

C Wood

Issue 4 Aug/Sept '82

IN THIS ISSUE:

- Programmers corner
- Soft review
- Manual Review
- Spiral Patterns

..... and lots more!

2 editorial – important please read

At last we have actually made issue 4. This means that we have been going for $\frac{1}{3}$ year. The main question is will we make it to our first birthday. The answer to this is most definitely yes—we will continue producing LASERBUG for as long as there is a demand (which judging by the response will be several years at least). The correspondence we receive still boils down to the same thing—despite the delays, when you actually receive the magazine you are pleased with it which is the main thing we try to achieve.

And so onto the delays. Firstly let me apologise for not being able to reply to most of the enquiries you have sent—most of you did not include an SAE. Unfortunately the costs of producing a magazine are such that the price of sending a letter to every member are simply too high.

Enough of that, the main thing you want to know is why have there been all these delays. As I said last month to some extent this is your fault. LASERBUG is a USER GROUP and provides a base for information exchange. Exchange generally means you telling us what you have found out so we can tell the other members—not us telling you what we can find out. 2,000 members together can find out much more than the half-dozen of us on the LASERBUG team.

Obviously there are many other reasons. For a start we have had some staffing problems and for a second our mailing address had a slight breakdown. This is the reason that many of you said you had to write more than once—a couple of you even wrote 4 times! Another reason was that until now LASERBUG was run and co-ordinated in our spare time, but we have now all decided how to improve the situation.

For a start LASERBUG is now being run full time. This should improve the quality of the magazine and allow the mail to be dealt with more promptly. Secondly we have taken steps to ensure that the letters get through to us quicker. Hopefully this should ensure that overall LASERBUG runs much more efficiently.

Many have written both to us and other people saying that this is not the way you expect a user group to be run and that the situation over the past few months has been quite unacceptable. I am afraid the only consolation I can offer you is that I agree entirely. After reading that last sentence most of you will put me down as a complete liar but it is very hard to convince you of my sincerity when you are simply reading what I write. Hopefully you will meet, or have met, me at one of the various computer exhibitions we attend so leave your judgements until then. All I can offer is that if you are so disillusioned with LASERBUG that you do not think it is worth subscribing to, write to us at the usual address marking the envelope REFUND and we will send you the remainder of your subscription back. I can assure you that LASERBUG is carrying on and will do for a long time to come.

There is one thing I will ask every member to do—if you know of anyone who subscribed to LASERBUG over a month ago and has not received this issue, get them to write to us marking the envelope QUERY. It is taking us a little while to catch up with the backlog of mail but this should be cleared before long.

Before I end the topic of the delays I would like to answer a definite NO to David Ware's question—“...are you trying to emulate the BBC?”

Have patience and I will personally see that LASERBUG continues as it should do. OK, now all you people who do not normally read Editorials can find something else to read, the remainder carry on...

Now all those people who have gone off and read something else would have missed an important piece of information. However once they notice the cover date they are bound to read through the rest of my little article—that'll teach them not to read LASERBUG from cover to cover! If you look at the cover date on the magazine it says AUGUST/SEPTEMBER. It is regrettable that we have to do this again but it is the only way we can get the cover dates to tie up with the months. In actual fact, issue 3 should have been June/July/August but this would have been too much to bear at one time. Remember that your subscription will last you for twelve issues of the magazine which would normally be 1 year. All subscriptions will be altered on our files accordingly.

Now that all the unpleasant business has been taken care of, lets continue in the style you are more used to.

What developments have happened since the last issue? Well, BBC

production details have been given in full, Cleartone have been rescued from bankruptcy, the Torch has come onto the market, a cheap printer suitable for the BBC Micro has appeared and at last we have reports that people are starting to receive computers with the new operating system. And of course the new manual has arrived.

First onto production details. Before I start let me make it quite clear that I am not going to print month after month endless details of waiting lists, etc.—I will leave that up to the magazines who enjoy agonising and complaining about such things. I will however tell you when the peripherals are eventually ready as appropriate. All the following details were given on REM, the teletext newsletter for the BBC Micro (CEEFAX, page 705). I will repeat them here for those of you who have either no access to a teletext set or cannot be bothered to dial the page concerned.

Up to the 23rd July, 14,836 Model A's and 10,865 Model B's had been sent out. This means that there are now well over 25,000 BBC Micro owners out there (why don't more of you read LASERBUG?) If you ordered a model A, you should now receive it within 28 days. For model B purchasers, if you bought your computer before June you should have received it by now. For those who bought their computer before August, their Micro should arrive sometime in September.

If you bought a model B with disk interface before May you should have received it by now. Those purchased before September should arrive this month.

The Econet interface and computers fitted with it should start being dispatched towards the end of September. If you are content with a single disk-drive (100K) these have a 4 week delivery period in line with the interfaces. If you want to wait for the dual disk-drives (800K) these went into production last month and should be available soon.

If the second processor is what you're waiting for then you will still have quite a wait. the 6502 and Z80 versions go into production in November with the 32-bit processor being postponed 'till February '83.

The teletext adaptor begins production in October and the Prestel receives must wait until early '83. If you want something simple like a monitor or tape recorder then you have a four-week waiting period.

Got over those shocks? Good. There have been many reports in various places that Cleartone have gone into liquidation. They have in fact been rescued from the receivers by AB Electronic Products. They will continue assembling BBC Micros as the 120 strong workforce will be keeping their jobs. There will however be management changes at Cleartone (sounds ominous!)

At the Computer Fair back in April you may have seen details of a disk based business computer known as The Torch. What you may not have known is that it uses BBC Basic. In fact, when it is first switched on you can see the normal BBC start-up messages briefly. Any details you have on this would be appreciated.

Considering the number of letters we have received about it, no doubt you would have seen the articles/advertisements for a new printer called the Amber 2400. This is a 24-column dot-matrix printer. Unlike the Sinclair printer, it uses cheap widely available paper and has a crisp display. Amber have kindly agreed to supply us with a model to review. This will appear next month but for the time being see the What Printer? article in this issue. By the way, it is readily compatible with the BBC Micro and costs £80 + lead.

Reports are starting to trickle in to us that people are starting to receive the new versions of the operating system. Once we can get our hands on one we will do a review for you.

Of course the big news this month is that finally the new manual has arrived. Elsewhere in the magazine is a review of it. This is only our opinion on this welcome addition so we would be pleased to hear your opinions. If you would like to comment on it, please write to us marking the envelope MANUAL.

Well, that's it from me this month. Please have patience with us and issue 5, the October edition should actually come out in October. Remember there is nothing to force you to resubscribe after a year and obviously you will not do this unless the service you get improves. I plan to see that you all resubscribe *and* the service improves.

Please address all correspondence to:

LASERBUG,
4 Station Bridge,
Woodgrange Road,
Forest Gate
London
E7 ONF.

It helps considerably when sorting the mail if, in the top left hand corner you can write one word that describes what the letter is about i.e. PROGRAMS, MEMBERSHIPS, QUERIES, etc.. We regret that if you require a reply to your letter, you must include an SAE.

Enjoy issue 4...

Paul Barbour

Here is the spot which hopefully won't need to be regular and describes all the errors made in LASERBUG to date.

ISSUE 1

For all those of you who thought you had Labyrinth correct, J. J. Woolsey of Great Gransden, Beds., informs us otherwise. The new corrections are:

- Labyrinth 1 — Line 26 should be 260
Line 706 should have a space after the THEN
- Labyrinth 2 — Line 1295 should have 1365 in it, not 1360
Line 4880 should read VDU31,23,8,228
Line 4890 should read VDU31,23,10,227

Last call for any more corrections to the now infamous Labyrinth.

oddspot

All the *Oddspots* in the previous issues have used the graphics modes. This month we publish a program from Gary Brett of Enfield, Middlesex, that works in MODE7.

```
10 *KEY10OLDMCLSMLISTM
20 ONERRORRESTORE:GOTO50
30 MODE7:VDU23;8202;0;0;0;
40 VDU144+RND(7):L%=1
50 FORL%=L%TO39
60 READX%:VDUX%
70 NEXT:GOTO40
80 DATA184,166,163,169,228,240
```

For a different pattern try changing the DATA at line 80 (use the Teletext chapter in the User Guide for reference—p.150).

If you write a short program (10 lines or less as a rough guide) that performs something "impressive" either using sound or graphics, please send it to us marking the envelope ODDSPOT. A free 3 months subscription to LASERBUG goes to any we publish (can you think of a better prize!)

By the way, if you are wearing your finger down to the bone pressing the escape key with little effect, try pressing BREAK and take a look at line 10.

contents

● Editorial	2
● Oddspot	3
● Corrections	3
● Hardspot	4
● Pixel power	4
● Programmers corner	5
● Softreview	6
● Moving things	7
● Manual review	8
● What printer?	9
● Spiral patterns	10
● Bazooka	11
● Four in a row	13
● Bookreview	15
● Competition	16
● Contacts	16
● Next month	17
● Stop press	17

ISSUE 2

On page 3 (User Definable Keys) it wasn't explained properly that the double bar symbol stands for CTRL-CTRL-M performs the same function as RETURN. On page 4 of the same article it was stated that the cursor keys and copy key can be redefined like the 10 red ones. This is only possible with OS 1.0 and is operated with *FX4,2 (see p. 423, new User Guide).

On page 5, the top row of the diagram goes b₇ b₆ b₄. Obviously b₄ should be replaced by b₅.

On page 11 there is an ad for Epsilon Software. This company has now been taken over by Software for All whose ad appears on the back cover of this issue.

ISSUE 3

On page 6 Trevor states that LASERBUG stands for London & SE region BBC User Group. This was our original title but as we have members as far away as The Netherlands and Spain, this seems a little silly. We have now de-lasered LASERBUG. Although we will keep our original name, our "title" will be The Independent National BBC Microcomputer User Group.

On page 10 I said that to move the screen down one line you should use *TV255,0. As the second variable alters the interlace, if you do not want to alter this *TV255 would be sufficient. A couple of paragraphs below this I said that my 3 procedures gave the exact equivalent of PEEK and POKE. What I should have said is that they provided the same functions as PEEK and POKE implemented on other micro's with regards to the graphics (memory mapped display).

On page 17 is my slightly controversial article on the Spectrum. Since this was written Mr. Sinclair has changed his advertising leaflet slightly. However, the majority of what I said still stands and my opinions are still the same.

On page 19 it was stated that the version of SOUND with &HSFC had not appeared elsewhere. Since this article was written many people have featured this but the Cryer's book was still the first place to have details of both this command and ENVELOPE. On the same page was an article called "not a bookreview". Here you might have got the impression that the Owl was given free with every copy of Computer & Video Games. The first issue appeared in the June edition, the second in the September one and a third is promised in December. The final two sentences in the 6th paragraph are slightly confusing. I meant to say that the Beebon's pages, like those of Beebugs, are reproduced from a computer printout.

As a final comment on the three magazines I reviewed here, after seeing further copies of all of them I feel that both the Beebon and the Owl has deteriorated in quality. Beebug has maintained its standard although still very hard to read and of course LASERBUG has improved no end.

That is about all the errors I managed to spot—I am sure you will tell me different!

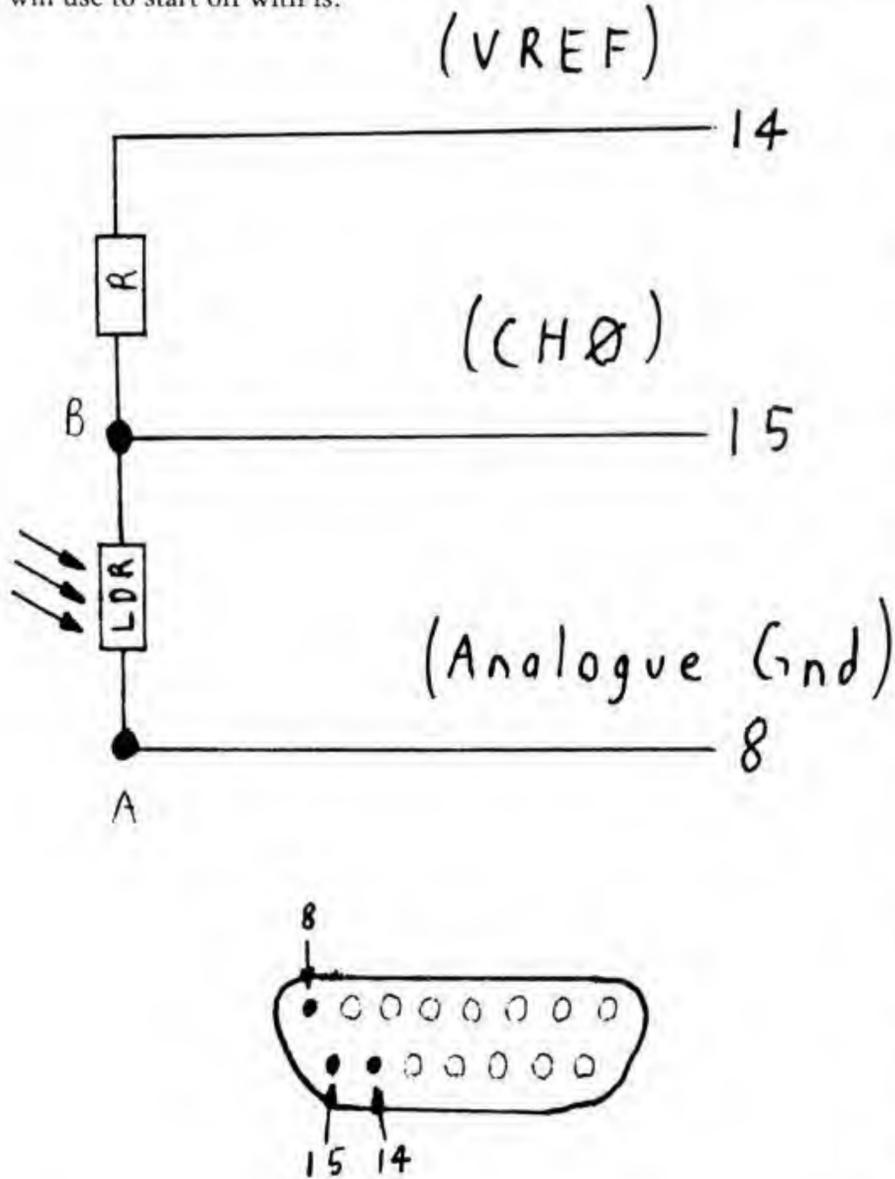
4 hardspot

If you are one of the so called "electronic wizards" then I am afraid this month's *Hardspot* is not for you. However, if you are a complete novice then continue reading.

In this article I am going to describe a few uses the Analogue-Digital convertors can be put to. All you will need is a small electronics kit, the kind you buy your children at Christmas. The one I am personally using for the purpose of *Hardspot* is a Radionics 70.

If you look on page 499 (External connections) of the User Guide (the new one) you will see details of the ports at the back of your computer. You are interested in the one marked analogue in. This diagram will be needed for reference throughout this article.

The first thing we must do is to build a very simple circuit. The one we will use to start off with is:



The two main components needed are a 100kΩ resistor (marked R) and a LDR (Light-Dependent Resistor). Another value resistor could be used but preferably one of higher resistance, not lower. There is a special way of telling the value of a resistor in the form of a colour code. The values are:

0 - Black	5 - Green
1 - Brown	6 - Blue
2 - Red	7 - Mauve
3 - Orange	8 - Grey
4 - Yellow	9 - White

There are two sections of bands on each one, ignore the silver/gold band. The first two bands indicate the initial digits, the third represents the number of zero's after it, i.e. if a resistor has the colour bands Brown/Black/Yellow this represents 10/4 × 0. This means a resistance of 100000Ω or 100kΩ.

The second component is a LDR. If your particular kit does not have this component then wait until later on in the article. LDR's are made of a light-sensitive semi-conducting material like cadmium sulphide. When

there is no light falling on it the resistance can be millions of ohms. When there is bright light its resistance will fall dramatically to only a few hundred.

In making the three connections to the A/D convertors I would not recommend that you simply stick bare wires into the sockets. Although there is no safety risk, you could damage the socket itself. What you should really do is purchase a special plug which is designed to fit into the socket. These can be purchased for £2.25 from Midwich Computing.

OK, now you should have the hardware set up. The next thing required is a program to read the A/D convertors. A simple program such as the one below would suffice to start off with:

```
10 PRINTADVAL(1)/16
20 GOTO 10
```

If when you run the program the values printed are greater than 4000 then it must be connected up wrongly. Otherwise in good daylight the value should be between 20 and 40 (this could vary depending on your particular LDR or light level). As you will see if you cover up the LDR, the values will increase. This could be a basis for a simple burglar alarm.

As opposed to numerical values on the screen, change line 10 to:

```
10 SOUND0,-15,3,1:SOUND1,0,ADVAL(1)/32,1
```

This will give a sound resembling a geiger counter. The value 32 might have to be changed depending on your particular LDR.

What if you haven't got a LDR? Well simply place any device which has a variable resistance between A and B on the circuit. If you place a diode in the circuit, this could be used for measuring temperatures. You could even connect a speaker/microphone into the system (a speaker and a microphone are essentially identical—the difference is whether the two wires are used for input or output). With the appropriate software you could have the basis for a simple speech recognition system.

I could carry on with this topic for many more pages. However, I will end there for this month and leave the experimenting up to you. Please write to us here at LASERBUG about your experiments and perhaps we will do a follow-up article. In the meantime if you have any tips to do with hardware or have built something for your computer yourself then write to the usual address, marking the envelope HARDSPOT.

Paul Barbour

pixel power!

Go to your cupboard and dig out that old leaflet that Acornsoft sent you or find one of their adverts. Look at a picture of their Pacman type game, Snapper. The actual Pacman or Snapper is in three colours even though it takes up only 1 character square. The method used for doing this is the subject of this article titled, for want of a better name, *Pixel Power*.

The first method that must come to mind for doing this is using PLOT69 which plots just one individual pixel. Good idea but this would take too long to plot an 8×8 grid (2/5 second was the best I managed in BASIC).

So how do you do it? Well, the answer is use VDU5 and 23. VDU5 joins both the text and graphic cursors and VDU23 defines the actual character.

OK, lets assume that you want to print a yellow circle inside a white square (though why you would want to I have no idea). Firstly, using VDU23, define (i) the outline shape and (ii) define the circle:

```
(i) VDU23,224,231,195,129,0,0,129,195,231
(ii) VDU23,225,24,60,126,255,126,60,24
```

Next we must find out how to position the square. You have probably been used to, when using any MODE, PRINTTAB(X,Y) when you need to

place a character on the screen. The same kind of method is used with VDU5. The difference is that instead of having a grid which can be as large as 80×32 to place your character, you can position it to the accuracy of a single pixel. To print say, the letter A in the centre of the screen, first enter VDU5 in one of the graphic MODES. Now enter MOVE640,512:PRINT"A". The MOVE command moves the joint cursor to the centre of the screen. The PRINT command prints the letter A on the screen with the bottom left-hand corner of the character being at position 640,512. VDU4 must be entered after this if you want to continue programming as when the VDU5 statement is executed, all scrolling stops—any text that would fall off the bottom of the screen wraps around back up to the top. Using this method you can position two characters on top of each other (see p.74/5, User Guide).

This demonstration ends, as all good demonstrations should with a program. There are two programs below, one is the full program if you wish to see how the program works, the second is if your typing finger is sore and can't be bothered to enter lines 80–220 in full.

```

10 MODE2:VDU23;8202;0;0;0;
20 VDU23,224,231,195,129,0,0,129,195,231:REM
   Background
30 VDU23,225,24,60,126,255,255,126,60,24:REM Main
   Circle
40 VDU23,226,0,24,60,126,126,60,24,0:REM Slightly Smaller
   Circle
50 VDU23,227,0,0,24,60,60,24,0,0:REM Small Circle
60 VDU23,228,0,0,0,24,60,60,24,0,0:REM Dot
70 VDU5
80 GCOL0,7:REM White
90 MOVE640,512:PRINTCHR$224
100 A=GET
110 GCOL0,3:REM Yellow
120 MOVE640,512:PRINTCHR$225
130 A=GET
140 GCOL0,2:REM Green
150 MOVE640,512:PRINTCHR$226
160 A=GET
170 GCOL0,4:REM Blue
180 MOVE640,512:PRINTCHR$227
190 A=GET
200 GCOL0,1:REM Red
210 MOVE640,512:PRINTCHR$228
220 VDU4

```

When you run this program you will first see the white background to a circle. Press a key and a yellow circle is printed in the centre. Press another key and a smaller green one appears, another and an even smaller blue one, once more and a red dot is put right in the centre. If you can't be bothered to type in the entire program, enter up to line 70 and enter:

```

80 FORCHAR=224TO228
90 READCOL:GCOL0,COL
100 MOVE640,512:PRINTCHR$CHAR
110 A=GET
120 NEXT
130 VDU4
140 DATA7,3,2,4,1

```

This version is much shorter and does exactly the same thing.

What use is this capability you might ask? The manual shows one use for it. If you've ever wondered how to use the underline character to underline things, here's your answer. Also it makes using multi-coloured graphics characters much easier.

For an example of it used in a complete program, see the "4-IN-A-ROW" program elsewhere in this issue.

Paul Barbour

The first *Programmers Corner* seems to have gone down well with most of you. Hence, due to its popularity, we decided (as was originally planned) to make it a regular feature.

On this page we are going to try to bring together all the little hints and tips we are sent or discover ourselves. However, all this will grind to a halt if you do not write to us so, if you find out something that would make all our lives easier then please write to us, at the usual address, marking the envelope P/C. Alternatively, if you have a programming problem write to us, because if you have a problem, the odds are that somebody else has the same problem.

Firstly, my sincere thanks must go to Dr D.E. Susans of Harrow, Middlesex for informing us of the following hints.

The first of Dr Susans hints are for all printer owners among you. For many printers, when you are using printouts other than those of the simplest kind, they require control characters to be sent to them. If you simply use PRINTCHR\$(—) statements with the printer switched on then these are also interpreted by the computer, often with disastrous results. This can be avoided by using VDU1, . . . methods. You should remember that all VDU statements, no matter of what type, can be run together with just one VDU prefix. For example, supposing you are using an Epson MX-82F/T printer and you wish to print 34 characters/line (which is the standard we use for the LASERBUG pages) in emphasised text. The control codes needed are ESC E (emphasised print), ESC Q 34 (34 print column width), CTRL-B (to switch the printer output on.) If you look on page 507 of the new User Guide, you will see the ESC command is VDU27 (do not confuse the ASCII command code ESC with the ESCAPE key which is sometimes abbreviated to ESC. They mean completely different things). Hence, the codes become converted to VDU1,27,1,69,1,27,1,81,1,34,2. Note that ASCII codes for E and Q (69 and 81) are used. In actual fact the ESC Q code was unnecessary as this could be replaced with WIDTH34 instead. However, remember this only affects things that the computer prints out and not what you enter.

In these hard times, many of you try to make up your own interconnecting leads for the computer ports and use insulation displacement connectors (IDC) and ribbon cable. However, it is quite common for people who are inexperienced at doing this to get an occasional bad connection and, as anyone who has done this will tell you, it is extremely difficult to repair this fault. Most of these bad connections can be avoided by warming the cable to about 60°C immediately before assembly. This softens the plastic and permits the correct entry of the metal contact fingers into the cable. The cable warming can be carried out either with hot water (but you must dry the cable before jointing) or by, say, 3 minutes at about 1" from a 60 watt lamp.

There appears to be a considerable amount of misunderstanding about *FX9 and 10 and unfortunately the new User Guide is not as clear as it could be. The colours 1 and 2 do not correspond to the COLOUR1 and COLOUR2 statements but to the two colours given in the list of available flashing colours (page 224, new User Guide) i.e. for COLOUR9—flashing red/cyan—*FX9 refers to the duration of the red phase and *FX10 to the duration of the cyan phase. All flashing colours change in unison.

The relay inside the computer is intended primarily to control the cassette tape motor. However, at other times it can be usefully employed for other applications. The relay contacts are opened by *FX137,0 (or *MOTOR0 or *M.0) and closed by *FX137,1 (or *MOTOR1 or *M.1). One such use for this is for morse enthusiasts who can use the relay to directly key the transmitter. It should be noted that this relay is a low-power, high-speed relay and is not suitable for currents in excess of about 100mA and 12V. For higher powers (such as those used for model train control) an external slave relay should be used. If more complex switching is required then the User I/O ports should be used instead (Reference: Interfacing the BBC Micro—simple digital input/output by P. Beverley, Electronics and Computing, July '82, p.44–46).

Mr S. Samson of Cheltenham, Glos., asks if there is any way of switching the Caps Lock/Shift Lock lights on and off under software control. In actual fact it is possible to both switch on the lights and their functions with a simple command. All you have to do is simply POKE the

appropriate value into location 216. The values are:

?216=	Caps Lock	Shift Lock
16	—	*
32	*	—
48	—	—
64	*	*

This is an extremely helpful feature which can help prevent many errors in programs and hence making them foolproof.

All of you are familiar with working in decimal in programs. However, many programmers need to work with hexadecimal as well. As you will find out if you experiment, there is no simple way to enter hex numbers—INPUTN will simply return 0 if either you use & or letters (A, B, C, D, E, F) in the input. If you want to enter hex straight into a program, say into the variable N, just use N=FNHEX after entering the following routine:

```
1000 DEFFNHEX INPUTHEX$
1010 IFHEX$="" THEN = - 1
1020 HEX$=" "&" + HEX$
1030 HEX = EVAL(HEX$)
1040 =HEX
```

Most of you are now familiar with VDU23;8202;0;0;0; for turning the cursor off (of VDU23;&200A;0;0;0;). The only way however that has been published to my knowledge for turning it back on again is by changing the screen mode. In actual fact, VDU23;29194;0;0;0; (or VDU23;&720A;0;0;0;) will turn it on again. There are much simpler versions available for the new operating system (see p.7, new user guide).

Just by chance one day, I left the CAPS LOCK off and tried to *CAT a program. In reality I did *cat and to my surprise it worked. After a little experimentation I found that any operating system command (i.e. one with a * preceding it) can be entered either in upper or lower case. Still with * commands, on pages 483/484 of the manual you will see the abbreviations that can be applied to normal commands. In general, all operating system commands can be abbreviated to one letter e.g. *CAT becomes *C.. You must be careful to distinguish between commands with the same initial letter i.e. *SAVE and *SPOOL.

Last month you will remember we published that amazing little routine that produced many more colours than those normally available. A small bit of exploring has determined that this is produced by over-ranging GCOL (although we poked colours directly onto the screen). Dig out from your cupboard the Welcome booklet and turn to page 9. Enter the little program and run it. OK, now changing line 10 to MODE2 and you get a more colourful version. However, by changing line 20 to GCOLRND(7),RND(7) you get the effects described last month. We have under development a way of producing those colours without the stripes.

Well, that's it for this month's *Programmers Corner*. There were several other hints that we were going to publish but these were made redundant by the new User Guide. However, just in case you missed any, try looking at the manual review elsewhere in this issue.

Paul Barbour

softreview

The programs we review this month are from the Romford based company *Software for All*. Many of you will be pleased to hear that not all of the programs they produce are games. Not all of the programs loaded first time, although we managed to get every one to load eventually. Instructions are either given in the program or on a separate sheet.

Software for All can be found at 72 North Street, Romford, Essex. Phone: 0708-752862.

We would like to thank *Software for All* for lending us their products.

GAME: Ghost Maze

SUPPLIER: Software for All

PRICE: £6.95

DESCRIPTION OF PROGRAM: Ghost Maze is Software for All's version of Pacman. Unfortunately it is a rather simple version. There are only two ghosts that get confused if you happen to stand still. The Pacman seems to have no eyes and keeps his mouth constantly open (like a certain other person I could mention). There is no fruit and the power pills last for an extremely short time. It is controlled using the cursor keys. The program is all right as a game but it is rather poor as an impersonation of Pacman.

PRESENTATION: ***

ADDICTIVE QUALITY: **

USE OF GRAPHICS: ***

LOADING: ***

VALUE FOR MONEY: **

—o0o—

GAME: Beeb Trek

SUPPLIER: Software for All

PRICE: £7.95

DESCRIPTION OF PROGRAM: Beeb Trek is a real time Star Trek game. It uses the function keys to enter the commands and kept me amused for many hours whilst playing it. It is loaded in two parts, the first part defining all the variables and the second part playing the game. This ensures that it will run on a Model A or B. Because of the way the second part is saved, it is impossible to make back-up copies. At a first glance, the instructions look complex, but a few games soon sort you out. Torpedoes are directed using angles and not the compass point method. The instructions say that you can return to the main computer by pressing ESCAPE. However, I found that if this was done at the wrong time it could cause the program to crash completely. Also occasionally the trace of a torpedo was left on the screen. However, I can honestly say it is the best and most enjoyable Star Trek game I have seen so far.

PRESENTATION: ***

COMPLEXITY: ****

USE OF GRAPHICS: ***

RESPONSE SPEED: ****

LOADING: **

VALUE FOR MONEY: ****

---o0o---

GAME: Cobra and Robo-Swamp

SUPPLIER: Software for All

PRICE: £6.95

DESCRIPTION OF PROGRAM: The first of these programs, Cobra, you might be more familiar with under the name of Snake. You control a snake around the screen. With this you must try and touch a special square worth a varying amount of points without doubling back on yourself or touching the wall. It is an extremely good game and would be worth buying even if it was on its own. As a hint, if you buy this program, try holding a key down to make the snake go faster. The one bad thing I will say about the program is that it has to be *RUN and not loaded. Two versions of Cobra are available, one for the Model A, one for the B. The game is identical in both cases, the only difference being the quality of the graphics. The second program on this tape is called Robo-munchers. You have to run around a swamp infested area and try to make the little munchers fall into one of the swamps. This game is also very good but is unlikely to be used as much as the first one. Both games use ZX/: as the control keys. The label on the front of the tape is slightly confusing. To load the first program enter *RUN, for the second try CH."ROBO-SWAMP".

	Cobra	Robo-Swamp
PRESENTATION :	***	***
ADDICTIVE QUALITY:	*****	***
USE OF GRAPHICS :	***	***
LOADING :	*****	*****
VALUE FOR MONEY :		****

---o0o---

GAME: JR
 SUPPLIER: Software for All
 PRICE: £6.95

DESCRIPTION OF PROGRAM: JR is a monopoly type game in which you and a friend take on control of oil companies. You really need a JR type personality to survive. Not only have you the problems of exploring and drilling for oil, you have an unsettled workforce who seem to spend half their time demanding pay rises. One of the first things the game asks is, "Have you got the CAPS LOCK on" — pity they didn't read this month's *Programmers Corner* before they wrote it. Also, during one part of the game, half a word ends up on the next line. This is a pity because it is so easy to ensure correct formatting. Despite this, JR is a game that should keep you amused for several hours.

PRESENTATION: ***
 USE OF GRAPHICS: ***
 LOADING: *****
 VALUE FOR MONEY: ****

---o0o---

PROGRAM: Database
 SUPPLIER: Software for All
 PRICE: £9.95

DESCRIPTION OF PROGRAM: Database, as the name suggests, is involved with storing large amounts of records (up to 300 for the Model B, was the quoted figure). The records can be kept in the computer's memory and loaded/saved on/from tape as needed. Standard manipulation techniques are used which includes FIND and ORDER commands. FIND will find you all files with a particular string in them. ORDER places the files in alphabetical order. The program is slightly complex to control and needs a lot of practice. It is not a particularly startling program and has poor formatting in several places. Files can be copied onto a printer. In my opinion this program is typical of its kind and is not worth the £10 charged.

PRESENTATION: **
 RESPONSE SPEED: ****
 VALUE FOR MONEY: **

---o0o---

PROGRAM: Character Generator
 SUPPLIER: Software for All
 PRICE: £4.95

DESCRIPTION OF PROGRAM: As soon as people found out that you could define characters on the BBC Micro, a whole stream of programs were published to help you do this. Many were sent to *Program of the Month*. However, none of them compare in any way to this. If you use user definable characters at all, I suggest you buy this program tomorrow. A 4x4 grid is drawn on the centre of the screen. Each square is then further divided up into an 8x8 grid representing one character. At one side of the screen is a picture of the characters before you redefined them (to proper scale) and on the other the character as it appears now. Briefly, you move a small cursor around the screen which can be in one of two modes—MOVE or INVERSE. There are several other functions available but I have not enough room here to describe them all. To you, this might seem the same as all the other versions—I suggest you get a look at this program in the flesh first. The only comment I would

make is that the program could have done with a *FX15,1 to flush the keyboard buffer. At £4.95 this is the cheapest program produced by this company and provides excellent value for money. Unfortunately it is available only for the Model B.

PRESENTATION: ****
 USE OF GRAPHICS: ***
 VALUE FOR MONEY: *****

---o0o---

To summarise, I would recommend Character Generator, Beeb Trek, Cobra and JR.

If there is any particular companies range of software you would like us to review, please write and tell us. Alternatively, if you buy some programs—whether they are good or bad—why not write your own *Softreview* in the style used above and send it to us.

Paul Barbour

moving things

This article is entitled *Moving Things*—a strange title for an article you might think. Well, you look at any game, most of them will involve moving an object around the screen either just left and right or in four directions or even 8 like a compass. It isn't only games either, much business software involves the moving of a cursor around the screen. That is what this article is about—moving things around the screen, whether they be Pacmen or a cursor.

OK, to start off with, enter the following program. It is written in a fairly straightforward and logical way and is typical of its kind.

```

10 MODE4
20 VDU23;8202;0;0;0;
30 *FX4,1
40 X=20:Y=16
50 PRINTTAB(X,Y);"*"
60 A=GET
70 PRINTTAB(X,Y);" "
80 IFA=136THENX=X-1
90 IFA=137THENX=X+1
100 IFA=138THENY=Y+1
110 IFA=139THENY=Y-1
120 IFX<0THENX=0
130 IFX>38THENX=38
140 IFY<0THENY=0
150 IFY>30THENY=30
160 GOTO50
    
```

Line 30 makes the cursor keys return ASCII values. (Reference p.81-83, User Guide—also gives details of GET and INKEY). When you ESCAPE from the program this must be reset with *FX4,0 otherwise you will not be able to edit the program. Lines 80-110 checks on the movement of the "thing" while lines 120-150 checks it is not out of range.

Now RUN the program and try moving the shape around with the cursor keys. This program is fine but it waits for the user to press a key before it does anything. That does not matter too much in this type of program, but when you are trying to move other objects as well—whether your "thing" moves or not—obviously, this will not work.

To get around this problem make the following changes:

```

60 A=INKEY(0)
69 IFA=-1THEN60
    
```

Line 60 will wait for 1/100 second for a key to be pressed while line 69 prevents flicker (try removing it). This altered routine is fine but the keyword in computing is efficiency—is this program as efficient as it could be? The answer, quite simply, is no. Use the program for a little while until you are completely familiar with it. Now make the alterations below and see if you can notice any difference when you run the program:

```
DELETE50,69
80 IFINKEY(-26)THENX=X-1
90 IFINKEY(-122)THENX=X+1
100 IFINKEY(-42)THENY=Y+1
110 IFINKEY(-58)THENY=Y-1
160 PRINTTAB(X,Y);"*"
170 GOTO70
```

Can you see any difference between the speed? I thought you might. You are probably used to using INKEY with positive numbers with the number specifying how long the computer should wait for an input. When you use negative values, the computer scans just one particular key depending on the number and returns either TRUE or FALSE (see page 100/101, User Guide). As you can see the scan is much much quicker. Before you ask, no, I didn't discover this feature myself—see page 273–275 of the User Guide (yet again!).

For most applications this is far too fast, so try changing the 1 at the end of lines 80–110 to 2. Again, this routine could still be more efficient but for the purposes of this article it is fine as it stands.

If the second version was suited to your purpose then any routines to move other objects should be placed between lines 60–69 (RENUMBER if necessary to give yourself more room).

If you require movement more than just up/down/left/right, then the final program would be the best one. Try pressing, say, the cursor up and cursor right keys together—you get diagonal movement. This is rather hard to control and needs a little practice but would be suitable for most purposes.

manual review

Well, by now a great deal of you would have received your new User Guide. If you haven't don't worry, the Beeb will get around to you eventually. By the time you are reading this you should have been able to get to grips with it fully and have drawn your own opinions of it—we thought we'd let you know ours. Also below we have picked out a few of the little things in the new manual that you might not have found.

The new manual is 518 pages long and, if you had to buy it separately, would cost £10. It is divided into three clear sections—(i) how to set up your computer when you first receive it which covers 3% of the book, (ii) a complete course on BASIC which covers 34% and (iii) a reference section which covers a massive 63%.

The first section on “Getting Going” is well written. It starts off by getting you to set up your micro on a TV (or monitor) and then gives you a variety of commands to enter which demonstrates both the computers graphics and sound. This is something I approve of as it shows the new owner of a BBC Computer that it can do quite impressive things from the start (remember that quite a few new owners had a computer such as the ZX81 before). Next it gets you to load in the welcome tape which again is a good introduction to the computer's capabilities. This section ends by giving you a “guided tour” around the keyboard.

The second section takes you through BBC BASIC, starting off with PRINT and ending with file handling. This section is written manificently and I have no hesitation in urging other computer manufacturers to write their manuals in a style such as this. It makes books such as Basic Programming/Let Your BBC Micro Teach You To Program completely redundant. Any beginner should have no trouble following this.

The third section, which is by far the largest, is a reference section. It contains details of most things you need to know when programming.

One disappointing thing is that the section on assembler isn't orientated more towards the beginner. Obviously it would have been impossible to outline this topic completely but a few little pointers wouldn't have gone amiss. We are promised sometime in the future a book from the Beeb on Assembler programming. At LASERBUG we are trying to arrange a series on this particular subject—more on that as and when I know more.

I did not go into too much detail about the manual because the majority of you will already have seen one. We would be pleased to hear your comments on the manual, whether they be good or bad. If you wish to write to us on this topic, please mark the envelope MANUAL.

The remainder of this review is simply a dissection of the manual. I will try to draw your attention to the little hints you can pick up from it but might have missed. Also I will point out any errors I have found.

- p. 13 The cassette leads are not labelled. The 7 pin DIN—5 pin DIN/2.5mm jack is part number ANG09, the lead terminating in 2 × 3.5mm and 1 2.5mm jacks is part ANG08. I have not yet seen a cassette recorder with a 7 pin DIN and so I think the Beeb have taken this lead off the market.
- p. 15 SHIFT-CTRL function is explained.
- p. 16/19 ROM socket is described.
- p. 18 The differences between MODE7 and the other screen modes are explained. Also here CTRL-N is mentioned but not how to turn it off (CTRL-O)
- p. 23 Details of *TV are given.
- p. 33 It states here that the only thing that will prevent OLD from working is if you have started to type in a new program. It will also not work if you have created any variables.
- p. 35 Here it states that the way to verify a program is using *CAT. This infers that there will not be a VERIFY command in the new operating system as certain other people have been campaigning.
- p. 51 Spaces can make up a program line by themselves as we said last month.
- p. 55 I couldn't help mentioning the way that the ESCAPE key is referred to as the Panic Button—perhaps the BBC should label it so?
- p. 57/59 It is explained how to operate the windows—you might have found this section rather confusing in Prov. manual.
- p. 75 The way that “foreign” letters can be displayed is shown using VDU5 (see also Pixel Power elsewhere in this issue).
- p. 86+ Note that here, as in the rest of the book, structured programming is encouraged.
- p.100/101 True and False with their use in IF...THEN statements is described. This again people might have found confusing before.
- p.142 The way the BREAK key will operate in the new OS is outlined fully here—CTRL-BREAK will be the equivalent of a “double BREAK” whereas SHIFT-BREAK will provide an auto-boot.
- p.153/154 Details of SHIFT-fn keys for generating teletext control codes are given. In general, this system should not be used for programs that will be published in LASERBUG as these codes will not be picked up by a printer.
- p.194/195 Two pages are given on ways to speed up your programs and save memory. Although most of these methods are elementary, they will be of particular use to Model A owners.
- p.198 I thought it was rather alarming to see a heading such as “known problems” under two of the keyboard descriptions (p.200/281). As references are made throughout to the 1.0 OS, I can only assume that these are present in all versions of the operating system!

- p.203/204 Note the use of ADVAL(-n) to return memory spaces in various buffers.
- p.237 Although those used to writing in machine code are probably used to doing this, DIM X 24 (i.e. a DIM statement with no brackets) is used for reserving memory space (similar to CLEAR on certain computers).
- p.245 To those who enjoy experimenting with sound effects and find only 4 envelopes a handicap, as long as you are not using the cassette buffer you can define 16 envelopes.
- p.273/275 For a much more efficient keyboard scan, use INKEY(-n). See also "Moving Things" article elsewhere in this issue.
- p.290 The explanation of LISTO 1 think should have been listed out in binary form (i.e. page 422). I have done this below for you:

	Space for case a	Space for case b	Space for case c
LISTO0	no	no	no
LISTO1	yes	no	no
LISTO2	no	yes	no
LISTO3	yes	yes	no
LISTO4	no	no	yes
LISTO5	yes	no	yes
LISTO6	no	yes	yes
LISTO7	yes	yes	yes

(Before we start getting letters I do know that I should have done it the other way round for a binary count.)

- p.314 Note the difference between OPT and *OPT—these are completely different and easy to mix up.
- p.342 If you are used to a RANDOMISE command on other systems, RND(-n) performs the same function on the BBC Micro.
- p.400 I think it is nice to see both DISC and DISK available—this is rather like the difference between scone (as in bone) and score (as in con).
- p.432 It states here that OSBYTE with the accumulator set to 135 (&87) is not available on the 0.1 OS. As you will know from the last LASERBUG issue (Programmers Corner) you can use this call.
- p.482 This does not outline all of the error messages. I have managed to get several others i.e. RENUMBER space.
- p.514 The "D" index has a great deal missing.
As you can see although I have tried to be as brief as possible, I have still taken up a considerable amount of room. I will end here for this month but if you spot any errors in the manual or think that there is something I should point out, please write and tell me so.

Paul Barbour

what printer?

While looking through the questionnaires, one of the most popular peripherals that people planned to buy was a printer. Many of you added the word 'cheap' to that. Because of this we decided to take a look at cheap printers compatible with your BBC Micro.

The cheapest proper printer is the Seikosha GP-100A (this has 10" wide paper—a GP-80A with 8" wide paper is available for about £20 less). For this you get an 80-column printer with full graphics capability. It runs fairly slow at 30 character/second but can offer a resolution of 480 pixels. 2000 sheets of listing paper costs £14 with a new ribbon costing

£4.75. this is generally accepted as the standard printer for the BBC Micro and is suitable for hobby use. If you are more interested in the business side of computing then I suggest you look towards the Epson range. This is outside the scope for 'cheap printers' but an article on professional quality printers will be printed in the future depending on demand.

As I said just now the GP-100A is fine for use as a hobby, however for many people the price might be prohibitive. So what is the alternative? the Sinclair printer must come to mind. It is possible to use the Sinclair printer with your BBC Micro for the cost of the printer (£59.95) and an interface (£29.95). This leaves you with a bill for £80.90 for a thermal printer capable of 32-columns. The Sinclair printer works by burning off small dots of aluminium on the metalised paper (which is very expensive). As any Sinclair owner will tell you the quality is not very good and so coupled with the expensive paper the purchase of a BBC compatible Sinclair is not really a good idea.

So far that leaves you with a choice of paying £250 for a normal printer or £90 for a poor quality thermal one. However, a company called Amber Controls Ltd. has seen the deficiency in the market and has produced a printer that retails for about £80. You will all be pleased to hear that it is completely compatible with the BBC Micro. Many of you have written to LASERBUG about this so we decided to look at this product in greater depth.



The Amber 2400 printer is a 24-column dot matrix printer which offers outstanding value for money (just like the BBC Micro!). As you can see from the photograph above it is fairly small (80 x 160 x 160mm) and sits neatly next to your micro.

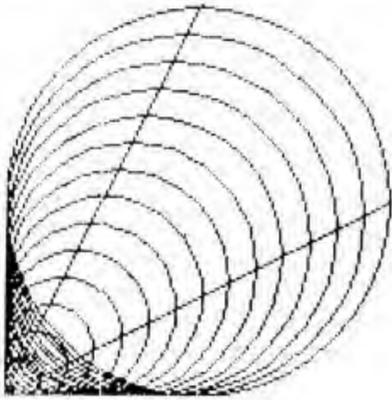
In these hard times, one of the main considerations when buying peripherals is cost. Hence before I continue the costs involved with the Amber is:

Amber 2400 printer	£80.44
Spare Ink Ribbon Cartridge	£1.90
5 Paper Rolls plus one Ink Cartridge	£2.97
RS423 Cable (custom made for BBC Micro)	£6.90
Parallel Cable (again custom made)	£11.44
RS423 Upgrade Kit	£14.37
Parallel Upgrade Kit	£11.44

In normal situations I would not go into depth about a product in an article such as this one but as I know many of you are seriously considering buying the Amber I will make an exception.

It has both an upper and lower case character set (an English style ASCII) totalling 96 characters in all. It uses a 5 x 7 matrix with normal text/double height/double width/double height and width/indented print/graphics options. In normal text it prints 24 characters/line and in graphics mode it is capable of 144 pixels. Sample prints are shown below:

GRAPHIC BY BBC COMPUTER



SAMPLE PRINT FROM
AMBER 2400
PRINTER

£ 69.95



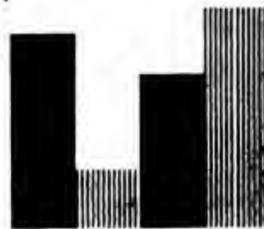
Uses normal plain paper.
For details please send
SAE to AMBER CONTROLS LTD
Central Way Andover
HANTS SP10 5AL

```

LIST
10 *FX5,1
20 REM BAR CHART
30 FOR L=1 TO 80
40 A%=&11:PROCout
50 PROCspaces
60FOR X=1 TO 4
70 READ A%,B%
80 IF B%>L THEN A%=0
90 FOR Y=1 TO 3
100 PROCout: NEXT Y
110 NEXT X
120 RESTORE
130 PROCspaces
140 NEXT L
150 A%=&14:PROCout
160 A%=&00:PROCout
170 VDU2
180 PRINT"      JAN. APR.
JLY. OCT. "
190 VDU3
200 END
210 DEF PROCspaces
220 FOR X=1 TO 3
230 A%=0:PROCout
240 NEXT X
250 ENDPROC
260 DEF PROCout
270 VDU1
280 PRINT CHR$(A%);
290 ENDPROC
300 DATA 255,10,170,60
,255,25,170,0

```

>RUN



JAN. APR. JLY. OCT.

You can use either the RS423 or Parallel port, the price difference being an added £5 for the centronics version. The difference in actual fact is only between the leads. If you have a model A, upgrade kits are available together with full instructions for fitting them. If you buy the parallel upgrade you also obtain a user port into the bargain since there are two 8 bit outputs provided from the 6522. The print speed is quoted as 0.7 lines a second with the paper being 58mm wide (or 2 1/4" for the old fashioned).

It would be unfair of me to go into further detail until I can review the product fully. Amber has kindly supplied us with one of these printers so a full review will appear in next months issue alongside with one of the Seikosha.

To summarise, if you want a printer for business use, I suggest you look at the Epson range. If your interest is more hobby orientated then consider the Seikosha. However, if you do not require 80-columns and the mention of a £250 price tag makes your wallet go into palpitations then the Amber 2400 is a more than acceptable alternative.

Paul Barbour

spiral patterns

This program continuously generates interesting coloured patterns of a spiral nature. The form of the pattern is controlled by a single constant N. When the program is run this constant can either be directly entered or generated internally initially from a list (line 490) afterwards randomly (510, 520). The list can be changed without any other changes being required. Small changes in N can have a profound effect on the pattern, try 0.555 and 0.5556.

For a model A computer MODE5 is selected (110) with PROCA and PROCC (540-590, 640-720) used to select and rotate the colours whilst a model B computer uses MODE2 and the procedures at 600-620, 730-780. Lines 60, 160, 210 remove the flashing cursor whilst line 220 sets the "print N" format. Several of the unsuitable values of N are removed at line 240. The section of the program from 280 to 340 draws triangles with a constant included angle at the centre of the screen. This angle is a function of N (290). The triangle is rotated to abut onto the previous one. Lines 350-410 perform a similar function but with a reducing multiplier I% so as to give the spiral effect. Finally 430 rotates the colours at a ever increasing rate (440) until the maximum possible rate is reached when a new N is chosen either from the list (480, 490) or at random (520). The whole program is then repeated (530).

D. E. Susans

>LIST

```

10 REM SPIRAL PATTERNS by D.E.
SUSANS JUNE 1982 (C).
20 *KEY 10 OLD ;M
30 *KEY 8 RUN ;M
40 *KEY 9 DIM C%(3) ;M GOTO 150
;M
50 CLS
60 VDU23;11,0;0;0;0
70 U%=-1:M%=1
80 DIM C%(3)
90 PRINT TAB(6,7);"FOR A NEW PA
TTERN AT ANY TIME",TAB(5,10);"PRES
S THE BREAK KEY FOLLOWED BY",TAB(1
1,13);"KEY f9 OR KEY f8.",TAB(9,20
);"COMPUTER MODEL A OR B?"
100 X$=GET$
110 IF X$="A" THEN X%=5:Y%=3 ELS
E IF X$="B" THEN X%=2:Y%=6 ELSE GO
TO 90
120 Z%=210
130 N = 0.63
140 GOTO 200
150 MODE 4
160 VDU23;11,0;0;0;0
170 PRINT TAB(9,7);"ENTER ANY RE
AL NUMBER,",TAB(2,10);"Positive or
negative numbers between",TAB(4,1
2);"0.61 and 0.8998 give interesti
ng",TAB(4,14)"patterns.",TAB(0,18)
180 Z%=150
190 INPUT N
200 MODE X%
210 VDU23;11,0;0;0;0
220 @%=132106
230 IF X%=5THEN PROCA ELSE PROCB

240 IF (N=0) OR (N=1) OR (N=-1))
OR (N=2) OR (N=-2) OR (N=3) OR (N
=-3) THEN GOTO 520
250 PRINT TAB(0,0);"N=";N
260 MOVE 640,512: MOVE 640,512
270 A=0 :B=0 :C=0 :D=0
280 FOR I%=ABS(N+1/N+20) TO 0 ST
EP -1
290 A=(A+360/N)MOD 360

```

```

300 PLOT 85,640,512
310 PLOT 85,(INT (640+(SIN (RAD
A) * 512))), (INT (512+(COS (RAD A)
* 512)))
320 D= (D+1)MOD Y%
330 GCOL 0, (D + 1)
340 NEXT I%
350 FOR I%=200 TO 1 STEP -1
360 A=(A+360/N)MOD 360
370 PLOT 85,640,512
380 PLOT 85,(INT (640+(SIN (RAD
A)*I%*2.56))), (INT (512+(COS (RAD
A)*I%*2.56)))
390 D= (D+1)MOD Y%
400 GCOL 0, (D + 1)
410 NEXT I%
420 F=200
430 IF X%=5 THEN PROCC ELSE PROC
D
440 FOR J=1 TO F :NEXT J
450 F=F-1
460 IF F>1THEN 430
470 IF M%=0 THEN 520
480 READ N: ON ERROR GOTO 510
490 DATA 0.85,0.81,17,0.7,0.6236
,0.5544,0.5556,0.775,0.776,0.778,0
.823,0.657
500 IF N<>0 THEN 530
510 M%=0
520 IF RND(1)>=0.5 THEN N=SGN(RN
D(1)-0.5)*1000/(RND(60)+30) ELSE N
=SGN(RND(1)-0.5)*(0.5*RND(1)+0.5)
530 MODE X%:GOTO Z%
540 DEF PROCA
550 C%(1)=RND(7)
560 C%(2)=RND(7):IF C%(2)=C%(1)
THEN 560
570 C%(3)=RND(7):IF (C%(3)=C%(2)
) OR (C%(3)=C%(1)) THEN 570
580 VDU 19,1,C%(1),0,0,0,19,2,C%
(2),0,0,0,19,3,C%(3),0,0,0
590 ENDPROC
600 DEF PROCB
610 VDU 19,15,7,0,0,0
620 COLOUR 15
630 ENDPROC
640 DEF PROCC
650 C%(0)=C%(1)
660 C%(1)=C%(2)
670 C%(2)=C%(3)
680 C%(3)=C%(0)
690 FOR E%=1TO3
700 VDU 19,E%,C%(E%),0,0,0
710 NEXT E%
720 ENDPROC
730 DEF PROCD
740 D=(D+1)MOD Y%
750 FOR E%=1TO 7

```

```

760 VDU 19,E%,(D+E%)MOD Y% +1,0,
0,0
770 NEXT E%
780 ENDPROC

```

>

bazooka

Bazooka is a one player game in which you take over the role of an artillery soldier. From your vantage point you must destroy all enemy tanks (BLUE) and the falling bombs while being careful not to hit your own tanks (GREEN) and the ambulances. If you let an enemy tank go past or destroy one of your own tanks you get a black mark on your record card. If you destroy an ambulance, 2 black marks are given. If at any stage you get 5 black marks then you get an immediate Court Martial and sentenced to life imprisonment. The other way to end the game is to let a bomb get as low as the top of your gun. As the missiles move down the screen they become harder to hit and hence worth more points. Your bazooka is controlled using ,. keys (less/greater than) and your gun is fired using L. You can only have 1 missile on the screen at the same time.

The game uses full sound and graphics so have fun!

```

10 REM      BAZOOKA
20
30 REM (C) Paul Barbour
40
50 REM      1/8/82
60
70 REM      Version 1.1
80
90 ONERRORPROC BANG
100 *KEY10OLD:MRUN:M
110 VDU23,224,255,255,255,255,25
5,255,255,255,23,231,102,60,24,60,
126,126,60,24:ENVELOPE2,1,4,-4,4,1
,2,1,127,0,0,-5,80,80:ENVELOPE1,1,
4,-4,4,10,20,10,127,0,0,-5,100,100
:MODE1:VDU23:8202:0:0:0:COLOUR130
:GCOL0,130:CLS
120 AX=90:FIRE=FALSE:FIRING=FALSE
E:MOVING=FALSE:BOMB=FALSE:VX=-1:VY
=-1:MX=-1:MY=-1:XS=-1:YS=-1:E#=STR
ING$(3,CHR$224):S%=0:M%=0:SCORE=TR
UE:PROCGUN(5)
130 IF INKEY(-103) THEN PROCGUN(7):
AX=AX-2
140 IF INKEY(-104) THEN PROCGUN(7):
AX=AX+2
150 IF INKEY(-87) AND FIRING=FALSE
THEN FIRE=TRUE
160 IFA%<50 THEN AX=50
170 IFA%>130 THEN AX=130
180 PROCVAN:PROCGUN(5):IFFIRE=TR
UE THEN PROCFIRE
190 IFScore=TRUE THEN COLOUR0:PRI
NTTAB(0,30):"SCORE ";S%:" ";TA
B(26):"BLACK MARKS ";M%:SCORE=FALSE
E
200 IF M%>=5 THEN PROC MARTIAL:END

```

```

210 PROCBOMB:GOTO130
220 DEFPROCFIRE IFFIRING=TRUE TH
EN240
230 MX=RX*7.1111111:MY=SIN(RAD(
RX))*128:XS=((MX-640)/256)*64:YS=(
MY/256)*64:FIRING=TRUE:SOUND0,-15,
6,10:SOUND0,-5,4,255
240 MOVEMX,MY:PLOT7,MX+XS,MY+YS:
GCOL0,3:MX=MX+XS:MY=MY+YS:MOVEMX,M
Y:DRAWMX+XS,MY+YS
250 HX%=(MX+XS)/32:HY%=32-((MY+Y
S)/32):IF(HY%=VY AND(HX%)=VX ANDHX
%<=VX+3)OR(HY%=VY-1 AND(HX%)=VX A
NDHX%<=VX+3)THENPROCDEAD:FIRE=FAL
SE:FIRING=FALSE:MOVING=FALSE
260 IFMX>1280ORMX<0ORMY>1024THEN
FIRE=FALSE:FIRING=FALSE:MOVEMX,MY:
PLOT7,MX+XS,MY+YS:SOUND&10,-10,6,5
270 ENDPROC
280 DEFPROCGUN(T) GCOL0,3:MOVE64
0,0:PLOTT,RX*7.1111111,SIN(RAD(RX
))*128:MOVE620,0:DRAW660,0:PLOT85,
640,30:ENDPROC
290 DEFPROCVAN IFMOVING=TRUE THE
N400
300 IFRND(25)>1THENENDPROC
310 T=RND(3):ONT GOTO320,330,340
320 VDU23,225,0,0,0,1,3,2,6,4,23
,226,0,0,0,255,241,245,196,223,23,
227,0,0,0,224,224,224,96,96,23,228
,15,15,63,63,63,56,59,3,23,229,196
,245,241,255,255,120,123,3,23,230,
96,224,224,224,224,96,96,0:GOTO350
330 VDU23,225,0,0,0,0,0,255,1,3,
23,226,0,0,0,60,126,255,255,255,23
,227,0,0,0,0,0,0,128,192,23,228,12
7,127,127,32,44,45,32,31,23,229,25
5,255,255,0,90,24,0,255,23,230,255
,255,255,2,26,154,2,252:GOTO350
340 VDU23,225,0,0,0,0,0,0,1,3,23
,226,0,0,0,60,126,255,255,255,23,2
27,0,0,0,0,0,255,128,192,23,228,12
7,127,127,32,44,45,32,31,23,229,25
5,255,255,0,90,24,0,255,23,230,255
,255,255,2,26,154,2,252:GOTO350
350 VS=RND(1):IFVS<0.1THEN350
360 IFT=1THENVDU19,1,1,0,0,0:S=0
370 IFT>1THENS=RND(3):IFS=1THENV
DU19,1,2,0,0,0:ELSEIFS>1THENVDU19,
1,6,0,0,0
380 MOVING=TRUE:VY=RND(15)+5:IFT
=3THENVX=3:ELSEVX=37:VS=-VS
390 IFT=1THENSOUND1,1,100,255:EL
SESOUND2,2,5,255
400 PRINTTAB(0,VY)SPC(40)TAB(0,V
Y+1)SPC(40):VX=VX+VS:IFVX>37ORVX<0
THENMOVING=FALSE:SOUND&11,0,0,0:SO
UND&12,0,0,0:IFT>1ANDS>=2THENT=-10
:S=-10:VX=-10:VY=10:M%=M%+1:SCORE=
TRUE:ENDPROC
410 IFVX>37ORVX<0THENT=-10:S=-10
:VX=10:VY=-10:ENDPROC
420 COLOUR1:PRINTTAB(VX,VY):CHR#
225:CHR#226:CHR#227:TAB(VX,VY+1):C
HR#228:CHR#229:CHR#230:ENDPROC
430 DEFPROCDEAD COLOUR3:MOVEMX,M
Y:PLOT7,MX+XS,MY+YS:SOUND&10,-15,6
,10:PRINTTAB(VX,VY):E#:TAB(VX,VY+1
):E#:NOW=TIME+50:REPEATUNTILTIME>N
OW:PRINTTAB(0,VY)SPC(40)TAB(0,VY+1
)SPC(40):SOUND&10,0,0,0:SOUND&11,0
,0,0:SOUND&12,0,0,0:VX=-1:VY=-1
440 SOUND&13,0,0,0:FORX=-15TO0ST
EP,1:SOUND0,X,7,1:SOUND1,0,RND(25)
+175,1:NEXT:PROCHECK:ENDPROC
450 DEFPROCBOMB IFBOMB=TRUE THEN
470
460 BX=RND(18)+5:BY=0:BOMB=TRUE
470 PRINTTAB(BX,BY)" ":COLOUR0:BX
=BX+((RND(3)/10)-.2):IFBX<5THENBX
=5
480 IFBX>33THENBX=33
490 BY=BY+(RND(1)/4):PRINTTAB(BX
,BY):CHR#231:IFBY>27THENPROCBANG
500 HX%=(MX+XS)/32:HY%=32-((MY+Y
S)/32):IFHX%=INT(BX) ANDHY%=INT(BY
) THENSOUND&10,-15,6,5:MOVEMX,MY:P
LOT7,MX+XS,MY+YS:COLOUR3:PRINTTAB(
BX,BY):CHR#224:NOW=TIME+25:REPEATU
NTILTIME>NOW:PRINTTAB(BX,BY):" ":B
OMB=FALSE:FIRE=FALSE:FIRING=FALSE
510 IFHX%=INT(BX) ANDHY%=INT(BY)
THENS%=S%+INT(BY*9.4):SCORE=TRUE:BX
=-1:BY=-1
520 SOUND&13,-7,(27-BY)*9.4,255:
ENDPROC
530 DEFPROCHECK IFT=1THENS%=S%-
2500:M%=M%+2:SCORE=TRUE:GOTO610
540 IFS=1THENS%=S%-1000:M%=M%+1:
SCORE=TRUE:GOTO580
550 IFS>=2THENS%=S%+1000:SCORE=T
RUE
560 ENDPROC
570 DATA"FOOL, you shot your own
tank !","Are you colour blind, he
was one of ours","IDIOT - he was
on our side !","Can't you tell one
of your own tanks ?","Do you alwa
ys shoot your own tanks ?"
580 IFS=1THENRESTORE570:FORL=1TO
RND(5):READC#:NEXT:SOUND&10,-15,2,
10:SOUND&11,0,0,0:SOUND&12,0,0,0:S
OUND&13,0,0,0:TA=INT((40-LEN(C#))/
2):COLOUR0:PRINTTAB(TA,10):C#:NOW=
TIME+500:REPEATUNTILTIME>NOW:PRINT
TAB(TA,10):SPC(40)

```

```

590 ENDPROC
600 DATA "That was an unarmed Ambulance you shot!", "Ever heard of the Geneva Convention?", "You just murdered defenseless People!", "MURDERER - that was an innocent vehicle", "Do you always kill injured People?"
610 IFT=1THENRESTORE600:FORL=1TORND(5):READC#:NEXT: SOUND&10,-15,2,10: SOUND&11,0,0,0: SOUND&12,0,0,0: SOUND&13,0,0,0: TA=INT((40-LEN(C#))/2): COLOUR0: PRINTTAB(TA,10): C#: NOW=TIME+500: REPEATUNTILTIME>NOW: PRINTTAB(TA,10): SPC(40)
620 ENDPROC
630 DEFPROC MARTIAL CLS: COLOUR0: PRINTTAB(13,1): "COURT MARTIAL" TAB(13)"=====" "'''" You have obtained 5 black marks on ""'"" your record card. These could have been ""'"" given for letting enemy tanks go past.""
640 PRINT "shooting your own vehicles or destroying ""'"" ambulances."" ""'"" Hence you have been Court Martialed ""'"" and sentenced to life imprisonment! ""'"" In your brief military service you ""'"" amassed a total score of ";S%
650 #FX15,1
660 SOUND&10,0,0,0: SOUND&11,0,0,0: SOUND&12,0,0,0: SOUND&13,0,0,0: PRINT ""'"" "Another game (Y or N)"
670 A#=GET#: IFA#="Y" THEN RUN
680 IFA#="N" THEN END: ELSE 670
690 DEFPROC BANG MOVEMX,MY: PLOT7, MX+XS,MY+YS: PRINTTAB(BX, BY): " " : SOUND&11,0,0,0: SOUND&12,0,0,0: SOUND&13,0,0,0: V=0: FOR S=-15 TO 0 STEP .1: VDU19,2,V,0,0,0: SOUND&10,S,RND(3)+3,1: V=V+1: V=V MOD 16: NEXT: VDU19,2,3,0,0,0: #FX15,1
700 PRINTTAB(10,10) "Another game (Y or N)"
710 A#=GET#: IFA#="Y" THEN RUN
720 IFA#="N" THEN END: ELSE 710
    
```

four in a row

Four-in-a-row is a game you might be familiar with under the name of Connect-4 (c) MB games. Use the cursor left/right keys to move the counter and the cursor down key to drop it. You may play either the computer or another player. The computer is not particularly good at the game—it can cope with horizontal and vertical lines but not diagonal. Its move is also calculated randomly if it is not trying to block you. To

improve both its speed and playing ability, try altering PROCcheck and PROCcomp. If you manage to write a version in assembler then please write in and we will publish your alternative version.

Oh and I never mentioned what the rules are—simply try to get four counters in a row!

```

>L.
1 REM FOUR-IN-A-ROW
2
3 REM (C) Paul Barbour
4
5 REM Version 1.1
6
7 IFRX=FALSE THEN A%=0: B%=0: R%=TRUE:
8 ON ERROR PROC error
9 #KEY10OLDIMPROC error IM
10 MODE5: VDU23,8202,0,0,0,0: #FX12,50
11 #FX4,1
12 PROC char_col: PROC intro: CLS: PROC @rid: PROC stand: DIM @rid(7,6)
13 FOR player=1 TO 2: IF player=2 AND comp=TRUE THEN PROC comp: X=90: GOTO 15
14 PROC move: X=(X-1)/2
15 PROC drop: IF error=TRUE THEN
14
16 PROC check: NEXT player: GOTO 13
17 DEFPROC char_col VDU23,224,0,24,60,126,126,60,24,0,23,225,255,231,195,129,129,195,231,255,23,226,255,255,255,255,255,255,255: VDU19,3,2,0,0,0,19,0,7,0,0,0: ENDPROC
18 DEFPROC @rid LOCAL @rid1#, @rid2#, @rid3#, X, Y: @rid1#=STRING$(15, CHR$(226)): @rid2#=CHR$(226)+CHR$(225): @rid3#=CHR$(226): PRINTTAB(0,5): FOR X=1 TO 6: PRINTTAB(2) @rid1# TAB(2): FOR Y=1 TO 7: PRINT @rid2#: NEXT: PRINT @rid3#: NEXT: PRINTTAB(2) @rid1#: ENDPROC
19 DEFPROC stand GCOL0,3: MOVE128,736: DRAW64,736: DRAW64,320: MOVE32,320: DRAW96,320: MOVE1088,736: DRAW1152,736: DRAW1152,320: MOVE1120,320: DRAW1184,320: COLOUR1: PRINTTAB(0,1): "RED "; A%: COLOUR2: PRINT " YELLOW "; B%: ENDPROC
20 DEFPROC intro PRINTTAB(3,1) "FOUR-IN-A-ROW" TAB(3) "=====" ""'"" TAB(3) "THE RULES ARE" ""'"" TAB(2) "SIMPLE - TRY TO" ""'"" TAB(1) "GET FOUR COUNTERS" ""'"" TAB(2) "IN A ROW IN ANY" ""'"" TAB(5) "DIRECTION." ""'"" TAB(1) "CURSOR OR LEFT/RIGHT" ""'"" TAB(1):
21 PRINT "KEYS MOVE COUNTER," ""'"" TAB(1) "CURSOR DOWN ENTERS" ""'"" TAB(5) "THE PIECE," ""'"" ""'"" DO YOU WISH TO SET""'"" "THE SCORE TO 0 (Y/N)"
    
```

```

22 zero#=GET#:IFzero#="Y"THENA%
=0:B%=0:ELSEIFzero#="N"THEN23ELSE2
2
23 PRINTTAB(0,27);SPC(19);'"TAB
(2)"1 OR 2 PLAYERS ? "'
24 no#=GET#:IFno#="1"ORno#="2"TH
EN25ELSE24
25 IFno#="1"THENcomp=TRUE:ENDPR
OC:ELSEcomp=FALSE:ENDPROC
26 DEFPROCmove IFPlayer=1THENX=
3ELSEX=15
27 COLOURPPlayer:PRINTTAB(0,3);S
PC(39);TAB(X,3);CHR#224:*FX15,0
28 move=GET:IFmove<136ORMove>13
8THEN28
29 IFmove=138THENENDPROC
30 IFmove=136THENX=X-2
31 IFmove=137THENX=X+2
32 IFX<3THENX=15
33 IFX>15THENX=3
34 SOUND1,-15,0,1:SOUND2,-15,0,
1:SOUND3,-15,0,1:GOTO27
35 DEFPROCdrop LOCALY,Y1:error=
FALSE:X1=(X*128)+64:IFgrid(X,1)<>F
ALSE THENPRINTTAB(0,3);SPC(39);err
or=TRUE:ENDPROC
36 DATA30,1,766,2,702,3,638,4,
574,5,510,6
37 PRINTTAB(0,3);SPC(39):RESTOR
E:VDU5
38 READY,Y1:GCOL0,Player:MOVEX1
,Y:PRINTCHR#224:SOUND0,-15,2,1:IFY
1=6THEN41
39 IFgrid(X,Y1+1)<>FALSE THENgr
id(X,Y1)=Player:GOTO42
40 GCOL0,0:MOVEX1,Y:PRINTCHR#22
4:GOTO38
41 grid(X,Y1)=Player:GOTO42
42 VDU4:SOUND0,-15,1,1:ENDPROC
43 DEFPROCcheck LOCALX%,X1,X2,X
3,Y%,Y1,Y2,Y3,P:IFPlayer=1THENP=2E
LSEP=1
44 COLOURP:IFPlayer=1THENPRINTT
AB(0,26)"YELLOW PLAYER'S TURN"ELSE
PRINTTAB(1,26)"RED PLAYER'S TURN"
45 FORX%=1TO7:FORY%=1TO6:Y1=Y%-
1:Y2=Y%-2:Y3=Y%-3:IFY3<1THEN47
46 IFgrid(X%,Y%)=Player AND
grid(X%,Y1)=Player ANDgrid(X%,Y2)=
Player ANDgrid(X%,Y3)=Player THENP
ROCwin
47 X1=X%-1:X2=X%-2:X3=X%-3:
IFX3<1THEN49
48 IFgrid(X%,Y%)=Player AND
grid(X1,Y%)=Player ANDgrid(X2,Y%)=
Player ANDgrid(X3,Y%)=Player THENP
ROCwin
49 X1=X%-1:Y1=Y%+1:X2=X%-2:
Y2=Y%+2:X3=X%-3:Y3=Y%+3:IFX3<1THEN
52
50 IFY3>6THEN52
51 IFgrid(X%,Y%)=Player AND
grid(X1,Y1)=Player ANDgrid(X2,Y2)=
Player ANDgrid(X3,Y3)=Player THENP
ROCwin
52 X1=X%-1:Y1=Y%-1:X2=X%-2:
Y2=Y%-2:X3=X%-3:Y3=Y%-3:IFX3<1THEN
55
53 IFY3<1THEN55
54 IFgrid(X%,Y%)=Player AND
grid(X1,Y1)=Player ANDgrid(X2,Y2)=
Player ANDgrid(X3,Y3)=Player THENP
ROCwin
55 X1=X%+1:Y1=Y%+1:X2=X%+2:
Y2=Y%+2:X3=X%+3:Y3=Y%+3:IFX3>7THEN
58
56 IFY3>6THEN58
57 IFgrid(X%,Y%)=Player AND
grid(X1,Y1)=Player ANDgrid(X2,Y2)=
Player ANDgrid(X3,Y3)=Player THENP
ROCwin
58 NEXT:NEXT:PRINTTAB(0,26)
;SPC(20):ENDPROC
59 DEFPROCwin IFPlayer=1THENVDU
19,1,9,0,0,0ELSEVDU19,2,11,0,0,0
60 PRINTTAB(0,26);SPC(20):COLOU
RPlayer:PRINTTAB(5,26)"THE WINNER
":RESTORE61:FORloop=1TO5:READfreq,
dur:SOUND1,-15,freq,dur:SOUND2,-15
,freq+8,dur:SOUND3,-15,freq+16,dur
:NEXT
61 DATA100,5,108,5,116,5,124,5,
138,25
62 IFPlayer=1THENA%=A%+1ELSEB%=
B%+1
63 PROCstand:now=TIME+500:REPEA
TUNTILnow<TIME:COLOUR3:PRINTTAB(1,
28)"ANOTHER GAME (Y/N)"
64 cont#=GET#:IFcont#="Y"ORcont
#="N"THEN65ELSE64
65 IFcont#="Y"THENCLEAR:RUN:ELS
EVDU22,7
66 *FX4,0
67 *FX12,0
68 END
69 DEFPROCcomp LOCALX%,Y%,X1,Y1
,X2,Y2:FORY%=1TO6:FORX%=1TO7:Y1=Y%
-1:Y2=Y%-2:Y3=Y%-3:IFY3<1THEN71
70 IFgrid(X%,Y%)=1 ANDgrid(X%,
Y1)=1 ANDgrid(X%,Y2)=1 ANDgrid(X%,
Y3)=0THENgo=X%:GOTO77
71 X1=X%-1:X2=X%-2:X3=X%-3:
IFX3<1THEN73
72 IFgrid(X%,Y%)=1 ANDgrid(

```

```

X1,Y%)=1 AND%rid(X2,Y%)=1 AND%rid(
X3,Y%)=0 THEN%90=X%-3:GOTO77
73      X1=X%+1:X2=X%+2:X3=X%+3:
IFX3>7 THEN75
74      IF%rid(X%,Y%)=1 AND%rid(
X1,Y%)=1 AND%rid(X2,Y%)=1 AND%rid(X
3,Y%)=0 THEN%90=X%+3:GOTO77
75      NEXT:NEXT
76      %0=RND(7)
77      IF%rid(%0,1)>0 THEN76
78      ENDPROC
79      DEFPROCerror VDU22,5,19,3,2,
0,0,0,19,0,7,0,0,0:PRINT" YOU HA
VE PRESSED"" THE ESCAPE/BREAK"
"" KEY."" DO YOU WISH
TO"" LEAVE THE PROGRAM ?"
80      reply%=GET%:IFreply%="Y" THEN
VDU22,7:GOTO66:ELSEIFreply%="N" THE
NRUN:ELSEGOTO80

```

bookreview

Let your BBC Micro Teach You To Program by Tim Hartnell

194 pages £6.45 Interface Publications

Produced by now computer veteran Tim Hartnell, this book "takes you, step by simple step, through programming in BBC BASIC—with a number of simple programs—and then goes further, giving you leads to develop and expand your programming skills in the coming months." Sounds good doesn't it. This was a direct quote from the back cover of the book. From this the absolute beginner might pick up and buy this book. However, when this fictional person gets home, looks on page 4 and reads "no warranty in respect of the contents of this volume, or their suitability for any purpose, is expressed or implied." In short this means that although the cover (and the introduction) says the book is perfect for beginners, when the beginner gets home and becomes totally confused with BASIC and starts complaining to Interface, they can come back and say "well we did warn you..." I should emphasise that this is my own cynical view—you may draw what conclusions you like.

Even though the introduction says the book assumes "you don't know anything about programming" I feel the absolute beginner has little to gain from buying it. Far too many programs are too complex for their chapter e.g. When all you had learnt was how to print numbers and strings, REM, LIST, RENUMBER, RUN, STOP, END, NEW, OLD and print formatting—MODE, FOR loops, variables, TIME, REPEAT loops, RND, SOUND, and READ/DATA/RESTORE are used. Although it is explained in the text that the user will not understand how most of the program works and I realise that sometimes complex programs are needed to demonstrate a simple point, this is not a practise I would either recommend or support.

When you first start looking through the book you feel "uncomfortable" with it for two reasons. Firstly there is no index at the front. The index you would expect to find at the front is at the back and the alphabetical index you would expect to find at the back is nowhere to be seen. Secondly there are no chapters at all, just many sections. These two breaks from tradition are not ones I particularly like—they make using the book harder than it need be.

I said just now that the book really isn't suitable for beginners. If it isn't suitable for beginners then that means at least $\frac{1}{4}$ of the book is unsuitable for slightly more experienced programmers which in turn must make you ask the question is it worth the £6 odd that Interface charge you for it? Well, to gauge this I will go through some of the chapters that deal with

the more complex subjects.

We all know of FOR...TO...NEXT loops but until I met the BBC Micro I had not met REPEAT...UNTIL loops. How does Mr. Hartnell's book deal with this? Well for a start how does the BBC Bible (otherwise known as the User Guide) deal with it. The manual devoted $3\frac{1}{4}$ pages and 3 programs to the topic. Here it is clearly explained.

Tim's book has $1\frac{1}{4}$ pages and 2 programs on the subject. Out of the 3 loops illustrated, 2 of them are of the REPEAT...UNTILFALSE format—not a good idea. The REPEAT loop is met after FOR loops which cannot encourage structured programming. It is stated here, incorrectly, that the only way to get out of a REPEAT...UNTILFALSE loop is to press BREAK. If the only way to get out of an endless loop was to press BREAK then OLD ect., then I would be most upset. The way to get out of such a situation is to press ESCAPE. Both of the programs used in this section are concerned with graphics. I would have thought that simplest way to illustrate the way you terminate such a loop was with numbers—not drawing lines.

The next section I looked at was SOUND/ENVELOPE. There have been many articles on sound published in the computer magazines, one of them an excerpt from this book (Personal Computing Monthly—August '82). You are unlikely to gain more information from this book than any of the articles just mentioned.

From this section you will probably be able to work out how the simplest sound command works. It does not explain how to use the sound effects channel and the full version of sound (&HSFC) is explained in a rather confused manner. He attempts to explain envelope in $1\frac{1}{4}$ pages in the complicated style you will have seen elsewhere. Your best bet if you want details of sound is to stick to the manual.

MODE7 was the next topic to come under my scrutiny. This subject is handled extremely well and in my opinion is better than the same chapter in the User Guide in some respects. It starts off logically with teletext colour and then proceeds onto double height and graphics characters. Flashing characters are also mentioned although I didn't notice separated graphics being described. Also how to determine the code of a graphic character was missing (see LASERBUG Issue 1/2).

The section on PLOT described only the versions available for drawing straight lines and triangles. There are many programs after this (13 to be precise) using these commands (or MOVE and DRAW where appropriate). After this is a fairly unremarkable chapter on the VDU drivers. There are two programs at the end, both written by Jeremy Ruston. The first one is particularly interesting and shows how to scroll the screen sideways.

Arithmetic on the BBC Computer is dealt with just under $\frac{1}{4}$ of the way through the book—rather late I would have thought. To me, considering the way the section is written, I wouldn't have bothered to even include it.

Just after this is a section on procedures. The subject matter is dealt with in an easy to understand way although again I felt it came far too late and well after subroutines which are dealt with half a book behind.

The final section in the book is entitled "Improving your programs". I agree entirely with Tim Hartnell's comments here although in many places in the book he does not seem to "practice what he preaches". He emphasises the need for structured programming but always in the book describes unstructured methods before the structured ways.

If I had reviewed the book before I had seen the new manual I might have had a better opinion of it. I know for a fact that Tim Hartnell had seen the new User Guide before (or while) he was writing the book and as such he should have realised that he should be trying to improve on this. Instead he wrote a book that, although better than nothing, does not compare with the section in the manual for beginners. If Tim had produced this book at the time Basic Programming/Practical Programs were published he would have sold many. I cannot see the use of the book now and cannot see it selling very well. The best programs in the book were those written by Jeremy Ruston and as such next month we will review his book, also published by Interface. In actual fact, not every program in the book even works. The first program I choose to test had 2 errors in it. This must be the final comment on the usefulness of the book—how can a beginner be expected to use a book when there are errors in some of the programs?

Paul Barbour

BUG-BYTE BUG-BYTE SOFTWARE

051-227-2642

BBC SOFTWARE

NOW AVAILABLE

PHONE FOR DETAILS

Access or Barclay Card

Telephone Orders Accepted

BUG-BYTE BUG-BYTE BUG-BYTE

competition

This month sees the start of a series of different puzzles which you will need to solve with your BBC Micro. The prizes will only be small—three months free subscription to LASERBUG (to be added on when your present subscription finishes) but the main idea is to attempt to solve the problem set. The closing date for this month's competition will be 1st October. The results will be published as soon as possible after that date. Please send all entries to the usual address, marking the envelope COMPETITION 1.

So, on to this month's puzzle. Leonardo Fibonacci (1170–1250?) was a Florentine mathematician. He developed a series of numbers which is known today as the Fibonacci sequence. The series starts with the numbers 0 and 1, the remainder being the sum of the previous two, i.e. 0, 1, 2 (0+1), 3 (1+2), 5 (2+3), 8 (3+5), 11 (5+8), etc. What is so hard about this you may ask?—a computer is ideally suited to producing series. True, but this month's competition is to produce the most efficient program to do this. For the purposes of the puzzle we will define efficient to mean the shortest in both time and memory. As it is easier to produce a program that does either one or the other, two prizes will be given.

To summarise, the puzzle this month is to produce two programs (or one) that will print out on the screen the Fibonacci series. The rules for this month are:

- (i) The program must start with \emptyset and end with 1.2712788E38 (or thereabouts) and should not finish with an error.
- (ii) All numbers must be printed on the screen in MODE 7 for the quickest program.

- (iii) The first command in the program must be TIME= \emptyset and must finish by printing up the time.

For the shortest program:

- (iv) Program length is determined by TOP-PAGE.

To give you an idea of the kind of program you must write, the shortest program should be less than 55 bytes long and the quickest run in less than 4 seconds. Whilst experimenting for half an hour, the LASERBUG team managed to write a program in 48 bytes and another one that ran in 3.58 seconds. Can all you budding programmers out there beat that?

Remember, the closing date is 1st October and the two prizes are 3 month's free subscription to LASERBUG. Any suggestions you have for competitions would be welcome, too.

Paul Barbour

contacts

After the mammoth Contacts Page and Meeting Place last month we decided to give them a rest this issue. In future the pattern will be this:

Complete Contacts Page with updates for Meeting Place (Issue 5)
Updates on both Contacts Page/Meeting Place (Issue 6)
Complete Meeting Place with updates for Contacts Page (Issue 7)
Updates on both Contacts Page/Meeting Place (Issue 8).

This pattern will continue in an endless cycle. For the months we are at computer exhibitions and selling sample copies (i.e. the April and September editions) we will have complete lists of both.

What is happening in next months LASERBUG you might ask. Well, by hook or by crook the October issue will come out in October—even if the magazine is 2 pages long and I have to deliver them all by hand!

We have more reviews planned—Jeremy Ruston’s new book comes under scrutiny as does various pieces of software. Also, the first proper hardware review to appear in LASERBUG will be seen on the Amber 2400 printer.

A summary of the events at the PCW show will be printed in a show report page. Last April I spent most of my time talking about the BBC delays—I bet you I spend most of the PCW Show talking about LASERBUG’s delays!

All the regular articles will still be there and hopefully a few will be written by people other than myself—after last issue and this one it’s getting to the stage where we are thinking of changing the mags name from LASERBUG to The Paul Barbour Magazine! Ian Sinclair of Electronics and Computing fame is one of the writers to appear.

If you are interested in writing for LASERBUG, by the time you are reading this the final copy date for Issue 5 would have passed. However, for material to reach us in time for the November issue, it must reach the LASERBUG office by 12th October.

Have fun over the next month and if you pass a Spectrum (assuming Mr. Sinclair has managed to get any out of his factory) don’t forget to think of your BBC Micro and then have a little snigger.

Oh, if you are thinking of decorating, please get in touch because I have about 100 Spectrum leaflets that you can use as wallpaper . . .

Paul Barbour

Mr. Sheridan Williams of the other independent user group BEEBUG has been making claims in MicroComputer Printout that LASERBUG is run by Computers for All who advertise in our magazine.

I can state categorically here and now that this is not true. This stop press was added into the magazine at the very last moment and as such I cannot say too much here.

Next month we will both reproduce the letter in question (assuming we get permission from Printout) and a full, formal reply. In the meantime we are also submitting a reply to the magazine concerned who hopefully will print our letter in the October edition of their magazine.

Paul Barbour.

© LASERBUG 1982. Reasonable care is taken to avoid errors in this magazine. However, no liability is accepted for any mistakes which may occur. No material may be reproduced in any way without the written consent of the Editor.

LASERBUG is edited by Paul Barbour.
Final copy date for Issue 6 (November) is Tuesday 12th October.

Printed by Commercial Colour Press, Station Approach, Wanstead Park Station, Wanstead Park Station, Forest Gate, London, E7 OHX.

LASERBUG 4 Station Bridge, Woodgrange Road, Forest Gate, London, E7 ONF.

Membership: £12 for 1 year (£14 for overseas members) £6 for ½ year (£7 for overseas members). Other time scales available on application.

Advertising rates available on request.

LASERBUG
4 Station Bridge,
Woodgrange Road,
Forest Gate,
London E7 ONF



SUBSCRIBE NOW TO GET THE BEST OUT OF YOUR BBC MICROCOMPUTER

Please enrol me as a Member of LASERBUG. I enclose a cheque/postal order for £12.00 for 12 months subscription.

Name

Address

.....

..... Date.....

Telephone (Home) (Work)

To run from issue.....

I would/would not like my name and address/telephone number printed in the Contacts page of the LASERBUG newsletter.

Please make all cheques/postal orders payable to LASERBUG ... and you can always send a copy of this form if you don’t want to cut up your copy of LASERBUG...

Software for all

72 NORTH STREET, ROMFORD, ESSEX. TEL 0708 60725

Announce the launch of their range of **BBC** programs



For Model B only. **Grand Prix**

Time Trials around our race track. Includes computer controlled car to hinder your progress. 9 levels of difficulty.

£5.95



GHOST MAZE

For Model B only.

Our own version of this popular Arcade game. With colour & sound. 9 levels of difficulty.

£6.95

BEEBTREK

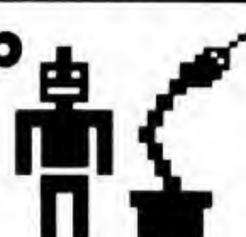
For Model A or B, real time advanced Startrek Game. Extra facilities include "probe satellites", "damage reports" & "on-board computer".



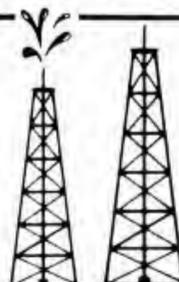
£7.95

Cobra/Robo-Swamp

For Model A or B (please state version required). Two addictive graphic games — insomniacs delight!



£6.95



J.R.

For Model B only. Two player game, features include exploration, drilling, employment and Price Wars.

£6.95



For Model A or B

DATABASE

Accepts up to ten definable fields.

Facilities include Quick Search, Sort and Hard copy — capable of storing up to 300 complete records in memory (Model B).

£9.95

INVADERS

For Model A or B. Classic Arcade game. With colour and sound.

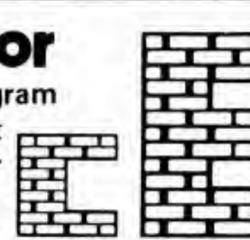
£6.95



Character Generator

For Model B only. Useful utility program enables user to re-define character set using Mode 4. Displays new character in graph form.

£4.95



Programmers

We are looking for good quality programs covering games, utilities and education. We pay excellent royalty rates. Please write or phone us on (0708) 60725

Dealer enquiries also welcome.

We're only a few minutes from your Post Box. So why not try our mail order service.



Paralander

For unexpanded model, graphics and sound are incorporated into this absorbing battle between you (the dare-devil) and the elements.



£4.50

ZX Spectrum

Disassembler

£4.00 Useful Utility Programme. Fits in 16K.

Please send me:-

Add £1 p&p per order.

..... £

..... £

..... £

I enclose Cheque/P.O. for - £

Please debit my Access/Barclaycard No.

Name

Address

Make cheques/P.O. payable to: **SOFTWARE FOR ALL**
72 North Street, Romford, Essex.
Tel: Romford (0708) 60725



**Attention
Dragon Owners!
Software Coming Soon!**

LIST RCN CLS CHAIN"" REN.5,5 ACTO5,5 *CAT

TIME OLD



LIST RCN CLS CHAIN"" REN.5,5 ACTO5,5 *CAT

TIME OLD



LIST RCN CLS CHAIN"" REN.5,5 ACTO5,5 *CAT

TIME OLD



