUB 1 \ 2 = FRENCH
1490 EQUS "R":EQUB 2 \ 3 = GERMAN
1500 EQUS "R":EQUB 3 \ 4 = ENGLAND
1510 EQUS "R":EQUB 4 \ 5 = DENMARK 1
1520 EQUS "R":EQUB 5 \ 6 = SWEDEN
1530 EQUS "R":EQUB 6 \ 7 = ITALY
1540 EQUS "R":EQUB&0B\ 8 = SPAIN(2)
1550 EQUS "R":EQUB&40\ 9 = LEGAL
1560 :
1570 .hashData
1580 EQUB 3:EQUS "R":EQUB &1B
1590 EQUB &23
1600 EQUB 0:EQUS "R":EQUB &1B
1610 :
1620 ] NEXT
1630 FOR N%=0% TO &70FF
1640 ?N%=0:NEXT
1650 :
1660 bytes%=P%-&400
1670 PRINT'"Bytes used: &";~bytes%
1680 PRINT"Bytes free: &";~&100-bytes%
1690 PRINT'"Save this driver? (Y/N)"'
1700 q$=CHR$(ASC GET$ AND &DF)
1710 IF q$<>"Y" THEN END
1720 INPUT"Filename? "f$
1730 IF f$="" THEN f$="DRVR"
1740 PRINT'"Saving as '"f$"'"
1750 OSCLI"SAVE "+f$+" 7000 70FF 400 40
0"
1760 END
```

Full-Screen Basic Program Editor

by John Cole

The Basic full-screen editor presented here is extremely fast and powerful considering its length.

Although BBC computers such as the Master and Compact contain a powerful text and Basic editor in the form of *EDIT*, owners of earlier machines either have to make do with Basic's built-in editor or purchase a dedicated editor that is usually ROM based.

The surprisingly short program listed here provides a true screen editor, and being written in Basic itself can be truly called a Basic Editor. Although the program was originally written for machines such as the Model B, it will work just as well on a Master or Compact, and can provide a very flexible alternative to *EDIT*.

But why use an editor written in Basic, not assembler? The high level language makes the listing much more compact for you to key in from this article, and far easier for you to customise according to your personal needs and preferences.

The reason for NOT doing so, you might think, is that Basic could not possibly be fast enough. Try it out and you will be surprised!

The editor consists of a procedure and five functions, designed to be merged onto the end of the program under development. Merging by means of *SPOOL and *EXEC is described in the User Guide. Once merged, they can be loaded and saved as one, until the program is finished when the editor lines can be deleted. The editor is invoked the first time by typing PROCED in command mode. This also defines function key 10. Thereafter you can just press Break to enter the editor in the same mode as that last used for editing. The editor also resets to their default values the *FX states set up at the time.

Upon entering the editor you will find yourself in 'procedure mode'. This means that only the first line of the program and lines beginning with the keyword DEF (one leading space is allowed) are listed, to give an overall view of properly structured programs (note that this also lists functions). Move through the definition list using the up and down cursor keys, and if it is long enough you will find that it scrolls properly in both directions. To actually list the program you need to enter 'line mode' by pressing Return, but first move the cursor either to the top of the program, or to the particular definition you want to home in on directly.

In 'line mode', the down cursor key produces a line-by-line listing, once again with true scrolling: whatever disappears off the top of the screen can be recalled with the up cursor key. The left and right cursor keys move the cursor within individual lines, so you can insert or delete characters at will. The Copy key gives a 'clear to end' function. Whatever changes you make are actually entered into the program only when you press Return, so if you accidentally mess up a line just press Escape or the up cursor key to recover the original version. When you do press Return, the screen is cleared and the new version of the line is listed, with any abbreviations (like P. for PRINT) properly expanded.

The Return key is also used to insert a new line between the current line and the next. The default line number is chosen, provided there is room, to be the normal 10 greater than the current line number. If this is not possible, a difference of first 2 then 1 is used. A warning beep means the line numbers are consecutive, leaving no room at all.

The fact that the line number itself can be edited as part of the line enables you to copy an identical, or modified, version of a line to a completely different part of the program. That then becomes the current line.

All this, with very little practice, becomes very natural to use. There are also just two other commands to learn. In line mode, Ctrl-D is a quick way to delete the current line. Ctrl-S is a string search which asks for a target string, then lists downwards from the current line until it is found. The string can include keywords and/or line numbers. It can be used in procedure mode as well to search out a definition by name.

Space doesn't allow a full explanation of how the editor works, but the crucial technique is a means of tying up communication between a Basic program and command mode, required for listing and entering new program lines. It is done without the involvement of the input buffer (whose 32 character capacity is not sufficient for program lines of a reasonable length) and without calling routines in the Basic ROM directly. The RDCHV and WRCV vectors are redirected to code located in Zero Page at &70, which in turn uses Page &A (normally the RS423 input buffer) as a buffer for input and output. The effect of using the same buffer for both purposes is that PRINT statements can generate commands which are obeyed as soon as command mode is entered. Provided the command string ends with a Basic function reference, that function will be re-entered as soon as the commands have been carried out. The code is so short it is poked directly with the ! operator, but an assembler listing would look like:

```
.WRCHentry STA &A00
      INC WRCHentry+1
      RTS

.RDCHentry LDA &A00
      INC RDCHentry+1
      CLC
      RTS
```

Surprisingly, such recklessness does not end in disaster. The command mode input causes echoes, intended for the screen, which overwrite the common I/O buffer, but the RDCH pointer keeps safely behind the WRCH pointer until the vectors are restored.

Note: This program was originally published in BEEBUG Vol.4 No.7.

```
  10 DEF PROCED
  20 MD=?&355:IFMD=0 ORMD=3 D%=80 ELSEI
FMD=2 ORMD=5 D%=20 ELSED%=40
  30 ?&8F=MD:*KEY10O.|MMD=?&8F:MODEMD|M
PROCED|M
  40 !&78=&1877E60A:?&7C=&60
  50 IFPAGE<6400M%=TOP+256ELSEM%=&1100
  60 T%=PAGE+1:L%=M%:A%=&A10:I%=!&20E
  70 K%=0:W%=0:S%=0:F%=TRUE
  80 VDU12;:PRINT'" Procedure mode"
  90 *FX4,1
 100 *FX229,1
 110 IFFNY
 120 ENDPROC
 130 DEF FNW
 140 IFL%=M%=0
 150 !L%=0:L%=L%-2:T%=!L%:=TRUE
 160 DEF FNX
```

Continued on page 27

Public Domain

Software from Poland, Zombies and a look at the lighter side of running a PD library are amongst the subjects covered this issue by Alan Blundell.

Those readers who were in contact with me almost two years ago will perhaps remember that I put quite a lot of effort into writing an on-disc news 'column' about developments in PD and shareware for the BBC Micro. That far into the past, there were about the same number of PD libraries as there are now, but the number was increasing rather than decreasing (as it sadly has been for some time). My own PD library had been going for about 18 months and, whilst it boasted a respectable range of software for the time, my full catalogue was easily 100 (double-sided) discs lighter than it now is. Newly available PD software was hot news to devotees, and I thought that I had some pretty hot news when I publicised the fact that I had received some excellent new software from a contributor in Poland. It would, I wrote, be available 'soon'.

'Soon' has turned out to be two years, as far as my library was concerned - a bit slower than usual! The author of the software concerned was Miroslaw Bobrowski, whose Robol game has recently been published in BEEBUG. I had had a little correspondence with him when I managed to lose a disc containing the said software, complete with his letter (the only disc/letter that I have so far succeeded in losing without trace, and it had to be from someone whose address I hadn't recorded). It was not until much later that I managed to get his address from another correspondent, and I wrote to him just before the first installment of Robol appeared in BEEBUG. Fortunately, he has forgiven me and has again sent me a selection of his software, collectively entitled Software from Poland.

This is more than just a title for the disc, however, as the content includes some definitely Polish items. The overall content of the disc is of various types, although all of good quality, ranging from a rendition of Scott Joplin's 'Maple Leaf Rag' to a program to handle non-linear equations. The most interesting part of the collection to me was a history of Poland. A two-part presentation is included in the collection which describes in text and graphics the geography and history of a country which went from its 10th century origins, through various partitions and annexations by other empires, to today. The whole subject is presented in an interesting and easily digestible style and is well worth a look. The rest of the collection includes a selection of Polish Christmas carols (did you realise how few shopping days there are left?....), three selections from the works of Mozart, a battleships game, a periodic table program and three utilities.

ZOMBIES

I can also report on another disc of software which has been sent to me by Paul Fletcher. This is another mixed collection (mixed in the sense of program variety rather than quality). It includes an ADFS directory deletion utility, which saves typing multiple commands to achieve the same effect, a car mileage per gallon evaluator, a Dungeons and Dragons character sheet printer, an Advanced Heroquest character generator, a Help ROM file which is intended for use with Steven Flintham's excellent SWR Help system ,and a number of text files, for which viewer programs are included.

It was the text files on this disc which particularly caught my interest. They include complete solutions to two adventure games; Melbourne House's The Hobbit and the public domain Bungle Brothers adventure. Also included are several humorous files and, the one I HAD to read, The Truth About Zombies. This is a factual text which concentrates on discovering whether the horror film zombie is based on anything in real life (the answer to the question is 'Yes', but I won't go into the details of the real-life Haitian basis of the zombie here).

HUMOUR

You have to have a sense of humour to run a PD library (apart from the obviously necessary masochistic tendency, that is!). Like anyone else, I occasionally do silly things by mistake (like losing Miroslaw's letter & disc). I always feel better though when I get an example of how it can work the other way as well.

The one I always remember is the writer to BBC PD in its early days, who wrote to me in December 1990 to order a copy of SPELLCHECK (it would obviously have been useful to him!). This is my longest standing unfulfilled order, because whoever it was forgot to identify him or herself at all. The only clue I have is that the letter was postmarked 'Sutherland'; if this rings a chord with you, write and let me know your address and I will gladly forward the disc! The method of payment was also rather unique, so describe that in the letter and I will know it was you.

People who send software in for addition to libraries also occasionally get things wrong, like the submission of a program to me which I knew because I had written it myself. In itself, this isn't particularly unusual, having happened a number of times. It also isn't that unusual for me to receive a copy of a disc with a BBC PD menu (i.e. originating from BBC PD) with a suggestion that I add it to the library. One example in particular stands out, though: I was, a while ago, sent a copy of a program which I had written, with a recommendation that I add it to my library and details of what the contributor would like in exchange. Not particularly unusual in itself, but this program was one which I had distributed to a limited number of people using another name (a long time ago, when I partly thought that 'Alan Blundell' didn't sound too good as a software author and partly just wanted to try something out) and had never released as public domain. More, the contributor had acquired it from another PD library!

SUBMITTING SOFTWARE

People who send me their own programs are frequently very dismissive of their own efforts ('I will understand if you do not add this to your library, but here it is anyway.. '). More often than not, when such programs are circulated, I get back very favourable reactions. It is rarer, but not unknown, to receive a letter saying 'here is my program - you will find that it is excellent'. The most recent of these which I actually did catalogue was hastily withdrawn from circulation when everyone who tried it found problems. The most excessive example thought that his contribution warranted my sending a dozen discs of software in return for a quite small submission which I never used.

I suppose that the moral I would like to draw from this is that, if you have written something which may be of interest to others, but you don't think it is of a high enough standard, think again. On the other hand, if you think your work is truly wonderful and can't understand how any BBC owner could do without it, ask a close friend for a second opinion before you get carried away.

Star LC24-200 Screen Dump

Doug Blyth gets some high-quality output.

I recently replaced my MX80 with a 24 pin Star. Unfortunately, the output from the *Dumpmaster* software spread onto a second page with the Star so I have written my own program for modes 1, 2, 4 and 5.

The program uses the 24-bit triple-density graphics mode of the LC24, giving 1440 pin positions in an eight inch printed line; using five dots/pixel gives a Y axis of 7.11 inches. For the right aspect ratio the X axis needs to be 8.88 inches, equivalent to 1600 dots or 5 dots/pixel. Each pass of the print head uses 20 pins, giving a four pixel depth for each line.

A selection is made of the 25 dots for each pixel to give a range of shades. These have been chosen to represent the relative brightness of colours on screen.

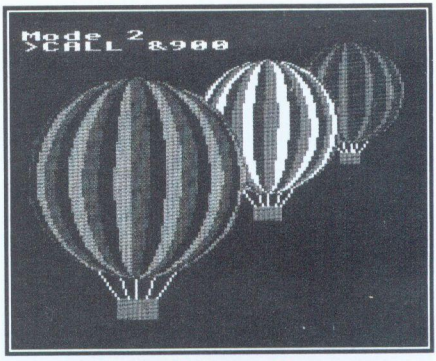
Colour	black (0)	blue (4)	red (1)	magenta (5)
No. of pins	0	4	7	10
Pin pattern	00000 00000 00000 00000 00000	00000 00000 01010 00000 01010	00000 00000 01010 10101 01010	01010 01010 10101 00000 10101

	green (2)	yellow (3)	cyan (6)	white (7)
	14	17	21	25
	01010 01010 10101 01010 11111	01010 01010 11111 10101 11111	11111 11111 10101 11111 10101	11111 11111 11111 11111 11111

The patterns are defined in a byte array *pattern* starting at &70. So, colour 5 is defined by &58 held in address &75. The MS nibble, &5, is used for columns 1, 3 and 5 after expanding to five bits by copying the MS bit; the LS nibble, &8, is expanded and used for columns 2 and 4.

The program is assembled at &900 and uses two pages. It is run with CALL &900; an inverse image is printed by using a parameter, e.g, CALL &900,A%, the value of the parameter is not important.

I am sure the program could be improved, particularly in the pattern arrangement for each colour. The conversion of the pattern into bytes is cumbersome and could be improved to reduce the striped effect, but might result in the code exceeding two pages in length. I hope readers find this of use. I have also written a mode 0 dump, this is provided on this month's disc as *DUMP0*.



Example screen dump

```
  10 REM Program SCDUMP24-8 Mk3
  20 REM Version B1.0
  30 REM Author  D.T.Blyth
  40 REM BEEBUG  Aug/Sep 1993
  50 REM Program subject to copyright
  60 :
 100 byte=&78: temp=&7E: Xlo=&80: Xhi=&
81: Ylo=&82: Yhi=&83
 110 logicol=&84: value=&85: count4=&89
```

```
: mask=&8A
 120 code=&900: oswrch=&FFEE: osword=&F
FF1
 130 :
 1000 FOR opt=0 TO 3 STEP 3
 1010 P%=code: REM use CFS buffer area f
or code
 1020 [OPT opt
 1030 lda &600: sta mask: beq start: lda
 #&FF: sta mask \ parameter
 1040 .start
 1050 lda #2: jsr oswrch \enable printer
 1060 \ set line spacing to 20/180 inch
 1070 lda #27: jsr printer: lda #51: jsr
 printer: lda #20: jsr printer
 1080 \ initialize X co-ord
 1090 lda #0: sta Xlo: sta Xhi
 1100 \ send triple density 24 bit image
 code (&500 = 256x5 bits)
 1110 :
 1120 .ycoord
 1130 lda #27: jsr printer: lda #42: jsr
 printer: lda #39: jsr printer
 1140 lda #00: jsr printer: lda #&5: jsr
 printer
 1150 \ reset Y co-ord
 1160 lda #0: sta Ylo: sta Yhi: jmp setc
ount
 1170 \ restore X base
 1180 :
 1190 .pixel
 1200 lda Xlo: sec: sbc #16: sta Xlo: bc
s setcount: dec Xhi
 1210 \ calculate & print 16*4 points
 1220 :
 1230 .setcount
 1240 lda #4: sta count4
 1250 \ calculate POINT(X,Y)
 1260 :
 1270 .byteloop
 1280 ldx #Xlo MOD 256: ldy #Xlo DIV 256
 1290 lda #9: jsr osword \ read logical
colour
 1300 ldx #logicol MOD 256: ldy #logicol
 DIV 256
 1310 lda #&B: jsr osword \ read palette
 1320 lda value: and #7: tay: lda patter
n,Y: eor mask
 1330 \transfer colour pattern to 2 byte
s
 1340 rol A: rol byte+1:rol A:rol byte+1
 1350 rol A: rol byte+1:rol A:rol byte+1
 1360 rol A: rol byte+4:rol A:rol byte+4
 1370 rol A: rol byte+4:rol A:rol byte+4
 1380 \ increment X by 4
 1390 clc: lda Xlo: adc #4: sta Xlo
 1400 lda Xhi: adc #0: sta Xhi
 1410 dec count4: lda count4: cmp #2: be
q second
 1420 cmp #0: bne byteloop: jmp printbyt
es
 1430 :
 1440 .second
 1450 lda byte+1: sta byte: lda byte+4:
sta byte+3: jmp byteloop
 1460 \ print prepared bytes
 1470 :
 1480 .printbytes
 1490 \ expand two bytes to 20 bits
 1500 lda #0: sta byte+2: sta byte+5: ld
x #3
 1510 :
 1520 .exp
 1530 lda byte+1,X: lsr A: ror byte+2,X:
 lsr A: ror byte+2,X: lsr A: ror byte+2,
X: lsr A: ror byte+2,X
 1540 lda byte+1,X: asl A: ror byte+1,X
 1550 lda byte,X: lsr A: ror byte+1,X: l
sr A: ror byte+1,X
 1560 lda byte,X: lsr A: ror temp: lsr A
: ror temp: lsr A: ror temp: lsr A: ror
temp
 1570 lda byte,X: asl A: ror byte,X
 1580 lsr byte,X: lsr byte,X
 1590 asl temp: rol byte,X: asl temp: ro
l byte,X
 1600 dex: dex: dex: beq exp
 1610 :
 1620 .loop
 1630 lda byte: jsr printer: lda byte+1:
 jsr printer
 1640 lda byte+2: jsr printer \ pins 17
to 20
 1650 lda byte+3: jsr printer: lda byte+
4: jsr printer
 1660 lda byte+5: jsr printer \ pins 17
to 20
 1670 lda byte: jsr printer: lda byte+1:
 jsr printer
 1680 lda byte+2: jsr printer \ pins 17
to 20
```

Continued on page 46

Hearing Test

Ian Crawford presents a sound application.

All things deteriorate in efficiency over a period of time; human hearing is just one example. We are often not aware by how much it deteriorates until it is too late - then we need a hearing aid. If we could regularly test our level of hearing then we could do something about it *before* it was too late.

Recently I had to undergo audio tests both at my local NHS Hospital and also at my home with a private audiologist. It struck me that a humble BBC'B' could help test, to an individual's satisfaction, if and where hearing loss was taking place.

The 'proper' tests consist of the patient fitting earphones and pressing a button to indicate to the operator the *precise* moment they hear *any* sound. The tests are conducted twice, once for each ear, and two overlapping graphs are hand drawn by the operator indicating the level at which you indicated you heard the sound and the pitch (frequency) of that sound - all very archaic.

Personally, I've got hearing (just) at normal speech frequencies but I have what is called 'high frequency cut off' which means that as sounds pitch higher (shrill noises, bird songs, etc), I have no hearing at all. This is interesting to know because I would have sworn that the treble control on my expensive radio was broken, because I can't hear any difference no matter how I adjust it. Now I understand. It is also the reason why, if I'm in a noisy environment (pub, noisy room, or with the radio on at home with people talking) I often can't hear what is said to me because the voice of the person speaking to me is overridden by the background noise. I often don't hear the telephone ring for this reason.

The program *Hearing* emulates the normal pitch, note and octave levels that are covered by the NHS tests but without individual testing to each ear via headphones. However, by being in an *absolutely* quiet room and running the program it is possible to obtain a good indication of one's overall hearing.

USING THE PROGRAM

Type in the listing below and save it. Run the program and as soon as your name is entered the test will begin. The program runs through 60 ascending frequencies in five octaves starting with octave 2 of the BBC's range. At each frequency the sound is played 15 times at increasing volume (noise level); press Return as soon as you hear something. This stops the test at the current frequency, displays the results and starts testing the next frequency. Pressing any other key will advance to the next volume level.

```
Audio Hearing Test for: Marshal
          START Key  ON
           STOP Key  OFF
         Noise Level=2
        Pitch Number=49
        Current Note=B
      Octave Number=2
           You Heard:A#
           At  Pitch:45
           At  Level:2
Press RETURN when You Hear ANY Sound
```

Using the hearing test

THE DISPLAY

The start and stop key parts of the display were part of the development of the program and are left in as their flashing indicates the program is running. The Noise Level is the volume starting at 0 and rising to 15, the Pitch

and Octave numbers relate to BBC Basic's SOUND command, and the Current Note is it's musical equivalent (see page 181 BBC User Guide). The second part of the display shows the result of the previous test.

Draw a graph before you start with the Y axis scaled from noise level 15 at the base up to 0 at the top and the X axis with 60 divisions (octave 2 to octave 6, 12 notes in each, representing 250KHz to 8000KHz). Mark off the noise level for each pitch and the resultant graph should *commence* at around noise level 2 and, if your hearing is OK at all frequencies, a virtual straight line results along the top of the graph.

To allow you to mark onto graph paper your response level, look only at the 'At Level' figure at the base of the screen, and, if necessary, alter line 300:time=100 to 200 or 400 to give you more time to react.

My problem is the graph rapidly falls away down to level 7 or 9 from Octave 6, Note E onwards. I hope readers find this project of interest.

Editor's note: We hope you find this article as interesting as we did; it shows yet another use for your machine. However, while this program is an interesting exercise we should point out that it in no way replaces a professional test by a qualified person with the correct equipment.

```
  10 REM Program Hearing
  20 REM Version B 1.0
  30 REM Author  Ian Crawford
  40 REM BEEBUG  Aug/Sep 1993
  50 REM Program subject to copyright
  60 :
 100 REPEAT
 110 ON ERROR GOTO190
 120 MODE7:PROCsetup
 130 PROCinitialise
 140 PROCpitch
 150 UNTIL TRUE
 160 PROCend
 170 END
 180 :
 190 REPORT:PRINT " at line ";ERL:END
 200 :
1000 DEF PROCsetup
1010 CLS:REM..Developed on BBC'B' 32K
1020 VDU14:REM..Paged Mode ON
1030 VDU23;8202;0;0;0;:REM..Cursor OFF
1040 ENDPROC
1050 :
1060 DEF PROCinitialise
1070 INPUT LINE "Please entre your NAME
: "yourname$
1080 time=100:REM..On/Off Cycle Time
1090 maxnoise=15
1100 note$=""
1110 yournote$=""
1120 yournoise%=0
1130 oldpitch%=0
1140 step=4:REM..Pitch Step Min-to-Max
1150 ENDPROC
1160 :
1170 DEF PROCpitch
1180 CLS:VDU7:PRINTTAB(0,2)"Audio Heari
ng Test for: ";yourname$
1190 PRINTTAB(9,4)"START Key ";CHR$(129
)"OFF"
1200 PRINTTAB(10,6)"STOP Key ";CHR$(129
)"OFF"
1210 I=INKEY(time/2)
1220 REPEAT
1230 READ octave,lowpitch,hipitch
1240 FOR pitch%=lowpitch TO hipitch STE
Pstep
1250 IF pitch%=5 OR pitch%=53 OR pitch%
=101 OR pitch%=149 OR pitch%=197 THEN no
te$="C"
1260 IF pitch%=9 OR pitch%=57 OR pitch%
=105 OR pitch%=153 OR pitch%=201 THEN no
te$="C#"
1270 IF pitch%=13 OR pitch%=61 OR pitch
%=109 OR pitch%=157 OR pitch%=205 THEN n
ote$="D"
1280 IF pitch%=17 OR pitch%=65 OR pitch
%=113 OR pitch%=161 OR pitch%=209 THEN n
ote$="D#"
1290 IF pitch%=21 OR pitch%=69 OR pitch
%=117 OR pitch%=165 OR pitch%=213 THEN n
ote$="E"
```

```
 1300 IF pitch%=25 OR pitch%=73 OR pitch
%=121 OR pitch%=169 OR pitch%=217 THEN n
ote$="F"
 1310 IF pitch%=29 OR pitch%=77 OR pitch
%=125 OR pitch%=173 OR pitch%=221 THEN n
ote$="F#"
 1320 IF pitch%=33 OR pitch%=81 OR pitch
%=129 OR pitch%=177 OR pitch%=225 THEN n
ote$="G"
 1330 IF pitch%=37 OR pitch%=85 OR pitch
%=133 OR pitch%=181 OR pitch%=229 THEN n
ote$="G#"
 1340 IF pitch%=41 OR pitch%=89 OR pitch
%=137 OR pitch%=185 OR pitch%=233 THEN n
ote$="A"
 1350 IF pitch%=45 OR pitch%=93 OR pitch
%=141 OR pitch%=189 OR pitch%=237 THEN n
ote$="A#"
 1360 IF pitch%=49 OR pitch%=97 OR pitch
%=145 OR pitch%=193 OR pitch%=241 THEN n
ote$="B"
 1370 PROCclearnote
 1380 PROChearnoise
 1390 SOUND &0011,0,0,1
 1400 NEXT pitch%
 1410 UNTIL hipitch=241
 1420 SOUND &0011,0,0,1
 1430 ENDPROC
 1440 :
 1450 DEF PROCend
 1460 VDU7:PRINTTAB(8,23)"Test Terminate
d"
 1470 ENDPROC
 1480 :
 1490 REM..1st No=Octave Number
 1500 REM..2nd No=Start level for PITCH
 1510 REM..3rd No=End level for PITCH
 1520 REM..See Page 181 BBC User Guide
 1530 DATA 2,5,49
 1540 DATA 3,53,97
 1550 DATA 4,101,145
 1560 DATA 5,149,193
 1570 DATA 6,197,241
 1580 END
 1590 :
 1600 DEF PROCclearnote
 1610 REM.Clears earlier Note off Screen
 1620 PRINTTAB(21,8)"    "
 1630 PRINTTAB(21,12)"    ";
 1640 PRINTTAB(21,14)"    ";
 1650 PRINTTAB(21,20)"    "
 1660 ENDPROC
 1670 :
 1680 DEF PROChearnoise
 1690 FOR noise%=0 TO maxnoise
 1700 SOUND 1,-noise%,pitch%,-1:REM..The
 1710 REM..-1 gives a CONTINUOUS sound
 1720 PRINTTAB(9,8)"Noise Level=";noise%
 1730 PRINTTAB(8,10)"Pitch Number=";pitc
h%
 1740 PRINTTAB(8,12)"Current Note=";note
$
 1750 PRINTTAB(7,14)"Octave Number=";oct
ave
 1760 PRINTTAB(11,16)"You Heard:";
 1770 IF yournoise%<2 THEN PRINTTAB(21,1
6)"   " ELSE PRINTTAB(21,16)yournote$
 1780 IF yournoise%<2 THEN PRINTTAB(11,1
8)"At  Pitch:";"   " ELSE PRINTTAB(21,18
);oldpitch%
 1790 PRINTTAB(11,20)"At  Level:";yourno
ise%
 1800 PRINTTAB(0,22)"Press RETURN when Y
ou Hear ANY Sound"
 1810 PRINTTAB(19,4)CHR$(130)"ON ":REM..
 Start Key 'ON'
 1820 I=INKEY(time):PRINTTAB(19,4)CHR$(1
29)"OFF":REM..Start Key 'OFF'
 1830 PRINTTAB(20,6)CHR$(130)"ON ":IF I=
13 THEN SOUND &0011,0,0,1:SOUND 1,-noise
%,pitch%,-1:yournoise%=noise%:noise%=max
noise:yournote$=note$:oldpitch%=pitch%:P
RINTTAB(21,16)"   "TAB(21,18)"   ":GOTO1
890
 1840 SOUND &0011,0,0,1:REM..This turns
 1850 REM..OFF the previous sound!
 1860 I=INKEY(time/2):PRINTTAB(19,6)CHR$
(130)"ON ":REM..Stop Key 'ON'
 1870 PRINTTAB(9,4)"START Key ";CHR$(129
)"OFF":IF I=13 THEN SOUND &0011,0,0,1:SO
UND 1,-noise%,pitch%,-1:yournoise%=noise
%:noise%=maxnoise:yournote$=note$:oldpit
ch%=pitch%:PRINTTAB(21,16)"   "TAB(21,18
)"   ":GOTO1890
 1880 SOUND 1,-noise%,pitch%,-1
 1890 PRINTTAB(10,6)"STOP Key ";CHR$(129
)"OFF"
 1900 NEXT noise%
 1910 ENDPROC
 1920 :
```

Loan Repayments

by Sheridan Williams

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

```
Loan repayments

Amount borrowed (or Return) £ 5000
Repayments (or Return) £
Annual interest rate (%) : 24
Number of years loan to run : 3
Number of repayments per year : 12
Variable or Fixed: V


Repayment = £ 196.16
Total repaid = £ 7061.91

Table of repayments (Y or N)?
```

Setting up the input

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

```
Loan repayments

------- End of year 2.00 -------

     25.00    1919.83     154.67
     26.00    1762.06     157.77
     27.00    1601.14     160.92
     28.00    1437.00     164.14
     29.00    1269.57     167.42
     30.00    1098.80     170.77
     31.00     924.61     174.19
     32.00     746.94     177.67
     33.00     565.71     181.23
     34.00     380.86     184.85
     35.00     192.32     188.55
     36.00      -0.00     192.32
------- End of year 3.00 -------


Annual interest rate (%) :

      Press Shift to continue
```

Loan repayments

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

Note: This program was originally published in BEEBUG Vol.4 No.5.

```
  10 REM Program REPAY
  20 REM Version 1.0B
  30 REM Author Sheridan Williams
  40 REM BEEBUG August/September 1993
  50 REM Program subject to copyright
  60 :
  70 MODE 7:@%=&2020A:VDU14
  80 FOR I=1TO2:PRINTTAB(8,I)CHR$131;CH
R$141"Loan repayments":NEXT
  90 VDU28,0,24,39,4,12
 100 A=FNinput("Amount borrowed",TRUE,T
RUE)
 110 R=FNinput("Repayments",A,TRUE)
 120 IF A=0 AND R=0 THEN 140
 130 REPEAT
 140 I=FNinput("Annual interest rate (%
)",FALSE,FALSE)
 150 UNTIL I>=0
 160 y=FNinput("Number of years loan to
 run",NOT(A=0 OR R=0),FALSE)
 170 IF (A=0 OR R=0) AND y=0 THEN 190
 180 REPEAT
 190 n=FNinput("Number of repayments pe
r year",FALSE,FALSE)
 200 UNTIL n>0
 210 PRINT"Variable or Fixed: ";
 220 REPEAT:type%=GET:UNTIL INSTR("VvFf
",CHR$(type%))
 230 PRINT CHR$(type%AND223):fix%=((typ
e%AND16)=0)
 240 nr=y*n:i=I/(100*n):i1=i+1
 250 IF i=0 i2=y*n ELSE i2=(i1^nr-1)/(i
*i1^nr)
 260 IF fix% i2=nr/(1+I*y/100)
```

```
  270 PRINT''
  280 IF A=0 A=FNamount:PRINT"Amount bor
rowed = ` ";A
  290 IF R=0 R=FNrepayment:PRINT"Repayme
nt = ` ";R''"Total repaid = ` ";nr*R
  300 IF y=0 y=FNnumberofyears:nr=y*n:PR
INT"Number of years = ";y
  310 PRINT''"Table of repayments (Y or N
)? ";:table=(GET AND 223)=89
  320 IF NOT table THEN 120
  330 VDU12,26:PRINTTAB(4,24)CHR$130"Pre
ss Shift to continue";:VDU28,0,22,39,4
  340 PRINT"Amount borrowed ";A:IF fix%
A=nr*R
  350 PRINT SPC4"Payment"SPC2"To pay"SPC
4"Repayment"
  360 FOR x=1 TO nr
  370 IF fix% N=A-R ELSE N=A*i1-R
  380 PRINT x,N,A-N
```

```
  390 IF x MOD n=0 PRINT"_______ End of
year ";x DIV n;" _______"'
  400 A=N:NEXT x:PRINT':REPEAT UNTIL INK
EY(-1):GOTO120
  410 END
  420 DEF FNamount=R*i2
  430 DEF FNrepayment=A/i2
  440 DEF FNnumberofyears
  450 IF R-A*i<0 THEN PRINT"NEVER REPAID
":=10000
  460 IF fix% THEN =A/(R-i*A)/n
  470 =LOG(R/(R-A*i))/(n*LOG(i1))
  480 DEF FNinput(text$,flag,`%)
  490 PRINT text$;:IF flag PRINT" (or Re
turn)";
  500 IF `% PRINT" `"; ELSE PRINT" :";
  510 INPUT " " reply
  520 =reply
  530
```

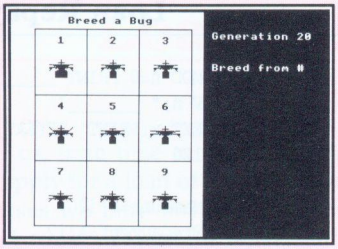
Full-Screen Basic Program Editor (continued from page 18)

```
  170 =256*?T%+T%?1
  180 DEF FNY
  190 !&70=&E60A008D:!&74=&AD6071:!&20E=
&760070:IFW%PRINTT$'"IFFNZ":=0ELSEPRINT"
L.";FNX'"IFFNZ":=0
  200 DEF FNZ
  210 !&20E=I%:IFW%W%=0:VDU12:=FNY
  220 R%=A%+LEN$A%+1:IFK%=139 PRINTSTRIN
G$(O%+2+LEN$R%DIV D%,CHR$11);
  230 O%=LEN$R%DIV D%:R%?(LEN$R%-1)=13:P
RINT'CHR$32;$R%;
  240 IFF%K%=GETELSEIFLEN$R%>LENS$:IFINS
TR($R%,S$)F%=TRUE:GOTO32230
  250 IFK%=139:IFFNW:=FNY
  260 IFK%=138X%=T%:REPEAT:X%=X%+X%?2:UN
TILS%OR?X%=255ORX%?3=&DDORX%?4=&DD:IF?X%
<255!L%=T%:T%=X%:L%=L%+2:=FNY
  270 IFK%=13 ANDNOTS%S%=TRUE:L%=M%:VDU1
2;:PRINT'CHR$32"Line mode":=FNY
  280 IFK%=19 AND?(T%+T%?2)<255F%=0:PRIN
T'CHR$32"Target:";:K%=137:S$=FNEL(""):K%
=138:VDU11:GOTO32250
  290 IFK%=27THEN32360
  300 IFK%=138ORK%=139ORNOTS%F%=TRUE:GOT
O32230
  310 W%=TRUE:IFK%=4:T$=STR$FNX:VDU7:=FN
W*FNY
```

```
  320 IFK%=13PRINT:H%=POS:V%=VPOS:VDU28;
24,39,V%,11,26,31,H%,V%,32:N=256*?(T%+T%
?2)+?(T%+T%?2+1)-FNX:T=FNX+1-(N>2)-8*(N>
10):PRINT;T;:K%=137:VDU6-(N=1):T$=FNEL(S
TR$T)ELSET$=FNEL($R%)
  330 T=VALT$:IFK%<>13ORT=0T$="":VDU7:=F
NY
  340 IFT<FNX:REPEAT:UNTILNOTFNW ORT>FNX
:IFL%=M%T%=PAGE+1
  350 IFT>FNX REPEAT:!L%=T%:L%=L%+2:T%=T
%+T%?2:UNTILT<=FNX
  360 =FNY
  370 *FX4
  380 *FX229
  390 =0
  400 DEF FNEL(A$)
  410 $A%=A$:C%=A%+LEN$A%:VDU152,8:REPEA
T:B%=K%:IFK%=127:IFC%>A%$(C%-1)=$C%:K%=1
36
  420 IFK%<135:IFK%>31$(C%+1)=$C%:?C%=K%
:K%=137
  430 IFK%=135:?C%=13:K%=0
  440 IFK%=136:IFC%>A%C%=C%-1:VDU8
  450 IFK%<>B%H%=POS:V%=VPOS:PRINT$C%;CH
R$152;:V%=V%+(VPOS=24):VDU10,31,H%,V%
  460 IFK%=137:IF?C%<>13:C%=C%+1:VDU9
  470 K%=GET:UNTILK%<32ORK%>137:=$A%
```

Best of BEEBUG

Applications I Disc

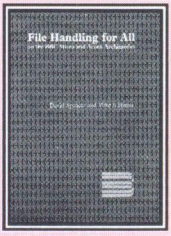


Business Graphics - for producing graphs, charts and diagrams
Video Cataloguer - catalogue and print labels for your video cassettes
Phone Book - an on-screen telephone book which can be easily edited and updated
Personalised Letter-Headings - design a stylish logo for your letter heads
Appointments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

File Handling for All

on the BBC Micro and Acorn Archimedes

by David Spencer and Mike Williams



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

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

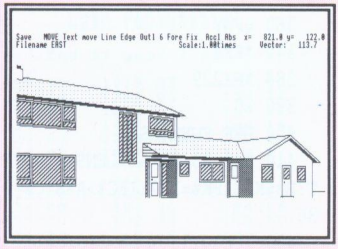
The book starts with an introduction to the basic principles of file handling, and in the following chapters develops an in-depth look at the handling of different types of files e.g. serial files, indexed files, direct access files, and searching and sorting. A separate chapter is devoted to hierarchical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

The topics covered by the book include:

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

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

ASTAAD



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

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

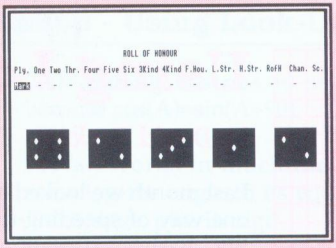
	Stock Code	Price		Stock Code	Price
ASTAAD (80 track DFS)	1407a	£5.95	**ASTAAD** (3.5" ADFS)	1408a	£5.95
Applications II (80 track DFS)	1411a	£4.00	**Applications II** (3.5" ADFS)	1412a	£4.00
Applications I Disc (40/80T DFS)	1404a	£4.00	**Applications I Disc** (3.5" ADFS)	1409a	£4.00
General Utilities Disc (40/80T DFS)	1405a	£4.00	**General Utilities Disc** (3.5" ADFS)	1413a	£4.00
Arcade Games (40/80 track DFS)	PAG1a	£5.95	**Arcade Games** (3.5" ADFS)	PAG2a	£5.95
Board Games (40/80 track DFS)	PBG1a	£5.95	**Board Games** (3.5" ADFS)	PBG2a	£5.95

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

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

Best of BEEBUG

Board Games



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

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

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

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

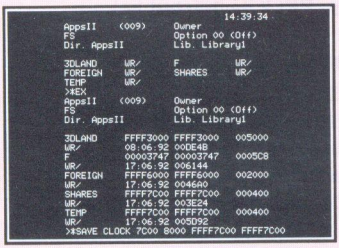
Cribbage - an authentic implementation of this very traditional card game for two, where the object is to score points for various combinations and sequences of cards.

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

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

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

Applications II Disc



Crossword Editor - for designing, editing and solving crosswords

Monthly Desk Diary - a month-to-view calendar which can also be printed

3D Landscapes - generates three dimensional landscapes

Real Time Clock - a real time digital alarm clock displayed on the screen

Running Four Temperatures - calibrates and plots up to four temperatures

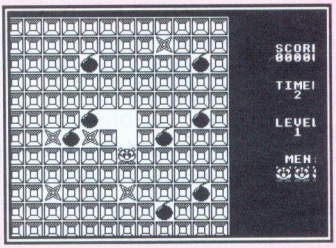
Julia Sets - fascinating extensions of the Mandelbrot set

Foreign Language Tester - foreign character definer and language tester

Share Investor - assists decision making when buying and selling shares

Label Processor - for designing and printing labels on Epson compatible printers

Arcade Games



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

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

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

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

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

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

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

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

	Stock Code	Price		Stock Code	Price
File Handling for All Book	BK02b	£9.95			
File Handling for All Disc (40/80T DFS)	BK05a	£4.75	**File Handling for All** Disc (3.5" ADFS)	BK07a	£4.75
Joint Offer book and disc (40/80T DFS)	BK04b	**£11.95**	**Joint Offer book and disc** (3.5" ADFS)	BK06b	**£11.95**
Magscan (40 DFS)	0005a	£9.95	**Magscan Upgrade** (40 DFS)	0011a	£4.75
Magscan (80T DFS)	0006a	£9.95	**Magscan Upgrade** (80T DFS)	0010a	£4.75
Magscan (3.5" ADFS)	1457a	£9.95	**Magscan Upgrade** (3.5" ADFS)	1458a	£4.75

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

BEEBUG WORKSHOP

Using Look-Up Tables

by David Peckett

Last month we looked at one way of speeding-up BBC Basic's calculations of sines and cosines. It was much faster than the computer's built-in routines, but only worked when the angles increased at steady rates.

We'll now go on to a way of using tables, or arrays, to speed things up. The approach means that a function's value can be looked up directly, rather than calculating it whenever it is needed. We can use tables to replace either specially written Basic functions or the computer's "intrinsic" functions (e.g. SIN, EXP and LOG).

Let's look at an easy example. Statistical work often needs to calculate factorials (for example 4!=4*3*2*1). It's easy enough to write a function to do the job, but then every one must be calculated whenever it's needed, which could slow things down quite a lot.

A different approach is to fill an array with all possible factorials. The procedure fills the array *Fact(33)* with the factorials of all integers from 0 to 33. By definition, 0! has the value '1', while 33! (value 8.6833E36) is the largest factorial the Beeb can handle. Run the procedure once at the start of your program and then, whenever you need a factorial, use the value from the array. For instance, the number of permutations of *P%* would be *Fact(P%)*. A suitable table can be set up with the procedure PROCfact listed below:

FACTORIALS

```
10000 DEF PROCfact
10010 DIM Fact(33)
10020 LOCAL i%
10030 Fact(0)=1
10040 FOR i%=1 TO 33
10050 Fact(i%)=Fact(i%-1)*i%
10060 NEXT
10070 ENDPROC
```

The number of permutations of P% items from a total of "N%" is:

```
Fact(N%)/Fact(N%-P%)
```

As an example of the speed improvement, it takes about 25.5 secs to calculate 500 random factorials directly, while using the table method takes only 1.0 secs, including setting up the table in the first place. The more factorials you need, the greater the advantage, but tables are quicker with as few as 4 calculations.

You are not just limited to your own routines, however - tables can just as easily replace trig functions, for example. It's a little more complex than factorials though, and needs a function to extract the data (if all angles are to be catered for) as well as a procedure to set it up.

TRIG FUNCTIONS

```
10000 DEF PROCanginit
10010 DIM Sintab(90),Tantab(90)
10020 FOR I%=0 TO 90
10030 Sintab(I%)=SIN(RAD(I%))
10040 Tantab(I%)=TAN(RAD(I%))
10050 NEXT
10060 ENDPROC

11000 DEF FNsin(ang)
11010 LOCAL sign
11020 sign=SGN(ang)
11030 ang=ABS(ang) MOD 360
11040 IF ang>180 THEN sign=-sign:
  ang=ang MOD 180
11050 IF ang>90 THEN ang=180-ang
11060 =Sintab(ang+0.5)*sign

12000 DEF FNcos(ang)
12010 =FNsin(ang+90)

13000 DEF FNtan(ang)
13010 LOCAL sign
13020 sign=SGN(ang)
13030 ang=ABS(ang) MOD 180
13040 IF ang>90 THEN sign=-sign:
  ang=180-ang
13050 =Tantab(ang+0.5)*sign
```

PROCanginit sets up tables of sines and tangents for single degree steps from 0 to 90 degrees. Once that's done, you can get the trig values for any angle whatsoever by using the fact that trig functions repeat themselves in a totally predictable way. For instance:

```
sin(-A)=-sin(A)
sin(A)=sin(A-360)
sin(A)=-sin(A-180)
```

FNsin uses relationships like these to get the sign and equivalent value, in the range 0 to 90 degrees, of any angle. It then extracts the sine of the equivalent angle from the array Sintab(), negates it if it has to, and returns the value. Calculating cosines is much easier, because cos(A)=sin(A+90).

FNTan works in much the same way, reducing the angle to its equivalent in a fixed range, plus its sign.

```
Trig functions - please wait

Enter angle in degrees: 73

Calculated directly:
SIN: 0.956304756      COS: 0.292371705
TAN: 3.27085262       TIME: 5

Calculated from tables:
SIN: 0.956304756      COS: 0.292371705
TAN: 3.27085262       TIME: 4
```

Using look-up tables for trig functions

These functions do not have the tremendous speed advantage of the factorials, but are still worth using. For instance, 500 random SINs take 12.5 secs, and TANs 22.2 secs, while the equivalent *FNsin* and *FNtan* take 7.5 secs and 6.45 secs respectively. Add the procedures and functions given above to the listing at the end of this article to provide a working demonstration of the comparative speeds of standard trig functions and the use of look-up tables.

If you are doing a lot of trig calculations, savings like this are well worthwhile but, inevitably, there is a price to pay. The procedures and functions take up memory, while *Sintab()* and *Tantab()* between them need over 900 bytes. You must allow for the time to run *PROCanginit*, so you have to be calculating at least 200 values before there is any overall speed advantage. Finally, the functions only give values for integer angles. Although you can put in

fractions of a degree, they are rounded to integers first. For example, *FNsin(1.5)* actually returns Sin(2.0). This gives a maximum error of around 0.0087, which is normally insignificant.

Why didn't I calculate tangents by using *FNsin(ang)/FNcos(ang)*? Try it, time it, and see! When using tables for trig functions, the arrays only need hold a limited range of values, because the functions are cyclic. We can then use this range to get the value for any possible angle. Many other functions are not symmetrical, and we need a different approach.

In such cases, set up the table for the most likely range of values, and use the normal Basic function for values outside that range. For example, suppose we normally need to use EXP on values from -5 to +5, but may have to handle any value.

EXPONENTIAL

```
10000 DEF PROCexpinit
10010 DIM Exp(100)
10020 LOCAL val%
10030 FOR val%=-50 TO 50
10040 Exp(val%+50)=EXP(val%/10)
10050 NEXT
10060 ENDPROC



11000 DEF FNexp(val)
11010 LOCAL ex
11020 IF val<-5 OR val>+5 THEN
  ex=EXP(val) ELSE
  ex=Exp(val*10+50.5)
11030 =ex
```

PROCexpinit sets up a 100-element array for the values of EXP(n) for n between -5 and +5, in 0.1 steps. *FNexp* returns EXP(val); if "val" is outside the -5 to +5 range, it uses the usual EXP function. Usually, however, it scales "val" to the range 0-100, extracts the value from the table, and returns that. The "50.5" term rounds the index to the nearest 0.1 step to reduce the errors.

You are not, of course, limited to using tables for just these functions, but can use them in many other cases. I've shown you different ways in which you might have to extract the data from the tables and you can pick'n'mix the methods for any particular case. The trick is to identify the range and step-size of the function, fill a suitable array, and write a function to pull out the right data.

Demonstrations of all routines listed are also included on the magazine disc in a demonstration Polar Curves program. By performing most of the calculations first, any subsequent graphics appear much smoother and faster.

Listing 1

```
   10 REM Program Angles
   20 REM Version B 1.0
   30 REM Author    David Peckett
   40 REM BEEBUG    August/September 1993
   50 REM Program subject to copyright
   60 :
   100 MODE7
   110 CLS:PRINT'"Trig functions - please
 wait"
   120 PROCanginit
   130 INPUT'"Enter angle in degrees: " a
ngle
   140  PRINT'"Calculated directly:"
   150  TIME=0:sin=SIN(RAD(angle)):cos=COS
(RAD(angle)):tan=TAN(RAD(angle)):time=TI
ME
   160 PRINT"SIN: ";sin," COS: ";cos,"TAN
: ";tan,"TIME: ";time
   170 PRINT'"Calculated from tables:"
   180  TIME=0:sin=FNsin(angle):cos=FNcos(
angle):tan=FNtan(angle):time=TIME
   190 PRINT"SIN: ";sin," COS: ";cos,"TAN
: ";tan,"TIME: ";time
   200 END
```

1st course

Sound (4)

Alan Wrigley concludes his look at programming the BBC micro's sound system.

Last month I started to describe the ENVELOPE command and the rather complex set of parameters required to use it. To recap, these are as follows:

```
ENVELOPE N,T,PI1,PI2,PI3,PN1,PN2,PN3,
AA,AD,AS,AR,ALA,ALD
```

In last month's article I gave a detailed description of the first eight parameters, from N to PN3. The remaining parameters will be described this month, and Figure 1 lists the functions of these.

As it happens, they are all fairly self-explanatory. AA to AR determine the change of amplitude for each step during the four stages of the envelope. These are attack, decay, sustain and release - refer to the illustration in last month's article if you are unclear about the meaning of these terms. The last two parameters, ALA and ALD, allow you to specify the amplitude level which should be reached by the end of the attack and decay phases respectively. Thus by carefully selecting the step length, the changes per step, and the final target volumes, you can achieve a wide range of amplitude envelopes (the pitch envelope was described last month and relies on the values supplied for PI1 to PN3).

Parameter	Range	Function
AA	-127 to 127	Change of amplitude per step during attack phase
AD	-127 to 0	Change of amplitude per step during decay phase
AS	-127 to 127	Change of amplitude per step during sustain phase
AR	-127 to 0	Change of amplitude per step during release phase
ALA	0 to 126	Target amplitude level at end of attack phase
ALD	0 to 126	Target amplitude level at end of decay phase

Figure 1. The last 6 parameters of the envelope command

You will notice that the values for AA and AD may be either positive or negative, while those for AS and AR may only be negative (or zero). This is because the volume can either increase or decrease during the attack and decay phases, but for the sustain and release phases it can only decrease or remain the same. Incidentally, if AR is given as zero then the note will continue indefinitely. Note that although values are given in the range 0-127, the hardware only supports 16 levels of amplitude, so any value above 16 will be treated as though 16 had been specified.

The easiest way to understand the use of the ENVELOPE command is to follow one through step by step, and we will take as our example the listing I gave at the end of last month's article:

```
10 ENVELOPE 1,2,1,-1,1,10,20,10,1,0,0,
-1,100,100
20 SOUND 1,1,100,80
```

If you run this short program, you will hear a siren-like sound (i.e. the pitch rises and falls) which increases in volume and then fades away. The first parameter is the envelope number, and here we have used 1. As explained last month, four envelopes are available as standard, and if your program makes no use of Basic's BPUT# statement, a further 12 envelopes are available (this is because the memory which would be used by BPUT# is free for the storage of

the extra envelopes). In the example here, the SOUND command in line 20 specifies envelope 1 in its 2nd parameter. It may seem obvious, but the program must have encountered an envelope definition *before* a SOUND command which uses it is issued. A common approach is to define all the envelopes to be used by a program during the initialisation procedure.

The step length (envelope parameter 2) has here been set to 2 centiseconds. The value you choose for this parameter may well have to be decided after a lengthy period of trial and error to get the most effective sound. One problem is that the same step length is used both by the pitch envelope and by the amplitude envelope, and in some circumstances these may have differing requirements. To give you some idea of how these may interact, consider the case of a short step time coupled with low values for the pitch change (PI1-PI3). This will result in a smooth transition of pitch from one level to another in a glissando fashion. Lengthening the step time and increasing the pitch change value will result in the individual pitch changes becoming more recognisable as such.

However, lengthening the step time will also increase the time it takes for the amplitude to change during each of the envelope phases. Thus if you double the step time, say, it will take twice as long to reach the target volume at the end of the attack phase for any given change of amplitude per step (parameter AA). Since the total time allocated for the attack-decay-sustain period of an envelope is determined by the D(uration) parameter of the SOUND command, you could find that the sound never reaches its intended maximum volume unless you also alter the amplitude step change and/or the total duration. This shows how inter-dependent everything is.

You can see how this works by altering the envelope as follows:

```
10 ENVELOPE 1,10,5,-5,5,10,20,10,1,0,0,
-1,100,100
```

This multiplies the step length and the pitch change per step by five in all cases, but leaves the other parameters intact. Running the program now illustrates the point that the pitch changes in more discrete steps, but the sound never reaches maximum volume before dying away. Multiplying the duration by 5 as well will cure this, at the expense of a very long overall sound:

```
SOUND 1,1,100,400
```

In our example here, bit 7 of the T parameter is zero (i.e. the pitch envelope is to be auto-repeated). Try setting this bit as follows:

```
10 ENVELOPE 1,130,1,-1,1,10,20,10,1,0,0,
-1,100,100
```

The siren will wail once, and then the sound will remain at the same pitch for the rest of the amplitude envelope. This can be useful for some types of sound, as we will see later.

Moving on to the pitch parameters, I have already explained that the pitch envelope is split up into three stages, and the pitch change per step and number of steps can be set independently for each stage, using PI1-PI3 and PN1-PN3 respectively. To create a wailing sound, therefore, you have to start at the central pitch, increase to the maximum in the first stage, decrease to the minimum in the second, and increase back to the start point in the third. This means that the pitch change is positive for stages 1 and 3 and negative for stage 2, while the length

of stage 2 is twice that of stages 1 and 3. This is shown in Figure 2. In the example we have specified 1 or -1 for the change of pitch per step; and 10 steps for the shorter stages and 20 steps for the longer.

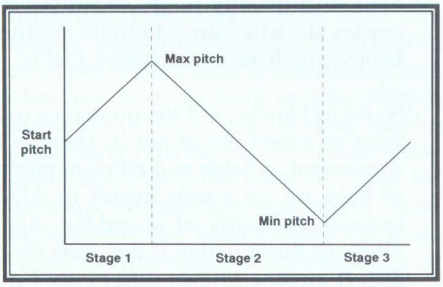


Figure 2. Using the 3 stages of the pitch envelope to construct a siren-like sound.

Having dealt with the repeating pitch envelope, we can now turn our attention to the amplitude envelope. We have decided that the sound we want must increase in volume to a certain point, hold that level for a short while, then fade away. In envelope terms, this means a gradual attack, no decay, a period of sustain, and finally a release phase which more or less mirrors the attack. We have (arbitrarily) decided that the volume at its maximum level should be 100, and so this must be given as the target volume for the end of the attack (ALA). Since there is no decay in this envelope, this value is also given for ALD (in other words, the end of decay will be the same level as the end of attack).

To achieve a gradual increase in volume, we have set AA to 1, which means that with a maximum volume of 100 it will take 100 steps, i.e. 2 seconds in this example, for the attack phase. AD is set to zero since there is no decay period, while AS is also set to zero as we don't want the sound to decay during the sustain. AR is set to -1 so that the fade at the end takes place at the same rate as the attack.

All it remains to do now is to decide the total duration of the sound. We have already said that, given the parameters specified, the attack phase will last 2 seconds. The release phase will take about the same time, but as I mentioned earlier this is not covered by the D parameter of the SOUND command anyway. The value we chose for D in the listing above was 80, which is 4 seconds (remember that duration is specified in 20ths of a second in this case). This means that the sustain period will last for a further 2 seconds before the sound starts to die away.

As you can see from this quite simple example, the ENVELOPE command can be extremely complex, but it can also be very rewarding if you take the trouble to plan your sounds and experiment with them. It is worth giving a couple of further examples which illustrate the inter-dependence of all the elements of the command. Firstly, here is an example which uses a non-repeating pitch envelope:

```
10 ENVELOPE 1,129,1,-1,0,20,20,0,100,0,
0,-1,100,100
20 SOUND 1,1,10,5
```

The final example illustrates a point which has not yet been mentioned. If the result of a pitch change takes the value below zero or above 255, the value wraps around. For example, if the current pitch is 200 and the change of pitch per step is 100, when the pitch changes at the next step it will become 45. By setting a large pitch step so that wrap-around occurs frequently, you can create some interesting sounds, for example:

```
10 ENVELOPE 1,129,50,-50,0,20,20,0,100,
0,0,-100,100,100
20 SOUND 1,1,10,5
```

512 Forum

Robin Burton looks at some of the problems and solutions involved in backing up discs on a 512.

At last, this month we finally get round to PKZIP. However, recent 512 converts may be unaware of the capabilities of archiving software in larger systems than the BBC micro, so we'll start with an overall introduction to the topic.

First though, I must deal with a couple of other brief matters.

ESSENTIAL SOFTWARE

I won't be renewing Essential Software's PO box this year; it costs over a hundred pounds and really is no longer justified. From September anyone who wants to contact me or Essential Software should write to:

24 Wallace Drive, Groby, Leics. LE6 0GQ

The second point is that for similar reasons, please note that in future all payments should be made payable to me personally. Now back to our main topic.

BACKING UP

One of the most onerous tasks in any micro is taking regular and adequate back-ups (hands up if you always have perfect back-ups! Hmm, well you two are untypical).

One reason for poor back-ups is that it's a job which, ideally, is a complete waste of time. More often than not it is of course, which reinforces that attitude. That, I'm sure is one factor, but it must be admitted that for the 512 Acorn didn't do much to make life easy. They didn't for any of their systems for that matter, but at least in the BBC the amount of data isn't usually huge, so simple disc copies do offer an adequate method. Unless you have a winchester, that is!

So far as I know, and the numerous users who've asked me about it, there is no convenient, reliable and efficient method of backing up a winchester in ADFS. Indeed in terms of standard Acorn offerings the situation is the same for the 512. If anyone knows differently I'd be interested to hear about it and pass it on.

I did try BACKUP a very long time ago, but it proved to be consistently useless, just about the only thing about it that was consistent. It doesn't just fail to work, if you try to use it there's no way of guessing how far it will get before it fails and which randomly selected error it will produce this time. I'm not sure whether BACKUP itself or the ADFS is at fault, but I'm not over impressed by ADFS either. The stand-alone version for the model B and B+ is truly dreadful, although I presume the Master's Mega-ROM versions are rather more reliable.

Happily, in DOS, the situation is much better, and there's quite a number of options. Given that DOS files tend to be so much larger than BBC files it's just as well, but recent PC methods are of no use for the 512.

Most business users of PCs these days employ a tape drive for back-up. These are adequately rapid devices, usually driven by automated software, so that backing up a couple of hundred megabytes or so can be left to run on its own when business is over for the day.

For non-stop operation on network file-servers, with ever falling hardware costs, RAID systems (multiple duplicate hard drives) are now the norm, again normally with tape back-up. Even so there's a huge number of individual PC users who still use floppy discs for back-up. For limited amounts of data they're reasonably cost effective and, so long as you keep a duplicate of each back-up disc, they're adequately reliable too. Of course a few years ago everyone used floppies for back-up, so basically successful back-up software has been around for some time, is thoroughly developed and highly reliable.

Low-level access to disc hardware isn't needed for backing-up or archiving as it often is in 'clever' disc utilities; standard DOS filing system calls are quite sufficient for most back-up/archiving programs. As a result the majority can be expected to work well in the 512, even on 800K discs. Since even the DOS partition of a 512's winchester looks like a standard drive too, hard disc files can be usually be backed up by the same software.

FUN AND GAMES

The main feature of most backup systems is data compression, the purpose of which is simply to cram more data onto the back-up medium than would otherwise fit. If you've used only an 8-bit machine like the BBC micro in the past you won't be familiar with this process - 8-bit machines just aren't up to the job, particularly in terms of speed. Appropriate compression methods require manipulation of units of data varying in size between 8 and 16 bits, not easy in an 8-bit system, plus a lot of large table indexing, something else that 8-bit machines aren't good at.

The prime benefit of data compression is that it can considerably reduce the amount of back-up media, but there are other gains too. For example, if you compress data to say, half its original size, not only do you get twice as much on a disc, but reading or writing a given amount of it is effectively twice as fast. In the 512 either your back-ups are on floppy or you don't have any, so speed is just as important to me as reducing the number discs. Another plus, at least in the software I use, is that the compressed data has numerous integrity checks built in so you can be sure that recovered data is reliable.

Those of you who recently had a copy of PCCE from me will know that it was delayed because I helped David Harper to eliminate a conflict which would affect hard disc users. Unfortunately, the way I discovered the problem was by the total corruption of the FAT on my hard disc (twice in one weekend, Grrrr!).

Some of you probably know the feeling when you see that wonderfully explicit message 'Outside file on channel 57' as you try to boot the 512. In case you've wondered, it means the FAT is directing the ADFS file pointer to somewhere outside the partition (itself an ADFS file) as the 512 tries to load DOS Plus. It's at this point that you reflect on your optimistic approach to backups and vow (once again) to be more disciplined in future. There's nothing else for it, you have to reallocate the DOS partition and recover all (well, nearly all) the disc's previous contents from back-ups.

My partition is 25Mb. but contained only (only? Huh!) about 17Mb of data in at the time. Obviously using simple disc copies, even with 800K discs, this amount of data would require upwards of 25

floppies, the number depending on how big individual items of data might be. In the event, thanks to data compression I was able to recover (most of) the data from a dozen 800K discs.

Of course not all files can be compressed; some are much better candidates than others. Text is usually very compressible, most programs, particularly .EXE files, are less so. Graphics files present a special case - some (e.g. simple bit-maps) are extremely compressible, while others are effectively already compressed by the application that produced them. These could even grow in size if you compressed them again, so compression software should check for this and store such files as they stand.

The extremes of compression performance are fascinating. If anyone is looking for a programming project to while away the coming winter nights I recommend data compression. You might recall that I wrote a compression system a few months ago, but despite the fact that I designed and coded it from 'scratch' I still don't entirely understand how the de-compression works. At its best my program compressed one 33K test file into less than 256 bytes while random samples of text averaged 50 to 75%. At the other end of the scale, much over 5% was good for most .EXE files. My routine may not be the best ever written, but such varying results are representative of what to expect from any compression system.

WHAT TO USE?

OK, data compression is a good idea, but what program should you use?

PC-Tools, which I've mentioned before, has a back-up system which offers various degrees of compression balanced by a small trade-off against speed, but I must admit I've never tried this feature myself in the 512. I use an old version of PC-Tools, version 5.5, and almost all the system works well, but I doubt this is as true, if at all, for recent versions. For what it's worth I've seen versions 7 and 8 in action, and I don't like the screens as much as earlier versions either. If you try PC-Tools buy an old version. You'll need a hard disc to store the full system though, as it's not really intended for floppy users.

I also know of a couple of 512 users who use Flexi-bak Plus, a very successful U.K. written shareware program for DOS back-up, and users are very happy with it. However, this is another I've never tried myself and I can't advise on versions if recent ones don't suit the 512. What I would say is that from its reputation alone Flexi-bak Plus is well worth a look if you're in the market for a DOS backup system. Since it's shareware the cost of evaluation is minimal.

The system I use for back-up is PKware's PKZIP. This is also shareware and is the successor to PKware's earlier system, PKARC, which served the same purpose until ZIP replaced it. There's little doubt that PKware have virtually made the task of archived data compression their own, PKZIP is the standard tool for DOS data compression, as was PKARC before it.

All versions of PKZIP up to and including the latest, 2.04G, are backward compatible, that is to say that they will unzip data from archives produced by all previous versions of PKZIP. However, an important point to note is that PKUNZIP will not extract data from PKARC archives - you still need PKXARC for that. You'll still come across .ARC files, in some shareware issues for example, but PKXARC usually will be included on the disc in such cases. If not, the only

solution is to find and keep a copy of PKXARC yourself, or PAK mentioned below.

There are a few other compression/archiving systems which offer broadly similar facilities to PKZIP, LHarc is a well known and, by some, preferred example. Others you might try include ARJ which has some unique options and PAK which can handle .ARC and .ZIP files as well as its own .PAK archives. These too are shareware, as indeed are virtually all such tools unless they're part of a larger utility package such as PC-Tools. This is one area where shareware authors 'got in' first and have done such a good job that most commercial software publishers decline to compete.

DISADVANTAGES

Obviously the system you choose might be any of the above, but I can only recommend PKZIP based on my own experiences. I've used PKZIP and PKARC before for many years with no problems whatsoever, apart from an occasional faulty disc which of course is nothing to do with the archiving program.

There is one criticism I'd accept of PKZIP (and PKARC too) which is that the command line syntax does tend to become a bit obscure once you leave the simplest operations behind. This complexity is partly a function of the fact that PKZIP has a very extensive range of options, but equally because of performance objectives.

As soon as you run the program you'll realise that PKZIP is all about speed and efficiency above all else, in which it is certainly impressive. Fancy menus and interactive selections are generally counter productive in this context, so rather than offering any sort of dialogue to allow you to select options these must be specified in the command line entry.

As it stands PKZIP can be quite daunting to the new user. In fact despite my unreserved recommendation of the software, I doubt that anyone, including those who wrote it, know how to use all the switches and options it offers without checking again first. All is not lost though. Also in shareware are a number of 'front-ends' and utilities which aim to make PKZIP more friendly and easier to use.

Equally, although I use some of the more complex options for my back-up and recovery procedures I can't remember off-hand what they are, but it doesn't matter. The relevant instructions are in two batch files on each of my backup discs, one called SECURE.BAT, the other RECOVER.BAT. The names may not be very imaginative, but they are easy to remember and they work when I need them when I need them, which is all that matters.

NEXT MONTH

Next month we'll explore some of the options in PKZIP (and UNZIP) so as to allow you to set up a reliable, largely automated back-up and recovery system. The only problem then is keeping that up to date!

Any shareware house will be sure to have PKZIP, so you'll have plenty of time to get hold of a copy if you don't already have it but feel like giving it a try.

Points Arising... Points Arising...

We would like to apologise for the unfortunate transposition of two illustrations in the July issue of BEEBUG (Vol.12 No.3). The pages concerned are 22 and 34.

Census (Part 2)

Paul Goldsmith continues his saga of questioning folk.

Now that you have had a chance to see how to set up the questionnaires we will complete the *Create* and *Census* programs to give you full editing and saving facilities.

Load Create and add the lines in *Create Part 2* as listed below, remove *PROCcode* from line 210, then save it. This will add the instruction procedure. Both Programs have an associated instruction text (provided on this month's disc) which can be accessed from their menus. These are stored in files as straight ASCII without control codes.

To create your own instructions you should follow these guidelines: texts should be formatted to 37 characters per line but do not use justify or the Tab key. The instructions for Create should be saved as *INSTCRE* and Census instructions as *INSTCEN*.

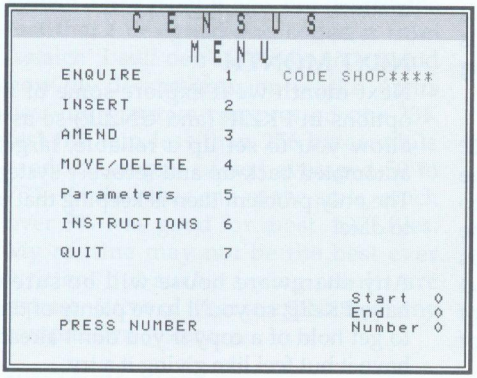


Figure 1. Census menu

Now load Census and add the lines in *Census Part 2* as listed (you may like to add :VDU7: just before the INPUT statements in lines 1480 and 2870 - we found this useful), then save it. The new facilities are as follows.

Selecting 'Enquire' and choosing a question and answer should result in a report 'Record not Found' until the record has been inserted. Once you have some records 'Enquire' allows you to browse through questions and records.

'Insert' will work just as before but now the data will be saved to disc, as will any of the alterations made by the other facilities.

'Amend' allows the amendment of any answer to any question in any record already entered.

'Delete/Move' enables records to be deleted or data moved to another record number. Moving a record is done by first copying the record to the new position, once this is done you will be asked if you want to delete the record from its original position. Answering 'N' here will effectively copy, rather than move, the record. This can help in the entry of lots of similar replies.

Always 'Quit' Census from the menu using option 7. This updates the file <code>PAR. If you think that the information in the bottom right hand corner of the menu screen is wrong select 'Parameters' and the program will update the information by reading the Data file <code>DA.

NEW BITS

The program *Select* is the 'Front end' menu for the suite of programs. Type in the listing and add the procedures *DEFPROCwindow1, DEFPROCwindow2, DEFPROCwindow3, DEFPROCblue* and *DEFPROCt* from Census to complete the program - this must be saved as *Select*. If you want a boot file on your disc it

should chain Select. Instructions for Select should be held in a file called *INSTSEL* as described above.

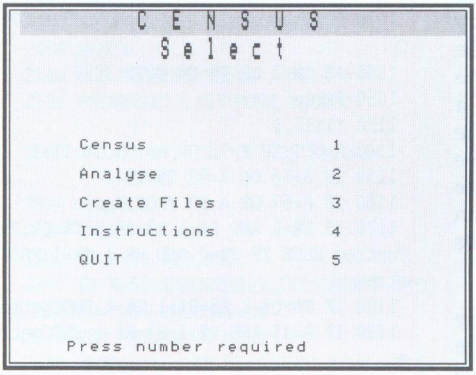


Figure 2. Select menu

When run, Select provides a route between Create, Census and *Analyse*, a sub-menu program that we will deal with now. Type in the listing for Analyse below, adding the procedures as you did for Select. Analyse, again, must be saved under that exact name and it's worth stressing that all the parts of this application must be on the same disc, unless you are very patient. The instructions for Analyse are held in *INSTANA*.

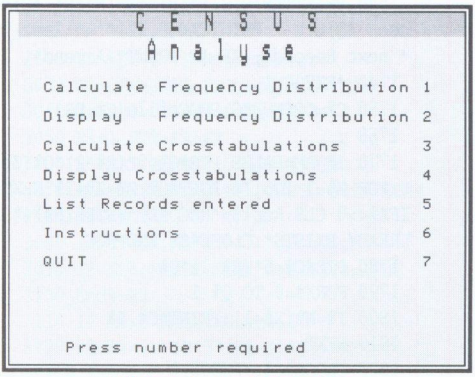


Figure 3. Analyse menu

Select should now be fully working with no 'Not Found' errors, Analyse will only take you back to Select. Now is a good time to enter the results of a fullish survey and test the above programs for typing errors. A substantial amount of data will also give you something to test the statistical programs on as we get to them.

MEANWHILE....

Having entered all your questionnaire data and satisfied yourself that all the responses have been correctly entered lock the <code>DA file using *ACCESS. Your disc drive instructions will tell you how to do this. Then, by selecting 'Parameters' from the Census menu update the <code>PAR file. This is important because it controls the analysis programs.

The Analyse sub-menu gives you access to all the statistical procedures available, it's also the place to add your own if you want to. There's room this month to deal with two of the programs it calls.

Listrec, the 5th option on the Analyse sub-menu, allows the production of a list of all records entered, showing missing records as asterisks; the list is sent to your printer. Type this in and save it as *Listrec*, there are no additional procedures. To help keep track of what is going on as your survey develops, and to be able to identify individual forms from Listrec, it is advisable to number questionnaire forms. However, the identity of the person to which the data relates must not be shown on the form or be available at the time the data is entered or used. If you do hold any kind of personal information remember that you will be subject to the Data Protection Act.

Now we're going to add the first piece of actual number crunching. Type in the listing for *Freqdu*, adding *PROCblue* and *PROCt* from the Census; Freqdu must be saved under that name. This program,

option 1 on the Analyse sub-menu, will let you save the frequency distribution data. This data is saved by the program as <code>DI and the program provides for a printout of the raw frequency distribution information. This can be used for an initial assessment of the results but next month I will give you a program for more lucid displays of the results, including bar charts which can be saved to disc and printed using a suitable screen printing program. The program also allows for the spooling of the results for transfer to the VIEW family.

Now you can create, enter and save the results of your survey and do a little processing of the results. Next month we will continue filling in the gaps in the Analysis sub-menu towards the complete package.

Listing 1

```
  10 REM CREATE PART 2
 160 PRINT''"Instructions"TAB(33)"2"
1110 PROCwindow3
1120 CLS:C%=OPENIN"INSTCRE":PROCefile(C
%,"INSTCRE"):CLOSE#C%
1130 PRINT"SHIFT to page on"
1140 PRINT"'R' Repeat     SPACE end"
1150 VDU26:PROCwindow1
1160 CLS:VDU14:*TYPE INSTCRE
1170 A=GET
1180 IF A=82 GOTO 1160
1190 VDU15,26:CLS
```

Listing 2

```
  10 REM CENSUS PART 2
 180 PRINTTAB(5,4)"ENQUIRE        1"
 200 PRINTTAB(5,8)"AMEND          3"
 210 PRINTTAB(5,10)"MOVE/DELETE   4"
 220 PRINTTAB(5,12)"Parameters    5"
 230 PRINTTAB(5,14)"INSTRUCTIONS  6"
 300 IF A=49 PROCenquire
 320 IF A=51 PROCamend
 330 IF A=52 PROCmove
 340 IF A=53 PROCparameter
 350 IF A=54 PROCt(8,"Instructions"):PR
OCinst
1010 CLS:INPUTTAB(10,10)"WHICH RECORD "
;R%:IF R%>Rmax% OR R%<1 VDU7:GOTO1010
1020 INPUTTAB(10,12)"WHICH QUESTION ";T
%
1030 PROCgetrec:IF Rep$<>"" VDU7:ENDPRO
C
1040 IF T%<1 OR T%>Q% GOTO 1020
1050 PROCe
1130 *FX15,1
1140 A=GET:IF A=78 OR A=101 T%=T%+1
1150 IF A=66 OR A=98 T%=T%-1
1160 IF A=67 OR A=99 PROCchoose
1170 IF T%=0 AND R%>1 R%=R%-1:T%=Q%:PRO
Cgetrec ELSE IF T%=0 AND R%=1:T%=1:PROCg
etrec
1180 IF T%=Q%+1 R%=R%+1:T%=1:PROCgetrec
1190 IF A=44 AND R%>1 R%=R%-1:PROCgetre
c
1200 IF A=46 R%=R%+1:PROCgetrec
1210 IF A=81 GOTO 150
1220 IF A=65 AND AM=TRUE PROCin:PROCa
1230 IF Rep$<>"" VDU7:GOTO150
1270 PROCwindow3
1280 CLS:C%=OPENIN"INSTCEN":PROCefile(C
%,"INSTCEN")
1290 PRINT"SHIFT to page on"
1300 PRINT"'R' Repeat     SPACE end"
1310 VDU26:PROCwindow1
1320 CLS:VDU14:*TYPE INSTCEN
1330 A=GET
1340 IF A=82 GOTO 1320
1350 VDU15,26:CLS
1540 PRINTTAB(0,21)"(B)ack (N)ext (C)ho
ose  (Q)uit ":PRINTTAB(0,22)"'<' last '>
' next Record";:IF AM PRINT"(A)mend"
1740 *FX200,1
1750 C%=OPENUPD$:PROCefile(C%,D$)
1760 :
1770 IF AM=FALSE PTR#C%=5*(R%-1)*Q%:Z%=
0:FOR X%=1 TO5:I%=BGET#C%:Z%=Z%+I%:NEXT:
IFZ%<>0 CLS:Rep1$="RECORD "+STR$(R%)+" A
LREADY EXISTS":CLOSE#C%:ENDPROC
1780 PTR#C%=5*(R%-1)*Q%
1790 FORX%=0 TO Q%-1
1800 I%=N%(X%+1):PRINT#C%,I%
1810 NEXT
1820 CLOSE#C%:*FX200,0
1830 PROCsaveparam
2410 AM=TRUE:HD$="Amend"
2420 CLS:INPUTTAB(10,10)"WHICH RECORD "
```

```
;R%:IF R%>Rmax% OR R%<1 VDU7:GOTO2420
 2430 INPUTTAB(10,12)"WHICH QUESTION ";T
%:IF T%<1 OR T%>Q% VDU7:GOTO 2430
 2440 PROCgetrec
 2450 IF Rep$<>"" VDU7:ENDPROC
 2460 PROCa
 2540 CLS:PROCt(8,"Move / Delete")
 2550 PRINTTAB(3,10)"MOVE RECORD
         1"
 2560 PRINTTAB(3,12)"DELETE RECORD
         2"
 2570 *FX15,1
 2580 A=GET
 2590 IF A=49 PROCshift
 2600 IF A=50 INPUTTAB(3,17)"DELETE WHIC
H RECORD ";R%:PROCdelete
 2610 REM RETURN TO MENU
 2650 PROCgetrec:IF N%=0 Rep$="Record "+
STR$(R%)+" does not exist":ENDPROC
 2660 IF Miss$="Y" A$="Y":Miss$="":GOTO2
680
 2670 PRINTTAB(3,20)"Are you sure you wa
nt to ":PRINTTAB(3,21)"delete Record No
";R%:INPUTTAB(3,22)"Press 'Y'/'N' then R
ETURN "A$
 2680 IF A$="Y" C%=OPENUPD$:PROCefile(C%
,D$)
 2690 IF A$="Y" THEN PTR#C%=(R%-1)*5*Q%:
FORT%=1TOQ%*5:BPUT#C%,0:NEXT:Rep$="Recor
d No "+STR$(R%)+" Deleted":CLOSE#C%:A$="
":Numb%=Numb%-1:PRINTTAB(3,20)"Updating
Parameters"TAB(3,22)"          "TAB(3,21
);SPC(30);TAB(3,22);SPC(30):PROCparamete
r
 2695 IF A$="Y" PROCparameter
 3030 C%=OPENIND$:PROCefile(C%,D$)
 3040 Start%=0:End%=0:Numb%=0
 3050 R%=1:ST%=FALSE
 3060 REPEAT:PTR#C%=5*(R%-1)*Q%
 3070 B%=BGET#C%:IF B%=0 GOTO 3120 ELSE
PTR#C%=PTR#C%-1
 3080 INPUT#C%,N%
 3090 IF ST%=FALSE Start%=R%:ST%=TRUE
 3100 End%=R%
 3110 IF N%>0 Numb%=Numb%+1
 3120 R%=R%+1: IF 5*(R%-1)*Q%>=EXT#C%:CL
OSE#C%:ENDPROC
 3130 UNTIL EOF#C%
 3140 CLOSE#C%
 3150 PROCsaveparam
```

Listing 3

```
  10 REM Program Select
  20 REM Version B 3.1
  30 REM Author  Paul Goldsmith
  40 REM BEEBUG  Aug/Sep 1993
  50 REM Program Subject to Copyright
  60 :
 100 ONERROR PROCerr
 110 MODE 7:PROCt(9,"S e l e c t")
 120 VDU31,6,9:PRINT"Census
   1"
 130 PRINT:PROCb
 140 VDU31,6,11:PRINT"Analyse
    2"
 150 PRINT:PROCb
 160 VDU31,6,13:PRINT"Create Files
    3"
 170 PRINT:PROCb
 180 VDU31,6,15:PRINT"Instructions
    4"
 190 PRINT:PROCb
 200 VDU31,6,17:PRINT"QUIT
    5"
 210 PRINTTAB(5,23)"Press number requir
ed"
 220 *FX15
 230 A=GET
 240 IF A=49 CHAIN"CENSUS"
 250 IF A=50 CHAIN"ANALYSE"
 260 IF A=51 CHAIN"CREATE"
 270 IF A=52 PROCinst:GOTO 120
 280 IF A=53 CLS:PRINTTAB(5,10)"Good by
e from CENSUS":END
 290 ELSE GOTO220
 300 END
 310 :
1000 DEFPROCinst
1010 PROCwindow3
1020 CLS:C%=OPENIN"INSTSEL":PROCefile(C
%,"INSTSEL"):CLOSE#C%
1030 PRINT"SHIFT to page on"
1040 PRINT"'R' Repeat    SPACE end"
1050 VDU26:PROCwindow1
1060 CLS:VDU14:*TYPE INSTSEL
1070 A=GET
1080 IF A=82 GOTO 1060
1090 VDU15,26:CLS
1100 ENDPROC
1110 :
1120 DEFPROCintro
```

```
1130 PROCwindow1PRINTTAB(7,10)"A Social
Survey"
1140 PRINTTAB(8,12)"and Analysis"
1150 PRINTTAB(10,14)"Program"
1160 ENDPROC
1170 :
1180 DEFPROCb
1190 VDU132,157,131
1200 ENDPROC
1210 :
1220 DEFPROCerr
1230 REPORT:PRINT" at line "ERL
1240 PRINT"Press a key":A=GET
1250 ENDPROC
1260 :
1270 DEF PROCefile(i%,I$)
1280 IF i%=0 PRINTTAB(3,19)"File "+I$+"
not found":CLOSE#i%:TIME=0:REPEATUNTIL
TIME=200:RUN
1290 ENDPROC
```

Listing 4

```
  10 REM Program Analyse
  20 REM Version B 4.1
  30 REM Author Paul Goldsmith
  40 REM BEEBUG Aug/Sep 1993
  50 REM Program Subject to Copyright
  60 :
 100 ONERROR PROCerr
 110 MODE 135:REM MODE 7 For BBC B
 120 CLS:PROCt(8,"A n a l y s e")
 130 PRINTTAB(3,5)"Calculate Frequency
Distribution 1"
 140 PRINTTAB(3,7)"Display   Frequency
Distribution 2"
 150 PRINTTAB(3,9)"Calculate Crosstabul
ations       3"
 160 PRINTTAB(3,11)"Display Crosstabula
tions        4"
 170 PRINTTAB(3,13)"List Records entere
d            5"
 180 PRINTTAB(3,15)"Instructions
             6"
 190 PRINTTAB(3,17)"QUIT
             7"
 200 PRINTTAB(5,23)"Press number requir
ed"
 210 GOTO220
 220 *FX15
 230 A=GET
 240 IF A=49 CHAIN"FREQDU"
 250 IF A=50 CHAIN"FREQDI"
 260 IF A=51 CHAIN"CROSSDU"
 270 IF A=52 CHAIN"CROSSDI"
 280 IF A=53 CHAIN"LISTREC"
 290 IF A=54 PROCinst
 300 IF A=55 CHAIN"SELECT"
 310 RUN
 320 END
 330 :
1000 DEFPROCinst
1010 PROCwindow3
1020 CLS:C%=OPENIN"INSTANA":PROCefile(C
%,"INSTANA")
1030 PRINT"SHIFT to page on"
1040 PRINT"'R' Repeat    SPACE end"
1050 VDU26:PROCwindow1
1060 CLS:VDU14:*TYPE INSTANA
1070 A=GET
1080 IF A=82 GOTO 1060
1090 VDU15,26:CLS
1100 ENDPROC
1110 :
1120 DEFPROCerr
1130 REPORT:PRINT" at line ";ERL
1140 PRINT"Press a key":A=GET
1150 ENDPROC
1160 :
1170 DEF PROCefile(i%,I$)
1180 IF i%=0 PRINTTAB(3,19)"File "+I$+"
not found":CLOSE#i%:TIME=0:REPEATUNTIL
TIME=200:CHAIN"ANALYSE"
1190 ENDPROC
```

Listing 5

```
  10 REM Program Listrec
  20 REM Version 1.1
  30 REM Author  Paul Goldsmith
  40 REM BEEBUG  Aug/Sep 1993
  50 REM Program Subject to Copyright
  60 :
 100 PROCcode
 110 MODE135:DIM N$(H%),T%(J%),N%(J%),A
%(J%),A$(J%,H%):VDU 15:Rep$="":Rep1$="":
Miss$=""
 120 PROCgetparam
 130 CLS
 140 C%=OPENIND$:PROCefile(C%,D$)
 150 PRINT"CHECK PRINTER ":A=GET
 160 VDU2
```

```
  170 PRINT"Listing of Records Entered f
or "F$
  180 FOR R%=Start% TO End%
  190 PTR#C%=5*(R%-1)*J%
  200 N%=BGET#C%
  210 IF N%=0 PRINT"          *"; ELSE PR
INTR%;
  220 IF R%MOD5=0 PRINTCHR$(&0D)
  230 NEXT
  240 CLOSE#C%
  250 PRINT:PRINT"End of Listing ";
  260 *TIME
  270  REM MASTER ONLY
  280 VDU3
  290  CHAIN"ANALYSE"
  300 END
  310 :
 1000 DEF PROCgetparam
 1010 C%=OPENINP$:PROCefile(C%,P$)
 1020 INPUT#C%,Start%:INPUT#C%,End%:INPU
T#C%,Numb%:INPUT#C%,Top%:INPUT#C%,Rmax%
 1030 CLOSE#C%
 1040 ENDPROC
 1050 :
 1060 DEFPROCcode
 1070 C%=OPENIN"CODE":PROCefile(C%,"CODE
")
 1080 INPUT#C%,F$,J%,H%,K%:CLOSE#C%
 1090 D$=F$+"DA":H$=F$+"HED":G$=F$+"QUE"
:Code$=F$+"****":P$=F$+"PAR"
 1100 ENDPROC
 1110 :
 1120 DEF PROCefile(i%,I$)
 1130 IF i%=0 PRINTTAB(3,19)"File "+I$+"
 not found":CLOSE#i%:TIME=0:REPEATUNTIL
TIME=200:CHAIN"ANALYSE"
 1140 ENDPROC
```

Listing 6

```
   10 REM Program Freqdu
   20 REM Version B 4.2
   30 REM Author Paul Goldsmith
   40 REM APR 1993
   50 REM Program Subject to Copyright
   60 :
  100 PROCcode
  110 DIMT%(J%),N%(J%),A%(J%),A$(J%,H%),
Z%(H%,H%),Array%(H%):Rep$="":VDU15:PROCz
er:MODE135:REM MODE 7 for BBC B
  120 H$=F$+"HED":D$=F$+"DA":S$=F$+"DI":
P$=F$+"PAR"
  130 CLS:PROCt(5,"Frequency Distibution
")
  140 PROCcontrol:IF C%>0 PRINTTAB(3,6)"
Control loaded"
  150 PROCgetparam:IF C%>0 PRINTTAB(3,8)
"Parameters Loaded"
  160 Print=FALSE:INPUTTAB(3,10)"SEND TO
 PRINTER Y/N "A$:IF INSTR("Yy",A$) Print
=TRUE
  170 PROCarray
  180 MODE131:PROCdisp:REM MODE 4 for BB
C B
  190 PROCfile
  200 CHAIN"ANALYSE"
  210 :
 1000 DEFPROCcontrol
 1010 ONERROROFF:C%=OPENINH$:PROCefile(C
%,H$)
 1020 INPUT#C%,Head$,Q%:FOR X%=1 TO Q%:I
NPUT#C%,A%(X%):FORY%=0TOA%(X%):INPUT#C%,
A$(X%,Y%):NEXT:NEXT
 1030 CLOSE#C%
 1040 ENDPROC
 1050 :
 1060 DEFPROCarray
 1070 :
 1080 R%=Start%-1:ONERROROFF:A%=OPENIND$
:PROCefile(A%,D$)
 1090 R%=R%+1:PRINTTAB(5,18)"Record ";R%
 1100 P%=5*(R%-1)*Q%:IF P%+5*Q%>EXT#A% G
OTO1200
 1110 PTR#A%=5*(R%-1)*Q%
 1120 B%=BGET#A%:IF B%<>&40 GOTO 1190:EL
SE PTR#A%=PTR#A%-1
 1130 FORT%=1TOQ%
 1140 INPUT#A%,N%
 1150 FORX%=A%(T%)TO1 STEP-1
 1160 A1%=A%(T%)+1-X%:IF N%>=2^(X%-1) Z%
(A1%,T%)=Z%(A1%,T%)+1: N%=N%-2^(X%-1)
 1170 NEXT
 1180 NEXT
 1190 IF R%<=End% GOTO 1090
 1200 CLOSE#A%
 1210  :
 1220 ENDPROC
 1230 :
 1240 DEFPROCzer
 1250 FORX%=1TOH%:FORY%=1TOH%:Z%(X%,Y%)=
0:NEXT:NEXT
```

```
 1260 ENDPROC
 1270 :
 1280 DEFPROCdisp
 1290 @%=&00003
 1300 IF Print=TRUE VDU2,1,27,69
 1310 VDU14:PRINTTAB(5,1)"FREQUENCY DIST
RIBUTION ";F$+" CENSUS":VDU28,0,24,79,2
 1320 FORX%=1TOJ%:FORY%=1TOA%(X%):PRINTZ
%(Y%,X%);:NEXT:PRINTTAB(58);"QUESTION ";
X%;:PRINTTAB(70);A%(X%);" ANS":NEXT
 1330 FOR X%=1 TO H%:PRINTX%;:NEXT:PRINT
 1340 PROCtime: VDU3:PRINTTAB(0,20)"Pres
s a key":A=GET
 1350 ENDPROC
 1360 :
 1370 DEFPROCfile
 1380 F%=OPENOUTS$
 1390 FORX%=1TOJ%:FORY%=1TOH%:PRINT# F%,
Z%(Y%,X%):NEXT:NEXT
 1400 CLOSE#F%
 1410 ENDPROC
 1420 :
 1430 DEF PROCgetparam
 1440 ONERROROFF:C%=OPENINP$:PROCefile(C
%,P$)
 1450 INPUT#C%,Start%:INPUT#C%,End%:INPU
T#C%,Numb%:INPUT#C%,Top%:INPUT#C%,Rmax%
 1460 CLOSE#C%
 1470 ENDPROC
 1480 :
 1490 DEFPROCcode
 1500 ONERROROFF:C%=OPENIN"CODE":PROCefi
le(C%,"CODE")
 1510 INPUT#C%,F$,J%,H%,Rmax%:CLOSE#C%
 1520 ENDPROC
 1530 :
 1540 DEFPROCtime
 1550 *TIME
 1560 REM MASTER ONLY
 1570 ENDPROC
 1580 :
 1590 DEF PROCefile(I%,I$)
 1600 IF I%=0 PRINTTAB(3,19)"File "+I$+"
 not found":CLOSE#I%:TIME=0:REPEATUNTIL
TIME=200:CHAIN"ANALYSE"
 1610 ENDPROC
```

Star LC24-200 Screen Dump (continued from page 22)

```
 1690 lda byte+3: jsr printer: lda byte+
4: jsr printer
 1700 lda byte+5: jsr printer \ pins 17
to 20
 1710 lda byte: jsr printer: lda byte+1:
 jsr printer
 1720 lda byte+2: jsr printer \ pins 17
to 20
 1730 \ inc Y co-ord by 4
 1740 clc: lda #4: adc Ylo: sta Ylo
 1750 bcc nocarry: inc Yhi
 1760 \ test for top of screen (&400 = 1
024 points)
 1770 :
 1780 .nocarry
 1790 lda Ylo: cmp #&00: beq hiY: jmp pi
xel
 1800 .hiY
 1810 lda Yhi: cmp #&4: beq lf:jmp pixel
 1820 .lf
 1830 lda #10: jsr printer \ line feed
 1840 \ test for rt edge of screen (&500
 = 1280 points)
 1850 lda Xlo: cmp #0: beq hi: jmp ycoor
d
 1860 .hi
 1870 lda Xhi: cmp #5: beq reset: jmp yc
oord
 1880 \ reset line spacing
 1890 :
 1900 .reset
 1910 lda #27: jsr printer
 1920 lda #50: jsr printer
 1930 \ disable printer
 1940 lda #12: jsr printer \ form feed
 1950 lda #3: jsr oswrch  \ printer off
 1960 rts
 1970 \ s/r to drive printer
 1980 :
 1990 .printer
 2000 pha \ save data
 2010 lda #1: jsr oswrch \ to printer on
ly
 2020 pla: jsr oswrch \ print data
 2030 rts
 2040 :
 2050 .pattern
 2060 EQUD &7D5B2500
 2070 EQUD &FFFA5805
 2080 ]
 2090 NEXT: PRINT"P% = "~P%
 2100 END
```

Machine Code Corner

This month Mr Toad finds out how many beans make five.

Last time, I mentioned that I'd been corresponding with one James Sugden of Cleckheaton in West Yorkshire. He actually uses a Beeb - well, an Electron, actually - to control his central heating. Remember a few years back, when computer enthusiasts were asked just *how* the advent of the home micro was going to revolutionise all our lives, the answer was always: "Well, you can use one to control the central heating, and then there's the... well... central heating..." It was always the ruddy central heating - and a good idea, too, given the cost of those ghastly motorised time-clocks and the second-hand price of the eminently suitable BBC B. But, have you ever met anyone who has actually done it? I never did until now. What's more, he uses the same computer as a telephone directory. More power to yer elbow, mate. Mr Sugden also sent me a program which generates those complicated headers for sideways ROMs, prompting you for details as it goes; myself I'd rather the budding programmer read up the subject - perhaps from my articles of last year - and then did it for himself. That's how you learn, I say, but full marks to Mr S for some very nice software, regardless.

This month's subject is to be machine-code arithmetic for the novice. Right. Well, it could always control the central heating... Oops! I was nodding off there. The 6502, unlike some other chips, can only add and subtract; multiplication is achieved mainly by adding, and most kinds of division are beyond the scope of this article, as is the floating-point arithmetic needed to deal with decimals - here we're dealing only with simple integer stuff.

There is only one form of 'add' on the 6502, that's ADC, the C meaning that the carry-flag is also added in, whether you like it or not - if the carry-flag is set, the answer comes out as one greater than the sum of the two numbers to be added. Some other chips give you a choice here. To begin with, one of the numbers must be in the accumulator, and the answer always ends up there. Where's the second number? Well: ADC #4 means 'add the number four (and the carry flag) to the number presently in the accumulator', whereas ADC 4 means 'add the number contained in memory address 4...' Watch that one, it's a prime source of bugs. Most addressing modes can be used with ADC: f'rinstance ADC (&80) means 'add the number *contained* in the memory location of which the address is stored in page zero locations &80 and &81...' Master-only, that one, but you'll have to look in the manual for all the other modes, which warrant a long article or none at all. (Oh, no. Please! Ed.)

What if the answer ends up bigger than &FF/255, and thus won't fit into the accumulator? Well, in that case the carry flag is set, which makes it possible to add multi-byte numbers with the carry flag acting in exactly the same way as the little figure you put below the tens column to remind you of a carry from the units column when you're adding numbers on paper. Note that since we can add only two numbers at a time, the carry can't be more than one; that applies in decimal, hex, binary or whatever.

Let's say you're writing a game, with scores that can mount up to more than 255, though you can't score more than 255 at one go. We'll reserve two bytes of memory to hold the scores, and give them labels: score=&80:scoreHi=&81. At the start of the game we'll set these locations to zero, of course. Now we'll write a subroutine to keep score; it will be called each time points are gained,

with the accumulator holding the number of points just gained:

```
.addToScore
CLC / clear carry flag at start, to avoid
      accidental errors
ADC score:STA score / lo-byte of score is
      updated and stored
LDA #0:ADC scoreHi:STA scoreHi / add carry
      flag & store result
RTS
```

The point of the last line is that ADC #0 means 'add one' if the carry flag is set, thus scoreHi will end up incremented by one if the first ADC instruction gave an answer greater than 255, causing the carry flag to be set. STA score and LDA #0 will not alter the carry flag. You might prefer this, which does the same trick:

```
.addToScore
ADC score:STA score / lo-byte of score is
      updated and stored
BCC return / skip next instruction if
      carry flag clear
INC scoreHi / in effect, add the carry
.return
RTS
```

Now d'you see why the add instruction involves the carry-flag? Mind you, if the 6502 had a simple ADD instruction as well as the ADC, you wouldn't always have to do a CLC 'just in case' before each addition (sometimes you know that the carry-flag will be clear anyway, but you'd better be quite sure!). Note also that the answer ends up in the ACCUMULATOR after the ADC, not in the variable 'score', so we did STA score (and, in the first version, STA scoreHi) to tuck the new total safely away.

Subtraction (SBC) works in a similar way, except that the carry flag works the other way about - you begin with an SEC to SET the carry flag, else the answer will end up one LESS than the difference between the two numbers. In subtraction we're concerned with a possible BORROW, rather than a carry, and since borrowing is the opposite of carrying, the flag works t'other way about. Think about it - I found it hard to grasp at first, but it's logical. Since SEC subtracts from the accumulator, we need another variable - call it .hold - to hold the points to be deducted (assuming, again, that the subroutine is called with A holding that number of points); we then put the lo-byte of the existing score into A and subtract from it the number we've just stored:

```
.deductFromScore
STA hold / that's the points to be taken
      off
SEC / set carry flag at start, to avoid
      accidental error
LDA score:SBC hold:STA score / subtract
      from score & store new value
```

then either:

```
LDA scoreHi:SBC #0:STA scoreHi:RTS
```

or:

```
BCS return / skip the next instruction if
c.flag SET (= no borrow)
DEC scoreHi / knock one off hi-byte
.return
RTS
```

This month's listing is of the second version of this routine; you can easily alter it to test any of the code so far. The Basic 'shell' will facilitate testing the code. N.B. I imagine you'll input in decimal, so I've written Hex readouts so you can see whether the lo-byte of score < decrement (thus involving a carry), or not. Use a score over 512 and a decrement less than 256; I've not bothered with any mugtrapping.

To multiply, you generally put one of the numbers into a location which we'll call .counter, the other into a variable like .hold, then go round the loop, adding the number in .hold to the value held in the lo-byte of a pair of locations like .score/scoreHi, then decrementing .counter and branching back to the loop if .counter isn't empty. I'm not going to hand you that one on a plate, you can do it if you try! Tip: be

careful about whether that branch should be BNE or BPL, or you'll end up multiplying by one more than you meant to.

If you want to multiply (or divide) by 2, 4, 8 etc., the quicker way is to use a rotate instruction, either ROL/ROR or ASL/LSR. We shall be considering those instructions next time, together with the AND, ORA and EOR operations, but here's an example for now: the well-known routine for printing a byte out in hex. The relevant bit is where we print out the hi-nybble (bits 4 to 7) first; to do this we have to divide their value by 16 before the convert-to-ASCII part. This division is done by four rotate-right instructions:

```
.printHex
PHA LSR A:LSR A:LSR A:LSR A / hi-nybble is
      now divided by 16
JSR printNyb PLA
.printNyb
AND #&0F / mask off top 4 bits
SED / set decimal mode
CMP #&0A / carry flag will be set if value
     >9
ADC #ASC "0"/ move into printable range
CLD / clear decimal flag
JSR oswrch /print nybble
RTS
```

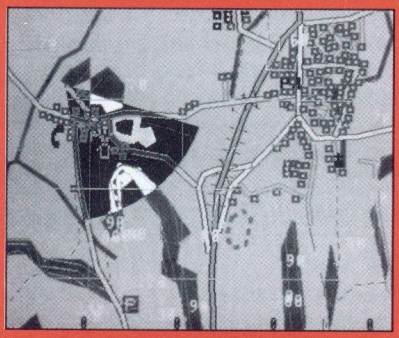
The reason for the SED is that if the decimal flag is set, 9+1=16! Well, that's nonsense, but 00001001 + 00000001 = 00010000. When you add one to 9, in decimal mode, and also add ASC "0", you get ASC "A", which is what you want for Hex. Again, if you think about it for a while, the penny will suddenly drop. I'm afraid the only good treatment of decimal mode - at least, that I know of - is in Rodnay Zaks, 'Programming the 6502', publ. Sybex. This is not Beeb-specific, but it's worth having if you see one, as Mr T did, at a car shoe sale (like a car boot sale, but not so working-class, y'know).

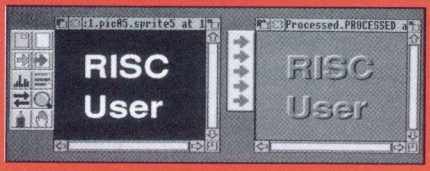
But I digress. (What? Really? Ed.) Also in next month's Frog Forum, we look at alternative electronics - curing hardware faults by counselling, aromatherapy and Ramupuncture. Until then, gentle reader, keep your griblets warm and whorple your hamster daily.

```
  10 REM Program Subtract
  20 REM Version B 1.0
  30 REM Author  Mr Toad
  40 REM BEEBUG  Aug/Sep 1993
  50 REM Program subject to copyright
  60 :
 100 PROCass
 110 :
 120 INPUT''''"Score so far "sc%
 130 ?score=sc% MOD &100
 140 ?scoreHi=sc% DIV &100
 150 PRINT'"NOTE: score now holds &";~?
score;" / ";?score
 160 PRINT"      scoreHi   holds &";~?s
coreHi;" / ";?scoreHi
 170 INPUT'"decrement by "dec%;
 180 PRINT;"decrement in Hex is &";~dec
%
 190 A%=dec%:CALL code
 200 ans%=&100*?scoreHi+?score
 210 PRINT'"Answer = ";ans%;" / &";~ans
%
 220 GOTO 120:And why not, indeed?
 230 :
 240 DEF PROCass
 250 FOR opt%=0 TO 2 STEP 2
 260 P%=&DE00
 270 [ OPT opt%
 280 .code
 290 STA hold
 300 SEC
 310 LDA score:SBC hold:STA score
 320 BCS return
 330 DEC scoreHi
 340 .return
 350 RTS
 360 :
 370 .hold BRK
 380 .score BRK
 390 .scoreHi
 400 ] NEXT:ENDPROC
```

RISC user

The number one subscription magazine for the Archimedes





RISC User, probably the most popular subscription magazine for the Archimedes, offers all the information you need as an Archimedes user. In every issue of RISC User you will find a wealth of articles and programs with professionally written reviews, lively news, help and advice for beginners and experienced users, and items of home entertainment.

The B5 size of RISC User allows a sophisticated design, big colour illustrations and pages full of information, and yet is still a convenient size to assemble into an easy-to-use reference library. Altogether, in its six years of existence, RISC User has established a reputation for a professional magazine with accurate, objective and informed articles of real practical use to all users of Acorn's range of RISC computers.

Contents of the latest Vol.6 Issue 9 of RISC User:

VIRUS UPDATE
A report on the viral threat to the Archimedes.

ACTION REPLAY
Irlam's Replay DIY kit brings movie-making within reach of the average user - RISC User explains how.

ALL ABOUT FONTS
Acorn's powerful outline font system demystified.

ACORN'S MULTIMEDIA EXPANSION UNIT
A look at Acorn's multimedia unit, which comprises of a fast CD-ROM drive, an audio preamplifier and mixer, neatly cased in a A4000 size box.

TOPOGRAPHER
A review of Clare's package for drawing and editing of maps, which can also create a 3D view from a map.

WIMP TOPICS
A major series aimed at readers interested in Wimp programs and Wimp programming. Each article looks at aspects of a particular topic.

WORKING WITH IMAGES
A series of articles discussing processing monochrome and colour images on the Arc.

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

INTO THE ARC
A regular series for beginners.

TECHNICAL QUERIES
A column which answers your technical queries.

THE DOS SURVIVAL GUIDE
A series of articles on how to use the PC Emulator.

EDUCATIONAL SOFTWARE AT HOME
A series of articles aimed at parents who want to use the Archimedes to help their children in their school work.

SUBSCRIPTION DETAILS

As a member of BEEBUG you may subscribe to RISC User for the reduced rate of £18.40 (a saving of £1.50 on the normal subscription rate). Overseas subscription rates are as follows: £27.50 Europe and Eire, £33.50 Middle East, £36.50 Americas & Africa, £39.50 Elsewhere

Validating Get Function

David Reed

This short routine checks key-presses for one of two letters in either upper or lower case.

```
100   DEF FNkey(K$)
110   LOCAL K%
120   REPEAT
130   K%=INSTR("@"+K$,GET$) DIV 2
140   UNTIL K%
150   =K%-2
```

For example: FNkey("YyNn") will return TRUE for 'Y' or 'y', and FALSE for 'N' or 'n', ignoring any other key-presses.

Galilean Satellites of Jupiter

L Theeses

Eric Hunt's program on the *Galilean Satellites of Jupiter* (11:6) has a slight problem in that it does not reject invalid date entries, and can therefore produce satellite displays for dates such as the 31st June or even the 31st February. This is therefore an excellent opportunity to implement the *Date Checking Function* by David Abbot (Hints and Tips 12:1).

The modification is very simple, and only involves adding the following two lines to the *Jupiter* program:

```
 1552 date=FNvalidate(DY,MN,YR):IF date=0
THEN 1550
 1585 DEFFNvalidate(D,M,Y)=D*M*(M<13)* (D<31
+((M+(M>7))AND1)+(M=2)*(2+((Y AND 3)=0)))
```

The modifications to produce the moving satellite display, suggested by Stephen Rose (Hints and Tips 11:8), also require a couple of changes, otherwise the assumption that every month has 31 days eventually causes a lack of correspondence between the displays and the dates shown on screen. The following changes to the *Jupiter* program will solve this problem:

```
 133   IF XD>=24 XD=XD-24:CDY=1
 134   IF CDY>28 AND MN=2 AND YR MOD4>0
THEN MN=3:CDY=1
 135   IF CDY>29 AND MN=2 AND YR MOD4=0
THEN MN=3:CDY=1
 136   IF CDY>30 AND (MN=4 OR MN=6 OR MN=9 OR
MN=11) THEN MN=MN+1:CDY=1
 137   IF CDY>31 MN=MN+1:CDY=1
 138   IF MN>12 MN=1:YR=YR+1

 1870 Delete
 1880 Delete everything after the colon
 1890 Change INT(DY) to CDY
```

If you want to be able to pause the program between each display, add:

```
, 1945 G$=GET$
```

Updating the Downloader

Francis Beach

Here is a useful update to the *Move Down Routine* by Alan Wrigley (Hints and Tips 8:8). The problem with the original routine was that it did not allow for displaying instructions on screen while the downloading took place. The following routine allows you to display as many instructions as you can fit into lines 0 to 4. Lines 6 to 8 perform the downloading, delete the routine, and run the program. The program itself must start at line 10.

```
   0 CLS:PRINTTAB(0,4)"You can modify this
text in any way you like. You can add
control codes, tabs, etc."
   1 PRINT'CHR$(130)"You can also use this
line..."
   2 PRINT'CHR$(130)"and this one..."
   3 PRINT'CHR$(130)"and this one..."
   4 PRINT'CHR$(130)"and this one... but no
more. That's quite enough room for
instructions, etc."
   5 PRINT' 'CHR$(136)CHR$(134)"PRESS A KEY
TO DOWNLOAD AND START": A=GET:
PRINT'CHR$(136)CHR$(129)"PROGRAM NOW
DOWNLOADING."
   6 *KEY 0 *T.|M FOR A%=0 TO (TOP-PAGE)
STEP4: A%!&E00=A%!PAGE: NEXT|MPAGE=&E00|M
OLD|MRUN|M
   7 *KEY 1 DELETE 0,9|M*FX138,0,128|M
   8 *FX138,0,128
   9 END
```

Personal Ads

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

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

Turbo board (65C102 co-processor) for Master 128 complete with User Guide and support disc, including Tubelink's Advanced Basic ROM and manual and Hi-Basic ROM £45. Tel. Dorset 0929 551450 work/ 553670 eves.

BBC series 7 with DFS also 5.25" 80/40T disc drive, loads of discs, also 3.5" single sided 40T drive (bare - no case), books on Assemblers and Compilers, other BBC oriented books, all Micro Users ever published! no reasonable offer refused. Around £75 to £100 or maybe less! Tel. Pontypridd 0443 206771.

BBC series 3 with DFS c/w 5.25" 80/40T drive, loads of tape based games all original, ATPL Sideways ROM board, Manager ROM, Replay (8271) ROM c/w instructions, T2D ROM, Disc Doctor, Wordwise ROM all for only £75 plus p&p, will split, will negotiate. Tel. Pontypridd 0443 206771.

10Mb hard disc, suit Master 512 user, c/w XEBEC interface card and BBC controller card, in large full size 5.25" box, drive itself appears to be faulty, hence low price of around £80, no reasonable offer refused. Pontypridd 0443 206771.

Master 128, ROM board 4, Spellmaster, ADT, Master ROM (Beebug) dual disc drive bridge unit, one 5.25" 40/80T switchable, one 3.5" 80T drive, Morley teletext adaptor, 2xROM software: EPROM writer, EPROM Programmer (Technomatic) inc. mains power pack, Voltmace Delta 4b joystick, all manuals for above. Books: Master welcome guide, reference manual 1&2, Master Advanced reference manual (Watford), View a dabhand guide + disc, Disc programming techniques for BBC micro + disc, Assembly language Programming (Birmbaum), Creative Graphics (BBC), Acorn Printer Driver disc. Software; Masterfile II (BEEBUG), Apricote Personal Accounts, Apricote Account Book + Invoicing, Colossus Chess 4, Extra Extra, AMX Super Art + new mouse, Stop Press, 4Mation Holed Out + a few extra bits and pieces. Offers around £300 for the lot. Tel. Kent 0322 664761 eves.

Interbase, Turbo co-processor, 512 co-processor, Accodata serial printer buffer 512k. Tel. Coventry 0203 410047.

BBC Master 128, 5.25" disc drive, lots of discs with magazine software on, Viewstore, Pascal, various manuals, BEEBUG and BBC User magazines, sensible offers over £160 please. Tel. Blackpool 0253 892709.

Silicon Vision Solidcad and Superdump both for £50 and also 12 cassette games Termulator B+ and GXR B+ Office Mate and Office Master, very few books left Viglen ROM cartridge and Watford Zif socket and Master compact keyboard case all must go hence cash required, s.a.e. for details. **WANTED:** Acorn DFS manual and 5" utilities disc also copies of DOS plus 1.2 or exchange all above for bitstick package or Novacad package BBC versions. S.a.e. please to Mr C Game, 24 Grosvenor Close, Tiptree, Colchester, Essex CO5 0JN.

Technomatic 10Mb Winchester disc, Acorn 32-bit co-processor for model B with 1Mb memory, single and double precision BASIC, Fortran 77, Iso-Pascal, C, Lisp, all manuals, Microvitec 1451 monitor. Offers. Tel. Cumbria 05395 32657.

Dual 5.25" disc drive £40, Wordwise, Teletext, Dumpout 3, Graphics, Speech, Starstick ROMs with relevant documentation, also books relevant to BBC. Tel. Luton 0582 454662 for details.

Master 128 £160, CC Interword £20, CC Spellmaster £30, CC Interbase £30, Watford Electronics dual disc drive £85, all in perfect order with original packaging, manuals etc. Offers considered. Tel. Blackpool 0772 635346.

WANTED: ADFS ROM written by Slogger Computers for the Challenger 3 system, also PMS Publisher DTP package. Tel. Lancashire 0253 712395 eves.

Pineapple diagram II CAD package £15, Floppywise Plus £10, Advanced Control Panel £10, Hyperdriver £10, Sleuth £5, Masterfile II ADFS £5, Clares Replica II £4, Clares Replica III £5, P.I.A.S. 14 £8, Vu-type £6, Time and Magik adventure game £5, Combat Lynx £3, Spycat £3, Steve Davis Snooker £3, Uridium £4. Tel. Lancashire 0253 712395 eves.

BBC Master ROMs: Sleuth £10, Exmon II £10, Toolkit + £10, Basic Editor £8, Watford Buffer £8, all with manuals, JVC colour monitor £60, Akhter dual disc drives in bridge unit £85, various software on disc and tape, books, mags etc. all must go! Tel. Gwynedd 0286 880997.

Master 512 + Turbo 2nd processor, 512 board with enhancement to 1Mb housed in Watford co-proadaptor, high definition colour monitor and mouse, resident ROMs include View, Viewsheet, Master ROM, Wordwise+, Spellmaster, also Akther 40/80 double disc drive in monitor stand with PSU, all well looked after and in perfect condition, all manuals £550. Many more ROMs and disc software and books for sale. Owner upgrading. Tel. Middlesex 081-902 8612.

BBC B with Wordwise and Viewstore, nearly new Cumana 40/80T single disc drive also a double disc drive 40/80T £125. Tel. Essex 0255 504956 anytime.

Master 512 with twin 40/80DD, ROMs include; View, Viewsheet, Basic 4, ADFS, DFS, Disc Doctor, Terminal and Edit, documentation for the above plus Master 512 guide, Ref manuals 1&2, Welcome Guide, 30hr Basic, Advanced disc user, also Philips green screen monitor 80, Acorn colour monitor, 2 yr collection of BEEBUG discs, BEEBUG mags Vol.1-12 all bound, and an assortment of books and discs, including Gem, Mini Trader, White Knight Chess. Asking price £275 no offers, buyer collects. Tel. Darlington 0325 483641 anytime.

BBC Master teletext adaptor £25, WE Video Digitiser £25, Care Smart cartridge £15, Morley AA ROMboard £20, Biogra[om £10, Illustrator and Chaufer with mouse £15, Hyperdrive ROM £8, ROM cartridge £5. Tel. Lincoln 0522 524977.

BBC B issue 7 with ATPL board many ROMs and RAM, Watford 5.25" dual drive, Panasonic 1080 printer, software, manuals & books, complete BEEBUG v.g.c to date £300 buyer collects (or may split). Tel. London 081-529 8479.

Marconi RB2 tracker ball £10, BEEBUG magazines 9/9 and 10/6 to 11/9 £5 (inclusive of postage). **WANTED:** Exmon II, Wordwise +, Toolkit +. Tel. Hampshire 0705 479082 after 5.30pm.

BBC B with 1770 DFS, Pace 40/80T disc drive, software, books including Advanced User manual, joystick etc. £140 o.n.o. Tel. Yorkshire 0274 576868.

Music Master with microphone interface, 5.25" disc, handbook, Mupados recorder tutor with Ensemble, Duet and Classroom Network packs (5 5.25" discs, handbooks and cassettes), Micro Maestro with 5.25" disc and 6 cassettes, all for £25 including postage (worth £179). Tel. Hampshire 0256 27018.

WANTED: Scrap print head for Panasonic KX-P1124 printer. Tel. 0462 682167.

BEEBUG 5.25" disc interface (Arc), 40T drive, DFS reader software and about 50 5.25" discs, the lot for only £40 o.n.o. + p&p. Tel. Derby 0332 557751.

WANTED: Zif socket for BBC B. Tel. Norwich 0603 416152.

Master 128, Wordwise Plus, Printmaster, View, Viewsheet, 2 Cumana 5.25" 40/80 disc drives, Datacorder tape unit, Electrohome 14" VDU, Star SG10 dmp, many books, BEEBUG mag from issue 1, software (eg. Fleet St Ed., Studio 8, Trivial Pursuit), Motorola MEK 6800D2 Evaluation unit for ic's plus manuals, CTL EPROM programmer plus manuals and various EPROMs. All at £300 but will consider separate offers. Tel. Kent 0322 226822 John eves.

BBC B+ (incl. Interword/Viewsheet), 1991 disc drive 40/80T, 1991 Acorn RGB colour monitor, Masterfile II, various extras and manuals £180 or p.o.a. Tel. Mr P O'Donnell Luton (0582 882502. B

THE FUTURE OF BEEBUG

Our mail bag has been swollen recently with correspondence regarding the planned demise of BEEBUG in April of next year, particularly since the publication of a letter from Arthur Adams in the July issue. We have extracted what seem to be worthwhile comments from a number of readers, and these are summarised below. Many other BEEBUG members also wrote in with similar thoughts.

Perhaps you would consider collecting together Hints & Tips and publishing in one book.

E.H.Ross

Over the years you have kept us updated on the Acorn range via BEEBUG. Why don't you incorporate BEEBUG items within RISC User magazine? It would keep we old ones informed and up to date, and you would be able to hold on to the BBC subscribers.

L.Faine

I realise that it (BEEBUG) may not be viable in its present form, and I appreciate your giving advance warning of your proposal to discontinue it, and the opportunity to comment. But it is when computer manufacturers and suppliers of software withdraw their support that a User Group becomes most important. We still need knowledge of sources of supply (new and secondhand), and of repair facilities as long as they exist.

John Adams

If you are convinced that the magazine must close, may I suggest that if readers cannot write to BEEBUG, then perhaps they could write to each other. Could you invite those readers who are interested to submit their names and addresses for publication in some form, so that current readers with problems would at least have some possible contacts?

Arthur Adams

On looking through a copy of RISC User, I came across the article by Gordon Gilmore *The PC Emulator Survival Guide*. The kind of information which he gives is what is most valuable in a magazine. Tell me, is there such a thing as a RISC Emulator, a DOS user group which is anywhere near as good as your user groups?

Thomas Carr

I speak from experience of two minority clubs. Both these clubs, with memberships in the low hundreds, produce bi-monthly magazines in A5 format, and provide information, spares, equipment and materials to members. Yet the annual subscription to both adds up to little more than half that for BEEBUG. So I see no reason why similar clubs should not be viable for Beeb users. Both these clubs are 30 years old, so why should not a Beeb User Group survive well into the next century?

Bernard Beeston

Might I be so bold as to suggest, through your auspicious pages, that you/we start the ball rolling with a suggestion that all interested parties meet in the Luton Hotel (next to the M1, J11). I appreciate that this venue will not suit everyone, but will give you/us a very good idea as to how many members are prepared to come to some arrangement to sort this out.

Tim Parsons

We look forward to further correspondence on the future of BEEBUG, and will address any reasonable and constructive ideas in the October issue of the magazine. B

BEEBUG MEMBERSHIP

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

RENEWAL RATES FOR BEEBUG MAGAZINE AND MAGAZINE DISC SUBSCRIPTIONS

See May 1993 Editorial for further explanation

The table below shows the renewal rate applying after the June issue 1993 according to the first issue of the renewal period. For joint BEEBUG/RISC User subscriptions add half the appropriate BEEBUG renewal rate to the full RISC User renewal rate; (UK £18.40, Europe & Eire £27.50, Middle East £33.50, Americas & Africa £36.50, Elsewhere £39.50).

Renewal Issue	Issues to go	Mag UK	Mag Europe	Mag Mid-E	Mag Am+Af	Mag Else	Disc UK	Disc O'Seas
Jun	9	16.56	24.75	30.15	32.85	35.55	45.00	50.40
Jul	8	14.72	22.00	26.80	29.20	31.60	40.00	44.80
A/S	7	12.88	19.25	23.45	25.55	27.65	35.00	39.20
Oct	6	11.04	16.50	20.10	21.90	23.70	30.00	33.60
Nov	5	9.20	13.75	16.75	18.25	19.75	25.00	28.00
Dec	4	7.36	11.00	13.40	14.60	15.80	20.00	22.40
J/F '94	3	5.52	8.25	10.05	10.95	11.85	15.00	16.80
Mar '94	2	3.68	5.50	6.70	7.30	7.90	10.00	11.20
Apr '94	1	1.84	2.75	3.35	3.65	3.95	5.00	5.60

BACK ISSUE PRICES (per issue)

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

Volume	Magazine	5"Disc	3.5"Disc
6	£1.00	£3.00	£3.00
7	£1.10	£3.50	£3.50
8	£1.30	£4.00	£4.00
9	£1.60	£4.00	£4.00
10	£1.60	£4.75	£4.75
11	£1.90	£4.75	£4.75

POST AND PACKING

Magazines and discs are postcode a. Please add the cost of p&p when ordering. When ordering several items use the highest price code, plus half the price of each subsequent code. UK maximum £8.

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

BEEBUG
117 Hatfield Road, St.Albans, Herts AL1 4JS
Tel. St.Albans (0727) 840303, FAX: (0727) 860263
Office hours: 9am-5pm Mon-Fri Showroom hours: 9am-5pm Monday to Saturday
(24hr Answerphone for Connect/Access/Visa orders and subscriptions)

BEEBUG MAGAZINE is produced by RISC Developments Ltd.

Editor: Mike Williams
Assistant Editor: Kristina Lucas
Editorial Assistance: Marshal Anderson
Production Assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Helen O'Sullivan
Managing Editor: Sheridan Williams

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, RISC Developments Limited.

CONTRIBUTING TO BEEBUG PROGRAMS AND ARTICLES

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

Please submit your contributions on disc in machine readable form using plain text format if possible for text, but please ensure an adequate written description is also included of your submission and the contents/format of your disc.

In all communication, please quote your membership number.

RISC Developments Ltd (c) 1993

Printed by Arlon Printers (0923) 268328 ISSN - 0263 - 7561

Magazine Disc

August/September 1993

CENSUS - This month's programs, in the second instalment of this three part series, provide extensions to analyse and list your collected records.

An electronic version of the traditional "Tower of Pisa" patience game.

VIEW PRINTER DRIVER - The ultimate printer driver for the View word processor giving access to many of the features available on modern printers that the original View printer driver is unable to cope with.

HEARING TEST - Use your computer to test your hearing with this audio utility.

A HALF SUBTRACTOR - Mr Toad's venture into machine code this month results in this machine code subtractor.

STAR DUMP - Two screen dump routines for the STAR LC24 printer to produce two tone large format print outs in all graphics modes. The multi mode dump routine is listed in the magazine and an alternative mode 0 routine is included on the disc.

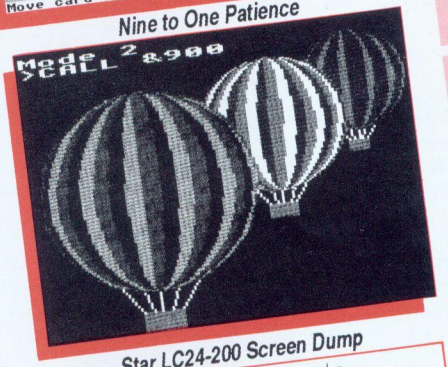
BASIC EDITOR - A compact and powerful full-screen Basic Editor which you can customise yourself.

LOAN REPAYMENT - A program to help you calculate the size of your repayments.

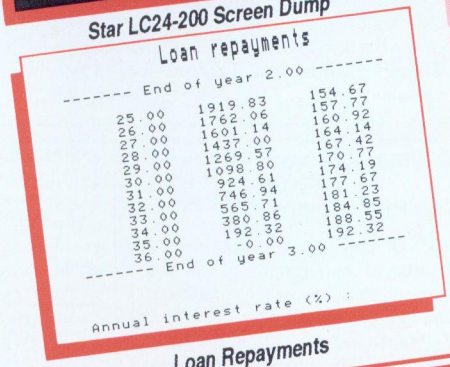
BEEBUG WORKSHOP - A demonstration of the look up tables discussed in this month's workshop, and a Polar curves plotting example making use of the routines described.

MAGSCAN DATA - Bibliography for this issue of BEEBUG (Vol.12 No.4).

Nine to One Patience



Star LC24-200 Screen Dump



Loan Repayments

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

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

FOR DISC (5.25" or 3.5") RENEWAL SUBSCRIPTION RATES (UK AND OVERSEAS) PLEASE SEE TABLE ON FACING PAGE

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

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

Special BEEBUG Members Offer

SPECIAL COMPUTER CONVERSION PACKAGE

As you may be aware Acorn has announced the demise of the BBC Master 128, additionally we have mentioned that BEEBUG magazine is unlikely to continue beyond its twelfth year (April 1994). As such it will become increasingly less cost effective to maintain Acorn's 8-bit computers, and the availability of new software and hardware will almost certainly be reduced to virtually zero. In recognition of this problem BEEBUG are urging you seriously to consider upgrading to one of Acorn's latest RISC computers. For many people this can be a daunting prospect. To help with converting systems we are making various offers aimed at simplifying the transfer of software from your BBC.
Our Special Offer falls into five parts.

FANTASTIC OFFERS ON NEW ACORN COMPUTERS

A3010

Code **0172g** ***RRP*** **£424.68**
Offer Price **£378** SAVE £46

Features: ARM 250 processor, mouse, stereo sound, 2Mb (unformatted) floppy drive, 1Mb RAM (upgradable to 4Mb RAM), mini-expansion slot, serial and parallel interfaces, TV modulator and The Family Solution software pack. Comes complete with EasiWord word processor and Quest for Gold game.

A3010 Learning Curve

Code **0173g** ***RRP*** **£680**
Offer Price **£606** SAVE £74

Features: In addition to the items in the Family Solution package, the Learning Curve includes 2Mb RAM, colour monitor, PC Emulator and the Genesis Collection (useful in education).

A4000 Home Office

Code **0209g** ***RRP*** **£999**
Offer Price **£890** SAVE £109

Features: Separate PC style keyboard, ARM 250 processor, 2Mb RAM (upgradable to 4Mb RAM), 80Mb hard drive, colour monitor, 2Mb unformatted floppy drive, serial and parallel interfaces and one mini-expansion slot and network interface slot. Software includes Easiwriter wordprocessor and Desktop Database.

A5000 4Mb HD162

Code **0206g** ***RRP*** **£1599**
Offer Price **£1409** SAVE £190

Features: ARM 3 processor, separate PC style keyboard, mouse, 4 slot backplane for expansion cards, 2Mb unformatted floppy drive, 4Mb RAM, 162Mb hard drive and multi-scan monitor.

A5000 Learning Curve

Code **0360g** ***RRP*** **£1445.96**
Offer Price **£1275** SAVE £170

As before but with only 2Mb RAM and 80Mb hard drive, together with a great collection of software including PC Emulator,

1st Word Plus wordprocessor, Acorn DTP, Genesis Plus and Pacmania Game.

These various conversion offers are designed to give you excellent value and a reasonably trouble free change to your new faster computer system. The new system will open up new horizons and give you a greater choice of an ever growing number of software packages.

All these offers are backed up by our telesales, technical support and computer repair services. If you require more information about any of the equipment you may need to transfer your system then reference to BEEBUG magazine issues Volume 8 numbers 9 and 10 will help or telephone us directly for technical advice.

FREE RISC USER SUBSCRIPTION OFFER

(when you purchase one of the above computers)

We will supply you with a subscription to RISC User magazine to run concurrently with your copies of BEEBUG until April 1994, whether or not your subscription ends before that date.

SERIAL LINK SOFTWARE TO EASE TRANSFER OF YOUR FILES

So as to cause you as little upheaval as possible when changing your system we are offering you Ivoryash's Serial Link and lead for only £25.

SOFTWARE OFFER

So that you may continue to use the software to which you are largely used, we are making a special offer on the following software packages to run under the BBC emulator on your new Acorn computer -

Wordwise £20, Interword £20, View £32, Masterfile II £5, Intersheet £25.

TRADE IN YOUR EXISTING EQUIPMENT FOR A NEW SYSTEM

You may prefer to keep your BBC and sell it privately (don't forget that members may place free ads in BEEBUG magazine). Should you wish us to take it in part exchange we are currently able to offer the following prices:

BBC Issue 7	£30	BBC Issue 7 with DFS	£70
Master 128	£100	Microvitec monitor	£40

Trade in offers can be applied up to two months after purchase of your new equipment. Trade in prices INCLUDE VAT, all others are ex. VAT.

BEEBUG 117 Hatfield Road, St Albans, Herts AL1 4JS
Tel: 0727 840303 (24 hours) Fax: 0727 860263 All prices are ex VAT and p&p