Issue 6 Nov'82

IN THIS ISSUE:

● Education

● Instant graphics

● Software protection Part 1.

● What monitor

. . . . and lots more!

You are now reading the 6th issue of LASERBUG which means we are half way through our first year. As you can see we have changed both our style and content from the first issue but even so I think you'll agree the magazine in itself is still informative.

First on my little list of things to say here concerns you—when I say you I mean you personally. You who are reading my Editorial. As you would have seen over the past few months LASERBUG has changed from being written almost totally by me to including other people's articles as well. Whilst talking to the people who have written articles, etc., for us they all seem to be surprised that what they have written has actually got in print. The thought of writing something and sending it off to a magazine seems so daunting a task that people are put off simply by thinking of it. Very few of the people who write things to submit to LASERBUG are so called "experts"—they are just normal people who do a normal job and in their spare time thought up this idea for an article and went ahead and wrote it. That is what I would like to encourage from you. No-one at LASERBUG or any of the magazines is a monster who ignores any contributions from the general public—we all welcome your ideas. If you have an idea in the back of your mind for an article then find a spare hour or two and go ahead, write it and send it off to whatever magazine you wish. Very few articles are dismissed completely. Even if the magazine you submit the article to isn't interested in printing it then try another one—don't be put off. The BBC Computer has a wide variety of users and hence a wide variety of opinions is required. Unfortunately at present we cannot offer payment for any articles we print. The idea of you writing something should not be for financial gain but for the satisfaction of knowing that your article has actually been published. Please if you have an idea for something that would make good reading go ahead and put it down on paper. Send it off to one of the magazines, be it us or whoever, and see if they like it. I wish you the best of luck if you attempt it.

Next it's moan and groan time. The BBC Computer started being "big news" in September 1981. This means the whole thing has been going for 15 months. Have you got a problem with anything? I know of people who ordered a Model B in October last year and still haven't got it. I personally have been waiting five months for an Acornsoft program without success. What I want to do is hear from you if you still have a problem that hasn't been solved and seems hopeless—if you have I will do my best to resolve it for you as fast as I can by going direct to the people concerned on your behalf. Even if your problem has been solved (eventually) let us know of it and I will air your views. The whole thing has been going for more than long enough to be running smoothly.

Remember me mentioning the Torch Computer a few months ago? Well as you all must know the Torch 2nd Processor/Dual Disk Drive for the BBC Micro was also launched recently (at the PCW Show). In the future we hope to carry a full article on the topic but for the time being I will outline both products for you.

First the Torch Computer—this I suppose you could describe as a BBC Micro in a posh box with a colour monitor and disk drives plus of course a 6MHz Z80A processor. The Torch Computer is styled like a proper business computer (which is what it really is). It comes in two sections, the keyboard and the main control unit which externally holds a 12" colour monitor and dual disk drives but internally holds the hardware. The main processor is a 6MHz Z80A with 64k RAM + 4k shadow ROM for bootstrap (this is essentially the 2nd processor) with the peripheral processor being a 2MHz 6502 with 32k RAM and 48k ROM and BBC BASIC (the BBC Computer part). All screen modes etc., are the same as the BBC Computer. The keyboard is not the standard BBC one but extended in so much as separate numeric, cursor control and editing pads. The disk drives are 5¼" 400k double track density types. The interfaces are the same as on the BBC Micro as are the Econet (called Torchnet), sound, etc. The computer is of course CP/M compatible. In the future a "Super Torch" is to be launched with a 16-bit Motorolas 68000 as well. Although Russel Lyons told me that "the Torch is a twin processor machine and uses the BBC Microcomputer board as one of its components", if you read between the lines it is not hard to come to the conclusion that Torch is in fact an expanded BBC Micro with 2nd processor in a new case with a monitor and disk drives built in.

And so on to the Torch Z80 disk pack for the BBC Computer. What this really does is give you the equivalent of the Torch computer except that instead of all being in a custom made case the unit is separate (and of course you need your own monitor). When you are buying the Torch Z80 disk pack you are buying the 2nd processor and dual disk drives in one unit. Two connections are made to your computer—one to the tube and one to the disk sockets. I know a good deal of people who are extremely interested in this system and the fact that they have beaten Acorn by at least six months must mean something.

It is nice to see that someone can actually produce hardware for the BBC Micro and sell it successfully without a huge backlog forming (I hope I haven't spoken too soon!). I do not know of the price of the Torch Computer (about £2,000?) but the Z80 Disk Pack is £995 + VAT. Torch can be found at Abberley House, Great Shelford, Cambridge CB2 5LQ. I would like to thank all at Torch who supplied me with information.

Before we leave Torch completely, I should just mention that they have also recently launched a light pen for the BBC Micro (and Torch) which as far as I know is the first one available (that makes two firsts for Torch—well done!).

Fancy seeing LASERBUG on TV? Well that in itself remains to be seen but what you might see on your TV soon is programs from us. There are two telesoftware mediums—Teletext and Prestel. As far as Teletext are concerned we will probably be allowing the Telesoftware & Primary Education Project to broadcast some of our programs on Teletext. For Prestel some of our programs might well be appearing on Aladdin's Cave and as well as all this we will probably have some pages of our own on this system so we can broadcast our latest news over the air (well, over the telephone line at least!).

You probably would have noticed last month that we started up a Special Offer section. How that went I do not know yet as at the time of writing this, Issue 5 was still at the printers. Hopefully it will be very popular and at any rate you will get some bargains here. To take the dust covers; the cheapest anyone else is selling these particular ones is £3.00—we offer them to you at only £2.50 inclusive. The idea of this section isn't simply so that we can make money—I don't want a Ferrari unlike certain other people. This means we can keep our costs down and hence the £12.00 membership fee to LASERBUG can stay at that rate for a good while longer. This is also the reason that we are taking on more advertising.

If people are to be believed, this is the month that the new operating system becomes available. I don't think Acorn can delay it much longer and so hopefully one way or another it will be released this month. We will bring you a full report on it once it becomes generally available.

At last Acorn have clarified the position with regards to people performing DIY upgrades. You can do this without invalidating your guarantee ON THE CONDITION THAT (i) you do not touch the rest of the machine and (ii) you only use the officially recommended parts. Please remember the second part—many people offer upgrade parts but most of these are unofficial and WILL invalidate the guarantee.

Rumour has it that the BBC is totally fed-up with the problems the BBC Computer has had. They are not upset with the computer itself— that works fine and performs extremely well. What they don't like is Acorn and the way they have handled the BBC Computer. To be precise all the delays that have occurred—this does not enhance the BBC's reputation. Hence the Beeb have decided never to lend their name to a product again—unofficially of course!

Our colleagues at Acorn User have decided to move their magazine onto the news-stands this month, following in the footsteps of the Sinclair-based magazines like Sinclair User. I wish them the best of luck although I think the £1 price tag might be considered a bit high by some people when compared with other magazines. However, as I am often saying, magazines such as ours (LASERBUG, Acorn User, BEEBUG, Beebon) do offer info about only the computer you are interested in (although Acorn User is for the Atom as well) which makes the extra money worthwhile.

Are you fed-up with the BBC Joysticks? It is the general opinion that Atari joysticks are the best ones around and I must admit I tend to agree with that statement. Hence you will be pleased to hear that Oakleaf

Computers of Grantham sell an interface that will allow you to connect two Atari joysticks to your BBC Micro. This costs £13.95 and connects the joysticks not through the analogue port but through the 20 way user port. This means that any Atari compatible joystick can be connected including Le Stick which is the ultimate in joysticks. Full software is supplied with the interface. Oakleaf can be found at 121 Dudley Road, Grantham, Lincs. NG31 9AD.

As I know we have a good number of teachers who read LASERBUG I will just add a little news item here for you. As you probably know schools can purchase one of three computers at half price—a Spectrum, BBC or RM380Z. During September, Sinclair offered that if a school bought a Spectrum they would receive a free ZX printer, Logo language and 10 vouchers worth either a £45 discount on other machines or another ZX printer (only 1 discount/mirco). Now Acorn has also launched a scheme for schools. If a school buys a Model B, whether it be under the half-price scheme or not, they will also receive a free Econet interface worth £45. This is a very helpful addition to schools and should be very encouraging to teachers. What will RM offer I wonder?

Please address all correspondence to:

LASERBUG,
10 Dawley Ride,
Colnbrook,
Slough,
Berks.
SL3 0QH.

Hopefully now we have changed our address all letters can be dealt with much more efficiently and quickly and should lead to a better service. It helps us considerably when sorting the mail if you can add one or two words in the top left-hand corner of your envelope to describe what your letter is about, e.g. MEMBERSHIPS, PROGRAMS, QUERIES, etc. We regret that we cannot reply to any letter unless you enclose an SAE.

Paul Barbour

contents

As we promised last month, a new series of educational programs starts in LASERBUG. As every teacher knows educational software is very thin on the ground, especially for the BBC Micro, at the moment. Hence we bring you the program below.

Maths Race is designed to test the mental arithmetic abilities of children. It tests addition and subtraction up to 50, i.e. 49+49 is the hardest question and multiplication and division up to 12×12/144/12. Children between 10 and 12 of normal ability should be capable of using the program, however anyone of the necessary level can use it.

Full instructions are given in the program which only runs on a Model B/Model A with memory upgrade.

```
  10 REM          Maths Race
  20
  30 REM Written by Paul Barbour
  40
  50 REM            5/10/82
  60
  70 REM          Version 1.0
  80
  90 REM    (c) LASERBUG 1982
 100
 110
 120 MODE7:VDU23;8202;0;0;0;
 130 VDU23,224,28,28,8,28,42,8,20
,34
 140 VDU23,225,0,24,24,0,255,0,24
,24
 150 PROCintro
 160 CLS
 170 PROCintro_test
 180 MODE2:VDU23;8202;0;0;0;
 190 PROCdraw_track
 200 PROCgame
 210 MODE7:VDU23;8202;0;0;0;
 220 PROCcontinue
 230 GOTO180
 240 DEFPROCintro_test
 250 PRINT"REMEMBER:";CHR$136;"DE
LETE";CHR$137"will rub out the las
t"'"         thing you entered."'
'"         You must press";CHR$13
6;"RETURN";CHR$137;"after"'"
   you have put the answer to a"'
"         question."'''
 260 C%=0
 270 FORC=1TO10
 280 ONRND(4)GOSUB400,470,520,580
 290 PRINTCHR$129;C;". What is ";
number1;function$;number2;" ?";CHR
$133;
 300 INPUT""answer
 310 IFanswer=real_answer THENPRI
NT'CHR$130"CORRECT !":C%=C%+1:PROC
right
 320 IFanswer<>real_answer THENPR
INT'CHR$131"WRONG !   It was ";real
```

```
_answer:PROCwrong
  330 PRINT''
  340 NEXT
  350 PRINTCHR$134;"You scored ";C
%;" out of 10"
  360 PRINT''CHR$157;CHR$129;"Pre
ss the Space Bar to start game..."
  370 REPEATA=GET:UNTILA=32
  380 ENDPROC
  390 REM +++ Addition subroutine
+++
  400 REPEATnumber1=RND(50)
  410 number2=RND(50)
  420 UNTILnumber1>number2
  430 real_answer=number1+number2
  440 function$=" + "
  450 RETURN
  460 REM --- Subtraction subrouti
ne ---
  470 GOSUB400
  480 real_answer=number1-number2
  490 function$=" - "
  500 RETURN
  510 REM *** Multiplication subro
utine ***
  520 number1=RND(12)
  530 number2=RND(12)
  540 real_answer=number1*number2
  550 function$=" x "
  560 RETURN
  570 REM ~~~ Division subroutine
~~~
  580 number2=RND(12)
  590 real_answer=RND(12)
  600 number1=number2*real_answer
  610 REM * This line only works o
n OS 0.1 *
  620 IF?&871=7THENfunction$=" ~ ":
ELSEfunction$=" "+CHR$225+" "
  630 RETURN
  640 DEFPROCintro
  650 PRINTCHR$157TAB(13);CHR$141;
CHR$132"MATHS RACE"'CHR$157TAB(13)
;CHR$141;CHR$132"MATHS RACE"'CHR$1
57TAB(14);CHR$132"_____ ____"
  660 PRINT''CHR$134;"      In this
game you must race against"CHR$13
4"the computer over 800 metres."
  670 PRINT'CHR$134;"      You move
by answering mental"'CHR$134"arit
hmetic questions which could be on
"'CHR$134"addition,  subtraction,
 multiplication"CHR$134"or divisio
n."
  680 PRINT'CHR$134"      The quick
er you answer the"'CHR$134"questio
ns the more you move.  The"'CHR$13
4"computer moves 20m per question.
  If"'CHR$134"you answer within 5
seconds you move"
  690 PRINTCHR$134"40m,  within 10
seconds 20m and any"'CHR$134"long
er than that 10m.  If you get a"'C
HR$134"question wrong then you do
not move at"'CHR$134"all."
  700 PRINT''CHR$157;CHR$129;"Pres
s RETURN for 10 sample questions"
  710 REPEATA=GET:UNTILA=13
  720 ENDPROC
  730 DEFPROCdraw_track
  740 COLOUR131
  750 CLS
  760 VDU29,640;512;
  770 FORX%=0TO360STEP4
  780 S=SINRAD(X%):S1=SINRAD(X%+30
)
  790 C=COSRAD(X%):C1=COSRAD(X%+30
)
  800 SOUND&11,-15,X%/2,5:SOUND&12
,-15,(X%/2)+4,5:SOUND&13,-15,(X%/2
)+8,5
  810 MOVES*440,C*312
  820 DRAWS*640,C*512
  830 PLOT85,S1*640,C1*512
  840 DRAWS1*440,C1*312
  850 PLOT85,S*440,C*312
  860 NEXT
  870 GCOL0,6
  880 MOVE0,300:DRAW0,512
  890 ENDPROC
  900 DEFPROCright
  910 S=RND(100)
  920 FORX=S TOS+100STEP.1:SOUND&1
1,-15,X,1:NEXT
  930 ENDPROC
  940 DEFPROCwrong
  950 FORnoise=1TO100STEP.1:SOUND&
10,-15,2,1:NEXT
  960 ENDPROC
  970 DEFPROCgame
  980 P=0:C=0
  990 VDU28,5,21,14,10
 1000 COLOUR134:CLS
 1010 VDU5
 1020 GCOL0,1
 1030 MOVESINRAD(0)*590-30,COSRAD(
0)*462+10:PRINTCHR$224
 1040 GCOL0,4
 1050 MOVESINRAD(0)*490-30,COSRAD(
0)*362+10:PRINTCHR$224
 1060 VDU4
 1070 ONRND(4)GOSUB400,470,520,580
 1080 CLS
 1090 COLOUR1:PRINT'" What is:"
```

```
 1100 PRINT''" ";number1;function$
;number2
 1110 TIME=0
 1120 INPUT''" ? "answer
 1130 VDU5
 1140 GCOL0,7
 1150 MOVESINRAD(P/2.2)*590-30,COS
RAD(P/2.2)*462+10:PRINTCHR$224
 1160 MOVESINRAD(C/2.2)*490-30,COS
RAD(C/2.2)*362+10:PRINTCHR$224
 1170 VDU4
 1180 IFanswer=real_answer THENCLS
:PRINT''''"CORRECT  !"
 1190 IFanswer=real_answer ANDTIME
<500THENP=P+40
 1200 IFanswer=real_answer ANDTIME
>=500ANDTIME<1000THENP=P+20
 1210 IFanswer=real_answer ANDTIME
>=1000THENP=P+10
 1220 IFanswer<>real_answer THENCL
S:PRINT''''" WRONG  !"''" IT WAS"
''"    ";real_answer
 1230 C=C+20
 1240 VDU5
 1250 GCOL0,1
 1260 MOVESINRAD(P/2.2)*590-30,COS
RAD(P/2.2)*462+10:PRINTCHR$224
 1270 GCOL0,4
 1280 MOVESINRAD(C/2.2)*490-30,COS
RAD(C/2.2)*362+10:PRINTCHR$224
 1290 GCOL0,6
 1300 MOVE0,300:DRAW0,512
 1310 IFanswer=real_answer THENPRO
Cright:ELSEPROCwrong
 1320 IFP>=800ORC>=800THENENDPROC
 1330 GOTO1060
 1340 DEFPROCcontinue
 1350 IFP>=800THENPRINTTAB(12)CHR$
141,CHR$129;"WELL DONE !"'TAB(12)C
HR$141,CHR$130;"WELL DONE !"''CHR$
131;"You managed to beat the compu
ter."
 1360 IFC>=800THENPRINTTAB(13)CHR$
141,CHR$129;"BAD LUCK !"'TAB(13)CH
R$141,CHR$130;"BAD LUCK !"''CHR$13
1;"Unfortunately the computer beat
you."
 1370 PRINT''CHR$133"Would you lik
e another go ?"''CHR$133"(Press 'Y'
or 'N')"
 1380 A$=GET$:IFA$="Y"THENENDPROC
 1390 IFA$<>"N"THEN1380
 1400 RUN
```

how to use joysticks

At the PCW Show a few months ago the new BBC Joysticks were launched. By now a good number of people should have them and hence LASERBUG explains how to use them.

The joysticks are connected to the BBC Micro through the Analogue to Digital Convertor (or ADC for short). An analogue signal is produced by the joysticks and represents a voltage between 0 and 1.8v, the digital value is that produced by the computer and is simply a number which in the case of the BBC Micro is between 0 and 65520. We find the VALue of the ADconvertors by a command called ADVAL. As far as controlling joysticks are concerned ADVAL comes in 4 forms. ADVAL(0) for looking at the buttons, ADVAL(1) and ADVAL(2) for joystick 1 and ADVAL(3) and ADVAL(4) for joystick 2.

Let us start with ADVAL(0) which looks at the "Fire" buttons of the two joysticks. This also helps in finding out which joystick is which. Enter the following line:

```
MODE7:REPEATPRINTTAB(19,12);ADVAL(
0):UNTIL0
```

On running the number 0 should appear on the screen. If you now press the button on one of the joysticks either 1 or 2 will appear on the screen depending on which joystick you are holding. If you press both buttons 3 will appear on the screen. All this is quite logical.

Now on to the hard part—working out movement. I will only describe how to operate joystick 1 (i.e. limit myself to channels 1 & 2). The second joystick (i.e. channels 3 & 4) work in the same way. To start off with we will look at channel 1, which deals with the horizontal movement. Again enter the following line:

```
MODE7:REPEATPRINTTAB(16,12);ADVAL(
1);"       ";:SOUND0,-15,ADVAL(1)/256,
1:UNTIL0
```

On entering a value will appear on the screen between 0 and 65520 as described earlier along with a sound of appropriate pitch. By moving the joystick around you will see how the number varies. From this it is not clear how you can get simple values to allow you to move objects around.

The BBC Computer calculates a number between 0 and 4095 with the ADC. It then multiplies the number by 16 for the value of ADVAL. This is to allow for future expansion of the BBC Micro when hopefully it will be able to read values much more accurately. Hence we could divide the value of ADVAL by 16 to get a smaller and more managable value. For our purposes even 0-4095 is too large a range—numbers between 0 and 255 are fine. Hence for all readings we will divide ADVAL by 256. To make life simpler we will define a FuNction to make the calculation:

```
> 1000 DEFFNjstk(channel)=INT(ADVAL
(channel)/256)
```

This could be speeded up by 9% using integer division. Last month in *Programmers Corner* we ran a program that performed these functions much better (display-wise) but in case you missed that I will describe all the functions clearly. If you add the following lines to the PROCedure:

```
>  10 MODE7:VDU23;8202;0;0;0;
   20 PRINTTAB(0,11);CHR$129;"Chan
nel 1 ";FNjstk(1);"  L Horizontal m
ovement  "
   30 PRINTTAB(0,13);CHR$130;"Chan
nel 2 ";FNjstk(2);"  L Vertical mov
ement  "
   40 GOTO20
```

you get a clear indication of how to get up/down/left/right movement. However to summarise the details:

UP — FNjstk(2)=255
RIGHT — FNjstk(1)=0
DOWN — FNjstk(2)=0
LEFT — FNjstk(1)=255

Using this method, below appears an adapted version of the Moving Things program from LASERBUG Issue 4:

```
>L.
   10 MODE4
   20 VDU23;8202;0;0;0;
   40 X=20:Y=16
   50 PRINTTAB(X,Y);"*"
   70 PRINTTAB(X,Y);" "
   80 IFFNjstk(1)=255THENX=X-.2
   90 IFFNjstk(1)=0THENX=X+.2
  100 IFFNjstk(2)=0THENY=Y+.2
  110 IFFNjstk(2)=255THENY=Y-.2
  120 IFX<0THENX=0
  130 IFX>38THENX=38
  140 IFY<0THENY=0
  150 IFY>30THENY=30
  160 PRINTTAB(X,Y);"*"
  170 GOTO70
  180 DEFFNjstk(channel)=ADVAL(cha
nnel)DIV256
```

This however does not cater for diagonal movement. There is no simple way I have found of calculating these figures but after trial and error the following ranges seem to apply:

DIRECTION	CHANNEL 1 (3)	CHANNEL 2 (4)
Up/Right	greater than 0 less than 100	greater than 100 less than 255
Down/Right	greater than 0 less than 100	greater than 0 less than 100
Down/Left	greater than 100 less than 255	greater than 0 less than 100
Up/Left	greater than 100 less than 255	greater than 100 less than 255

For a program that means a good number of comparisons have to be made. If you have found an easier way then please let us know. We end with a program that incorporates all of the above features. It is designed for a Model B (Model A + 32k upgrade) but can be run on a standard Model A if line 10 is changed to MODE5. It is readily adaptable and can be used as the basis for several games.

Please write in and let us know what applications you have put your joysticks to and please let us know of uses other than simply games.

```
>
   10 MODE1:VDU23;8202;0;0;0;
   20 VDU19,0,6,0;0;19,2,4,0;0;
   30 REM Increment step
   40 S=8
   50 REM X-coord
   60 X=640
   70 REM Y-coord
   80 Y=512
   90 GCOL0,2:PROCcross
```

```
  100 GCOL0,0:PROCcross
  110 IFADVAL(0)=1THENGCOL0,RND(7)
  120 PROCjoystick_move
  130 GOTO90
  140 DEFPROCjoystick_move
  150 jstk1=FNjstk(1)
  160 jstk2=FNjstk(2)
  170 REM ** UP **
  180 IFjstk2=255THENY=Y+S
  190 REM ** RIGHT **
  200 IFjstk1=0THENX=X+S
  210 REM ** DOWN **
  220 IFjstk2=0THENY=Y-S
  230 REM ** LEFT **
  240 IFjstk1=255THENX=X-S
  250 REM ** UP/RIGHT **
  260 IFjstk1>0ANDjstk1<100ANDjstk
2>100ANDjstk2<255THENX=X+S:Y=Y+S
  270 REM ** DOWN/RIGHT **
  280 IFjstk1>0ANDjstk1<100ANDjstk
2>0ANDjstk2<100THENX=X+S:Y=Y-S
  290 REM ** DOWN/LEFT **
  300 IFjstk1>100ANDjstk1<255ANDjs
tk2>0ANDjstk2<100THENX=X-S:Y=Y-S
  310 REM ** UP/LEFT **
  320 IFjstk1>100ANDjstk1<255ANDjs
tk2>100ANDjstk2<255THENX=X-S:Y=Y+S
  330 IFADVAL(0)=1THENPROCfire
  340 ENDPROC
  350 DEFFNjstk(channel)=INT(ADVAL
(channel)/256)
  360 DEFPROCcross
  370 MOVEX-25,Y:DRAWX+25,Y
  380 MOVEX,Y-25:DRAWX,Y+25
  390 ENDPROC
  400 DEFPROCfire
  410 SOUND&10,-15,6,5
  420 GCOL0,1:MOVE100,0:DRAWX,Y:MO
VE1180,0:DRAWX,Y
  430 FORL=1TO100:NEXT
  440 GCOL0,0:MOVE100,0:DRAWX,Y:MO
VE1180,0:DRAWX,Y
  450 ENDPROC
```

Telesoftware update

In Issue 3 of LASERBUG we carried a small piece on Teletext and telesoftware for the BBC Mirco on this system. This month just to keep the record straight we are going to look at British Telecom's alternative, Prestel.

Although almost one million people have access to teletext through a special decoder either separate to the television or built-in a much smaller number of people have Prestel. The difference however as far as you are concerned is that adaptors to download telesoftware from Prestel are almost available—not, let me emphasise, from the BBC. A Telesoft Tantel which costs £190 + VAT with appropriate software soon to be released will download software through the RS232 into your micro. Other decoders are also being made from people other than Acorn at very competetive prices so I have been told but no information is available on these.

The difference in real terms between Teletext and Prestel is that once you have bought the adaptors or special TV's (which you need to buy for both systems) Teletext is completely free. With Prestel to start off with you have to have a jack 96A socket installed by BT which will cost you £15 and also raises your telephone rental by 15p per quarter. All the time you are using Prestel you are in fact making a phone call which means that you then incur these charges as well. On top of this, although most pages are completely free (as far as added charges are concerned) just to look at certain pages will cost you between ½p and 50p per page! You are given plenty of warning for pages that will cost you however to allow you to back out. However to balance all the downfalls out Prestel has approximately 200,000 pages whereas Teletext has about 800 altogether. Teletext because it comes down the aerial is only one way communication but Prestel allows 2-way talk, i.e. between you and the central Prestel computer. Uses of this can be between ordering goods through Prestel by giving companies your credit card number or leaving messages for other Prestel users.

The way Prestel works is this. Firstly, somebody must be the information provider for the system. For telesoftware this could for instance be Electronic Insight. Their information is then sent to the Prestel Centre where the Prestel computer stores it. When you call up this information (page 800) the information then goes through the telephone line and into your adaptor. The adaptor then transmits the data to either the TV or to your computer.

Back to telesoftware itself. On Prestel there are two main sources of telesoftware, Aladdin's Cave (page 700) and Electronic Insight (page 800). For the index enter ★70012# and for EI enter ★80066# A telesoftware program on Prestel looks extremely confusing and you have no chance of copying it out as it is coded quite differently from its

This month Hardreview *looks at the new BBC Joysticks. They retail at £13.00 and are available either direct from the BBC or any Acorn dealer.*

The BBC Joysticks are good in that the joystick control is very smooth in operation and requires very little force to move. The range of numbers produced by these controls is also very good, &20 to FFF0. However there is a very considerable variation in the readings even when the controls are not adjusted. These variations are about 1% of the full-scale reading and would make it essential to use an averaging technique if it was necessary to be able to steer the control of a particular pixel on a full screen control.

The two joysticks are not labelled in any way so which is which must be determined by a trial and error procedure.

It is unfortunate that the way the controls have been wired makes it that if the **ADVAL** readings are used directly in MOVE or DRAW statements (after division) then although the vertical direction is as expected the horizontal movement is reversed.

The whole size of the joysticks is unnecessarily large making it difficult even for an adult to grip easily and impossible for many children to hold in one hand. Hence a separate hand is required to operate the joystick control. There is very little inside the case and it would have been easy to have reduced the overall dimensions so that single-handed operation was possible.

It would also have been much easier to use the joysticks if the cables had been a foot or two longer.

D. E. Susans

What did people do before Rubik's Cube you may ask? The program below gives you one answer which was reproduced in many forms before the dreaded cube came onto the market. I can't give you the proper name for this puzzle simply because I don't remember it!

The game is based around a 4 × 4 grid. On the grid there are 15 letters and a space (or star in the program). You must rearrange the grid so as to form:

```
A B C D
E F G H
I J K L
M M O *
```

The piece you have to look at is the star. You may swap this with any of the letters surrounding it by pressing the appropriate cursor key. You continue ad infinitum you get the correct sequence. If you give up with the puzzle ESCAPE will get you out of it.

Persevere with it because it can be done and will give you many hours of enjoyment. Let us know you best times.

Paul Barbour

```
>L.
   10 REM   Puzzle Program
   20
   30 REM   by Paul Barbour
   40
   50 REM      Version 1.1
   60
   70 REM        7/10/82
   80
   90 REM  (C) LASERBUG 1982
  100
  110 ONERRORMODE7:PROCerror:CLS:E
ND
  120 DIMgrid$(4,4)
  130 VDU23,224,-1,-1,-1,-1,-1,-1,
-1,-1
  140 PROCmake_up_grid
  150 MODE5:VDU23;8202;0;0;0;:C%=1
  160 VDU19,4,4,0;0;
  170 TIME=0:C=0
  180 PROCdraw_grid
  190 PROCswap_get
  200 PROCswap
  210 PROCwin
  220 GOTO180
  230 END
  240 DEFPROCmake_up_grid
  250 DIMalpha$(16)
  260 FORalpha=65TO80
  270 alpha$(alpha-64)=CHR$(alpha)
  280 NEXT
  290 alpha$(16)="*"
  300 FORX=1TO4:FORY=1TO4
  310 REPEATchar=RND(16):UNTILalph
a$(char)<>""
  320 grid$(X,Y)=alpha$(char):alph
a$(char)=""
  330 NEXT:NEXT
  340 ENDPROC
  350 DEFPROCdraw_grid
  360 COLOUR135:COLOUR4:IFC%=1THEN
CLS:C%=0
  370 line$=STRING$(17,CHR$224):sP
ace$=" "+CHR$224+" "
  380 open$=" "+CHR$224+" ":close$
=" "+CHR$224
  390 blank$=open$+" "+sPace$+" "+
space$+" "+space$+" "+close$
  400 PRINTTAB(1,7)line$
  410 FORY=1TO4
  420 PRINTblank$
  430 PRINTopen$;:COLOUR1:PRINTgri
d$(1,Y);:COLOUR4:PRINTspace$;:COLO
UR1:PRINTgrid$(2,Y);:COLOUR4:PRINT
space$;:COLOUR1:PRINTgrid$(3,Y);:C
OLOUR4:PRINTspace$;:COLOUR1:PRINTg
rid$(4,Y);:COLOUR4:PRINTclose$
  440 PRINTblank$
  450 PRINT" "line$:NEXT
  460 FORX=1TO4:FORY=1TO4
  470 IFgrid$(X,Y)="*"THENCX=X:CY=
Y
  480 NEXT:NEXT
  490 ENDPROC
  500 DEFPROCswap_get
  510 *FX4,1
  520 *FX15,1
  530 direction=GET:IFdirection<13
6ORdirection>139THEN530
  540 SX=CX:SY=CY
  550 IFdirection=136THENSX=CX-1
  560 IFdirection=137THENSX=CX+1
  570 IFdirection=138THENSY=CY+1
  580 IFdirection=139THENSY=CY-1
  590 IFSX<1ORSX>4ORSY<1ORSY>4THEN
530
  600 SOUND&11,-15,200,2
  610 C=C+1
  620 *FX4
  630 ENDPROC
  640 DEFPROCswap
  650 grid$(CX,CY)=grid$(SX,SY)
  660 grid$(SX,SY)="*"
  670 CX=SX:CY=SY
  680 ENDPROC
  690 DEFPROCwin
  700 IFgrid$(1,1)="A"ANDgrid$(2,1
)="B"ANDgrid$(3,1)="C"ANDgrid$(4,1
)="D"ANDgrid$(1,2)="E"ANDgrid$(2,2
)="F"ANDgrid$(3,2)="G"ANDgrid$(4,2
)="H"ANDgrid$(1,3)="I"ANDgrid$(2,3
)="J"ANDgrid$(3,3)="K"ANDgrid$(4,3
)="L"THEN720
  710 ENDPROC
  720 IFgrid$(1,4)="M"ANDgrid$(2,4
)="N"ANDgrid$(3,4)="O"THEN740
  730 ENDPROC
```

```
   740 VDU22,7:PRINTCHR$131"WELL DO
NE !"
   750 PRINT''CHR$130"You managed t
o do it in ";(TIME DIV 360000);" h
rs ";(TIME DIV 6000)MOD 60;" mins
"'CHR$130;(TIME DIV 100)MOD 60;" s
ecs";
   760 GOTO800
   770 DEFPROCerror
   780 *FX4
   790 PRINTCHR$130"You gave up aft
er ";(TIME DIV 360000);" hrs ";(TI
ME DIV 6000)MOD 60;" mins "'CHR$13
0(TIME DIV 100)MOD 60;" secs";
   800 PRINT" and ";C;" moves."
   810 PRINT''CHR$131;"Want anothe
r attempt ?"
   820 A$=GET$:IFA$="Y"THENRUN
   830 IFA$<>"N"THEN820
   840 ENDPROC
```

seasonal 2–firework display

Following on from last month's program we present the LASERBUG firework display. The listing is longer than we would normally like for this section but as the long nights are closing in fast we thought you wouldn't mind too much.

We will leave you guessing as to what next month's program will do although its theme is obvious.

L..

```
    10 REM Seasonal 2 - November 5t
h
    20
    30 REM       Firework Display
    40
    50 REM   Written by Paul Barbour
    60
    70 REM           8/10/82
    80
    90 REM          Version 1.1
   100
   110 REM      (C) LASERBUG 1982
   120
   130
   140
   150 REM *** ROMAN CANDLE ***
   160
   170 REM * Set up counter for fus
e *
   180 C%=0
   190 REM * Set up screen *
   200 MODE2:VDU23;8202;0;0;0;
   210 REM * Draw firework & headin
g *
   220 REM (Switch off RED)
   230 VDU19,1,0;0;:GCOL0,1
   240 REM (Firework)
   250 MOVE0,0:DRAW640,0:MOVE600,0
   260 DRAW680,0:PLOT85,680,300
   270 DRAW600,300:PLOT85,600,0
   280 COLOUR1
   290 PRINT" LASERBUG FIREWORK"
   300 PRINT'TAB(6);"DISPLAY"
   310 REM (Switch RED on again)
   320 VDU20
   330 REM (Burning sound)
   340 SOUND0,-10,5,255
   350 REM Draw sparks
   360 FORZ%=2TO15
   370 REM (Switch off colour Z%)
   380 VDU19,Z%,0;0;
   390 FORX%=0TO90STEP4
   400 GCOL0,Z%:VDU29,640;0;
   410 PLOT69,SINRAD(X%+FNRND)*512,
COSRAD(X%)*512
   420 PLOT69,SINRAD(-X%+FNRND)*512
,COSRAD(-X%)*512
   430 C%=C%+1:VDU29,0;0;:GCOL0,1
   440 VDU29,0;0;
   450 REM (Draw fuse)
   460 MOVE600,0:PLOT7,0,0:MOVE600,
0
   470 DRAWC%*1.86335404,0
   480 NEXT:NEXT
   490 REM Bring up sparks
   500 SOUND&10,-15,6,10
   510 FORX=2TO15:VDU19,X,RND(15),0
;0;
   520 SOUND0,-RND(10),4,1:SOUND0,-
RND(10),4,1
   530 TIME=0:REPEATUNTILTIME=25
   540 NEXT
   550 REM * Make sparks sparkle *
   560 FORX=1TO1000
   570 REPEATA1=RND(15):UNTILA1<>1
   580 VDU19,A1,RND(15),0;0;
   590 SOUND0,-RND(10),4,1:NEXT
   600 REM * Kill off sparks *
   610 FORX%=2TO15:FORY%=X%TO15
   620 VDU19,Y%,RND(15),0;0;
   630 SOUND0,-RND(10),4,1:SOUND0,-
RND(10),4,1
   640 NEXT:VDU19,X%,0;0;
   650 TIME=0:REPEATUNTILTIME=50
   660 NEXT:FORY=300TO0STEP-1
   670 MOVE0,Y:PLOT7,1280,Y
   680 SOUND&10,-Y/20,4,1:NEXT
   690 DEFFNRND=10*(5-RND(10))
   700 TIME=0:REPEATUNTILTIME=200
   710
   720 REM *** TRAFFIC LIGHTS ***
   730
   740 REM * Draw firework *
   750 MODE1:VDU23;8202;0;0;0;
```

```
 760 GCOL0,1:VDU19,1,0;0;
 770 MOVE590,0:DRAW690,0:PLOT85,6
90,200
 780 DRAW590,200:PLOT85,590,0
 790 COLOUR1:PRINTTAB(7,0);"LASER
BUG FIREWORK DISPLAY"
 800 VDU19,1,6,0;0;
 810 REM * Burn fuse *
 820 FORX%=590TO0STEP-1:MOVE590,0
:PLOT7,0,0
 830 MOVE590,0:DRAW590-X%,0:SOUND
0,-10,5,1:NEXT
 840 SOUND0,-15,6,10
 850 RESTORE1090
 860 REM * 10 fireballs *
 870 VDU23,224,24,126,126,255,255
,126,126,24
 880 FORF=1TO10
 890 VDU4:COLOUR1:PRINTTAB(7,0)"L
ASERBUG FIREWORK DISPLAY"
 900 REM * Bring up noise *
 910 FORX%=0TO255:SOUND&10,-15,7,
1:SOUND&11,0,X%,0:NEXT
 920 REM * Fireball move *
 930 READC:VDU19,2,C,0;0;
 940 SX=640:SY=350
 950 SXS=5*((RND(3)-2)*RND(5)):SY
S=5*RND(5)
 960 SX10=100-RND(200):SY10=100-R
ND(200):SX20=100-RND(200):SY20=100
-RND(200)
 970 VDU5:SOUND0,-10,6,255
 980 REPEATGCOL0,2:MOVESX,SY:PRIN
TCHR$224
 990 MOVESX+SX10,SY+SY10:PRINTCHR
$224
 1000 MOVESX+SX20,SY+SY20:PRINTCHR
$224
 1010 GCOL0,0:MOVESX,SY:PRINTCHR$2
24
 1020 MOVESX+SX10,SY+SY10:PRINTCHR
$224
 1030 MOVESX+SX20,SY+SY20:PRINTCHR
$224
 1040 SX=SX+SXS:SY=SY+SYS
 1050 UNTILSY>1024ORSX<0ORSX>1280
 1060 SOUND&10,-15,6,4:NEXT
 1070 TIME=0:REPEATUNTILTIME=50
 1080 VDU4:COLOUR1:PRINTTAB(7,0)"L
ASERBUG FIREWORK DISPLAY"
 1090 DATA2,3,1,2,3,1,2,3,1,2
 1100 FORX=200TO0STEP-2
 1110 MOVE0,X:PLOT7,1280,X
 1120 SOUND0,-X/13.4,4,1:NEXT
 1130 TIME=0:REPEATUNTILTIME=200
 1140 CLS:COLOUR1:PRINT"GOODBYE"
```

instant graphics

This article is all about how to make graphic displays (and text to a lesser extent) appear instantly. Let us start with a simple program:

```
10 MODE4
20 MOVE100,100:DRAW1000,100
30 PLOT85,1000,1000:DRAW100,100
0
40 PLOT85,100,100
```

All it does is draw a simple square using PLOT85. This is all very good but you can see that the square is drawn using two triangles because of the speed that PLOT85 works at. If the program is only for your own use this does not matter but if you are trying to give the program a professional touch it is unacceptable. Try adding the following two lines to the program:

```
>15 VDU19,1,0;0;
>50 VDU20
```

As you can see although there is a delay of $\frac{1}{2}$ second before the picture appears the overall effect is much better.

The method used is by turning off the colours (i.e. making them black —the same as the background colour) using VDU19, drawing the picture and then restoring it using VDU20. On OS 0.1 you could use VDU19,colour,?906,0;0; to set colour to off (i.e. background). For examples of this method see TV Test Signal Generator elsewhere in this issue. Also some of the Acornsoft games use this, i.e. Monsters.

Now you have a method of drawing instant pictures but is this all the method can be put to? The answer is no—animation is made possible as well. Try the following program:

```
>

10 ONERRORVDU20:END
20 MODE2:VDU23;8202;0;0;0;
30 FORZ%=1TO8:MOVE640,1024
40 VDU19,Z%,0;0;:GCOL0,Z%
50 FORX%=0TO360STEP30
60 DRAWSINRAD(X%)*(510/Z%)+640,
COSRAD(X%)*510+512
70 NEXT:NEXT
80 FORX%=1TO8:VDU19,X%,6,0;0;
90 A=INKEY(10):VDU19,X%;0;0;
100 VDU19,X%;0;0;:NEXT
110 FORX%=8TO1STEP-1
120 VDU19,X%,6,0;0;:A=INKEY(10)
130 VDU19,X%;0;0;:NEXT
140 GOTO80
```

What this does is to produce a pulsating circle. When you press ESCAPE you can see how it is done. Eight shapes are drawn all in different colours (all turned off) and then each colour in turn is changed to the colour and then turned off again.

If instead of drawing shapes you drew a proper picture and used the same method of turning on one picture at a time decent animation becomes the possibilities.

The possibilities of these methods are huge. Several of the programs in this month's magazine use this technique. I could go on for a considerable time about the method in much more detail and with several more programs but if I did you would have nothing to do! Try experimenting with the method and let us know what you come up with.

Paul Barbour

Dear LASERBUG,

Have you or anybody you know come across a problem of wavering and oscillating graphics and characters that get better or worse as the micro heats up but never stopping at a particular point. This only happens on UHF output or BNC output *with* colour. BNC composite video Black and White is perfect and not affected nor is RGB through a Microvitec monitor.

The Obstacle Course game from the first issue of Personal Software shows the fault to extremes, but I find MODE1:COLOUR2: PRINT"HHHHH" and MODE7:PRINTCHR$(&93)"HHHH" also shows the problem. Although I have not seen the characters put through any other micros, I have seen the Obstacle Course program put through 3 Model B's with the same fault as well. I enclose the tape for your opinion:

(1) Do you get moving bars when played through a TV UHF socket (green and orange bars)

(2) If you do, any ideas on how to stop it?

(3) If you do get bars are your characters stable (i.e. H's as described above) for more than a couple of minutes (once again put through a TV via UHF aerial socket).

Your consideration to this problem would be *much appreciated* as RCS, Acorns repairers can't work it out except "It happens on another machine as well" and Acorn's Robin Payne—"what do you expect via a TV, you can't expect non-oscillating characters"—I'm sure Paul you would agree this is rubbish—perhaps you can help me prove it and solve this problem. We both know the limitations of TV's but surely my problem is not one.
DAVID GLEW,
Beckenham,
Kent.

*REPLY: Well David, I'm sorry to disappoint you but I feel you don't quite understand the limitations of TV's after all. Your fault does occur on my TV as well and this does not really surprise me. The PAL system does have some limitations—on viewing normal TV pictures these go un-noticed but when using it to view your computer the problems are apparent, the moving diagonal green and orange bars are typical. To stop the oscillating to some extent you could try turning off the interlace by using *TV0,1 but I doubt if this will help you much. Even if you tried playing around with your tuning I doubt if you could get rid of the moving bars. As we described in last month's Programmers Corner you could try slowly adjusting the trimmer located in the PAL encoder of the micro. It sits about 1½" away from the UHF modulator and seems to drift a little with age. This will only minimise your problem. The reason the Obstacle Course program shows up the fault so much is because it uses a checkered character which stretches PAL to its limits. I know it isn't much help but you honestly can't expect too much from your poor old TV. I suppose one answer could be to buy either a pure monitor or a converted TV/monitor. We do review a pure RGB monitor in this issue and reviews of converted TV/RGB's will be appearing in later issues. I am only sorry I couldn't be of more help.*

Paul Barbour

printer review

For the second month running we have meant to be reviewing the Amber 2400 printer and for the second month running Amber have been unable to supply us with one. We were, however, contacted by the managing director of Amber, Mr. Rayner, who told us that there were some production problems with the original batch but they are trying to reduce their backlog as soon as possible. However, they have not reduced it sufficiently to be able to get one to me by the copy date. Hopefully a full review of the Amber 2400 will appear next month? (Even though that is what I said last month!)

Paul Barbour

soft review

PROGRAM: 3D-Maze
REQUIREMENTS: 32k
SUPPLIER: Sinclair (I.J.K. Software), 55 Fitzroy Rd., Bispham,
Blackpool, Lancs.
PRICE: £3.95
DESCRIPTION OF PROGRAM: Firstly I think I should explain the 3D aspect of this game. You are only trying to solve a two-dimensional maze but you can look at the maze as if you were in it giving a 3D display. The maze you are solving is a square and its length varies between 3 and 29 squares long, depending on what you feel competent to solve. You can move between two displays—an overhead view of the maze (2D) from which you can judge your position and the normal display which is what you would see if you were inside the maze (3D). The game is quite enjoyable once you have got the hang of the direction controls and its price makes it good value for money.

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

---oOo---

PROGRAM: Row-of-4
REQUIREMENTS: 32k
SUPPLIER: Software for All, 72 North Street, Romford, Essex
PRICE: £6.45
DESCRIPTION OF PROGRAM: Row-of-4 as the name implies is Software for All's version of Connect-4. It is a very well written program and incorporates some clever game strategies which makes it very hard to beat. No skill levels are offered unfortunately which means it is only suitable for someone who is good at the game. Graphics are used to good effect which makes this the best version of the game that I have seen so far.

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

---oOo---

PROGRAM: Missile
REQUIREMENTS: 32k
SUPPLIER: CJE Microcomputers, 25 Henry Avenue,
Rustington, West Sussex BN16 2PA
PRICE: £9.00
DESCRIPTION OF PROGRAM: This game is the first implementation of Missile Command on the BBC Micro. It is written very much to be an arcade-style program, not only in the game itself but in the way the display acts in between games. The controls are very close together which makes the game hard to control until you are used to it. Once you can control your sight properly the game becomes a close representation of the proper arcade version. The game is very good but at £9 it is overpriced.

PRESENTATION: ★ ★ ★ ★
ADDICTIVE QUALITY: ★ ★ ★
USE OF GRAPHICS: ★ ★ ★
VALUE FOR MONEY: ★ ★

---oOo---

PROGRAM: Kingdom
REQUIREMENTS: 32k
SUPPLIER: Computer Shop (Bucon Ltd., '42' Software),
18 Mansel Street, Swansea SA18 5SG
PRICE: £7.95
DESCRIPTION OF PROGRAM: This program's full name is Curse of the Middle Kingdom, and with a name like that it couldn't be anything

but an adventure program. Unlike most of the adventure programs currently available it is not text but graphics based. It does use MODE7 but even still does have understandable graphics. The scenario is that you must explore the kingdom to try and recover 10 treasures which must be placed in sanctuary while battling a variety of monsters with different strengths and magical powers. You have a choice of 5 characters and 4 weapons plus 7 overall skill levels. The game is quite good and varied but rather excessive in fights. For an adventure the price is reasonable. The game loads in two parts with the first part simply giving extensive instructions.

PRESENTATION: ★ ★ ★
COMPLEXITY: ★ ★ ★
RESPONSE SPEED: ★ ★ ★
VALUE FOR MONEY: ★ ★ ★

---oOo---

PROGRAM: Fun Games (Breakout, Dodgems, Flash!, Snake)
REQUIREMENTS: 16k
SUPPLIER: BBC Soft (copyright Acornsoft),
                  35 Marylebone High Street, London W1M 4AA
PRICE: £10.00
DESCRIPTION OF PROGRAM: This is one of the long-awaited program packs for the BBC. It is very nicely presented in the same style as the Welcome tape. One disappointing thing is that rather than being independent of Acorn the copyright does in fact belong to them. There are four games in this pack and all but one uses MODE7. Breakout is the same program that you would have on the Computer Programme. Dodgems is a race game in which you must move around a track eating up dots whilst avoiding a computer-controlled car—this game uses MODE5. Flash! is a computerised version of Simon, whilst Snake is another implementation of this classic game. All of the programs are very well written and perform without a single fault. However with all of them there is something missing—I can't track it down completely but I think the games simply lack a bit of imagination. I know that there isn't a great deal that can be done to standard games such as these but I honestly was rather disappointed with them. I much prefer the pure Acornsoft programs. We will however, be looking at other programs from the Beeb in later issues. The programs are good value for money though, as in real terms you are paying £2.50 per program.

PRESENTATION: ★ ★ ★
ADDICTIVE QUALITY: ★ ★
USE OF GRAPHICS: ★ ★ ★
VALUE FOR MONEY: ★ ★ ★

---oOo---

PROGRAM: Cards
REQUIREMENTS: 32k
SUPPLIER: Micro-Aid, 25 Fore Street, Praze-an-Beeble,
                  Camborne, Cornwall TR14 0JX
PRICE: £2.95
DESCRIPTION OF PROGRAM: This game is meant to simulate Bruce Forsyth's card game program "Play Your Cards Right" in as much as the higher/lower part. "Exciting graphics, colour and sound with game" the publicity material says. The graphics are boring, the colour is boring and the sound is—you guessed it—boring. The game is sleep-enducing to put it mildly and is not even worth the effort of loading in. Yuk, is my opinion. At £3 it is cheap although I wouldn't take it if they gave me the £3.

PRESENTATION: ★ ★
ADDICTIVE QUALITY: ★
USE OF GRAPHICS: ★ ★
VALUE FOR MONEY: ★

PROGRAM: Defender
REQUIREMENTS: 32k, 6522 VIA chip
SUPPLIER: Acornsoft, 4a Market Hill,
                  Cambridge CB2 3NJ. (0223-316039)
PRICE: £9.95
DESCRIPTION OF PROGRAM: Wow!!! That is the only way to describe this fantastic game which is just like the arcade version. It takes a good hour to get used to the display and the commands but once you do you can play the best game I have ever seen written for the Beeb micro. The controls are varied and cover 6 keys in several different areas of the computer which can leave you fumble-fingered quite easily. The program uses excellent graphics and really does use the computer to its full capacity. Forget the rest, make this the next program that you buy.

PRESENTATION: ★ ★ ★ ★
ADDICTIVE QUALITY: ★ ★ ★ ★ ★
USE OF GRAPHICS: ★ ★ ★ ★ ★
VALUE FOR MONEY: ★ ★ ★

---oOo---

PROGRAM: American Trader
REQUIREMENTS: 32k
SUPPLIER: R. Steggles, 79 Glenlyon Rd., Eltham,
                  London SE9 1AL
PRICE: £6.95
DESCRIPTION OF PROGRAM: This program is another adventure. However instead of being a complete fantasy it is based around an American truck driver with a large debt. You have to try to get the debt down by trading with people in 6 different towns. The game is mainly in MODE7 although for moving between towns MODE5 is used for a better graphical display. This program is a classic example of the difference between amateur programs and professional ones. The game itself is sound but there are so many features that could be simply added to make the game foolproof. The string handling commands could be utilised to ensure better error handling (see String, String, String elsewhere in this issue) and in places the cursor could do with being turned off. The game itself is OK but it could do with a good hour of alterations by a professional to make it better—this should not however normally be necessary for a program that is being marketed.

PRESENTATION: ★ ★ ★
COMPLEXITY: ★ ★ ★
VALUE FOR MONEY: ★ ★

---oOo---

PROGRAM: Landfall/Serpent
REQUIREMENTS: 32k
SUPPLIER: GT Software, 8 Bull Street, Potton, Sandy,
                  Bedfordshire SG19 2NR
PRICE: £6.50
DESCRIPTION OF PROGRAM: The first program on this cassette is Landfall which is a Lunar Lander type game—don't groan though as it is approached from a different angle (in more ways than one!). Instead of watching the spacecraft land from the outside you have a head-up display with a window so you can see outside and a variety of controls. To start off with you just try to land by varying the thrust of your spacecraft but once you have mastered that you can try to land at the correct landing site whilst taking into account a strong wind. This game was very interesting although hard and makes compulsive playing. Landfall makes the tape worth buying on its own. Serpent is not another version of Snake, you will be pleased to hear. The game uses teletext graphics completely—the idea is that you have to take so many eggs away from a serpent who is trapped in a kind of maze. Every time you take an egg a hole appears in the maze which only the serpent can go through. One touch from the serpent and you are dead! This game is OK but not one that interests me too much, especially in relation to the other program.

	Landfall	Serpent
PRESENTATION:	★ ★ ★	★ ★ ★
ADDICTIVE QUALITY:	★ ★ ★ ★	★ ★
USE OF GRAPHICS:	★ ★ ★ ★	★ ★ ★
VALUE FOR MONEY:	★ ★ ★	

---oOo---

EDUCATIONAL PROGRAMS: Shape and Race
REQUIREMENTS: 16k
SUITABLE FOR: Ages 6 +
SUPPLIER: H & H Software, 53 Holloway, Runcorn,
Cheshire WA7 4TJ
PRICE: £5.50
DESCRIPTION OF PROGRAMS: The first program on this tape is
meant to aid in shape recognition ranging from a square to a heptagon. It
is a competitive game for 2 players which makes it more interesting as far
as children are concerned although it can lead to wrong answers
occasionally just to beat the child's opponent. On the screen there is a big
box with a dot bouncing around leaving a trace. In the middle of the box
is an invisible shape which can be one of 13. Not only does the dot bounce
off the edges of the box but also the invisible shape. Hence you are left
with an image in the centre of some kind of shape. The children have to
guess what the shape is within about 100 seconds. It is fairly hard to do as
the shape can be hard to pick up but even so it does perform its task well.
You all must have played at one time the game of Nim—there is a pile of
stones and you may take between 1 and say, 5 at once. You take it in turns
with the computer to take the stones and the one who takes the last stone
wins. The theory is the same for this game but instead of stones you play a
race game re-enacting the Hare and the Tortoise. The game is fine for
young children once they have got the idea of it and assists them in simple
addition. Both games are well written and the second one has informative
instructions well written for children.

	Shape	Race
PRESENTATION:	★ ★ ★	★ ★ ★ ★
FOR AGES:	8 + (in my opinion)	6 +
SUBJECT:	Maths-geometry	Maths-addition
USEFULNESS:	★ ★ ★	★ ★ ★
NUMBER OF PLAYERS:	2	1
MONEY:	★ ★ ★	

---oOo---

BUSINESS PROGRAM: Alphabeta
REQUIREMENTS: 32k, Seikosha GP-100A Printer
(versions or alterations for other printers are
available on request)
SUPPLIER: H & H Software, 53 Holloway, Runcorn,
Cheshire WA7 4TJ
PRICE: £28.50
DESCRIPTION OF PROGRAM: This is the first wordprocessing
program that is available for the BBC Machine beating other such
programs for various reasons. I would like to digress a moment and make
a comment about wordprocessing programs in general. When writing a
program such as these, the actual work involved is no harder than, say, a
game that retails at £5. However because it can have the name
wordprocessor added to it, that gives the distributor a free licence to
charge 3 times the rate he would for a normal program. Hence to take this
program as an example, the highest price for any of this company's other
programs is £8.50, however Alphabeta is £20 more expensive than this. I
am not trying to put down H & H Software because this is common
practice but I simply wish people would start trying to sell programs at
reasonable rates. This particular program is fairly good although the

choice of printer it is designed for is unusual. It loads in 3 parts and has a
maximum capacity of 80 columns × 224 lines. Text can be edited on
screen, saved to tape or printed out. The screen editing is fairly
comprehensive including delete (which also closes the gap) and insert,
move lines up and down and centralising with up to 10 tabs. Two
methods of saving on tape are offered—bulk and byte. For printing the
computer asks you several questions about what type of printer you are
using. Up to 999 copies of the same document can be made with right-
justification made possible if you wish to give letters a professional look.
A manual is given with the program and although I only had the
provisional copy it does give you all the information necessary. The
program did work well and once you are used to it, writing letters etc.,
becomes very easy. I was impressed with it in general and would buy it
myself despite its price.

PRESENTATION: ★ ★ ★
DOCUMENT SIZE: 80 × 224
SCROLLING SCREEN: Yes
RIGHT JUSTIFICATION: Yes
SAVE FILES ON TAPE: Yes
EASE OF USE: ★ ★ ★
PERFORMANCE: ★ ★ ★
VALUE FOR MONEY: ★ ★ ✱

---oOo---

I would like to thank I.J.K. Software, CJE Microcomputers, Micro-
Aid, R. Steggles, GT Software and H & H Software for supplying us
with their products. The BBC Soft games pack was supplied to us by an
independent source.
Paul Barbour

This month we present an article on how to make back-up copies of programs that you otherwise would have thought impossible to copy. We are not trying to encourage piracy and Phil explains this clearly below—he does not explain how to list unlistable programs. Next month we present an article on how to protect your software from other people looking at it.

"Software Protection" is a frequently used term amongst the writers of games programs. It exists totally to protect the writer of the program (or more precisely his profits) from what he regards rather pompously as the ocean of pirates out there who are just itching to bootleg his latest machine code wonder. What it does not do is to offer any protection at all to the person who spends a, sometimes, quite considerable sum of money on what turns out to be a very cheap and poor quality cassette which, nevertheless, is protected—thus rendering the making of a back-up copy impossible for the majority of people. There are some software houses who do not protect their software—these companies tend to sell their programs cheaply at, they believe, a level at which piracy would not be considered worthwhile. They also claim that their profits are unaffected due to the high volume of sales. These companies are to be applauded. What then can we do about the rest?

There are two methods of making a back-up copy of your program on cassette. The first method and the easiest, is to make a straight audio copy. The second method and perhaps the most satisfying, is to make a "protection defeated copy" download onto cassette from your micro.

If you wish to make an audio copy i.e. feed the recorded signal from the output of one recorder to the input of another, you will require, in addition to the pair of recorders, an interconnecting lead. For the most reliable results one of your recorders should have VU meters and record level controls, although they are not essential. The cassette recorder supplied by the BBC for use with the BBC Micro is the Ferguson 3T07. This has a DIN rec/play socket in addition to EAR, MIC and remote as is current practice. The other recorder that I use is an AKAI stereo cassette deck that is a part of my HI-FI system. Incidentally, if you own a 3T07, when using it with your micro connect it to the micro by the EAR, MIC and remote sockets and insert a 5kΩ resistor in series with the lead inside the 3.5mm Jack Plug that goes into the MIC socket. In the absence of this the recording amplifier within the recorder is overloaded by the output of the micro. Another useful tip, if I may digress a little further, is to connect a small switch across pins 6 and 7 of the 7-pin DIN plug that you insert into the cassette socket at the rear of the micro. I have glued this switch to the side of the plug and at the flick of a switch I have monitor control of the cassette recorder:
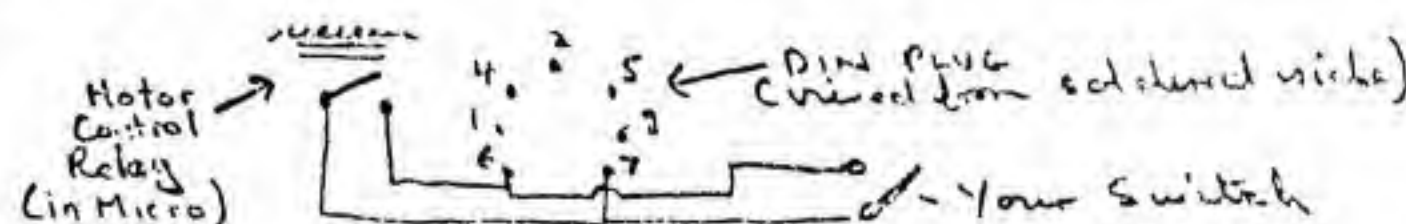
Back now to the subject of this article. Make up a copy lead, DIN to DIN in this case, as the signal levels tend to be more accurately defined there being a DIN standard.
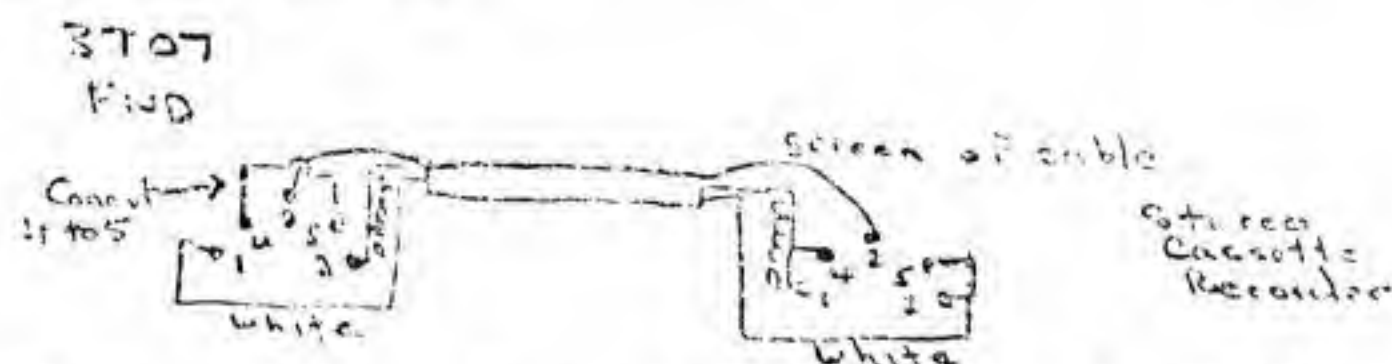
The conections are for a stereo deck. If you are using another Micro recorder such as another 3T07 then take the green wire to pin 1 only and the white wire to pin 3 only. In the case of the 3T07 you would have to connect 4 and 5 together as well. Having connected the two recorders together insert the cassette you wish to copy into the 3T07, which you will use as the source recorder. Insert a blank cassette that has been wound past its leader into your stereo deck and go E to E, i.e. depress the RECORD key only, so that the VU meters respond to the incoming signal. Press PLAY on the 3T07 and set the record level on the destination recorder to give you about 0 VU on both left and right

channels. When you are satisfied with the levels rewind the source recorder. The volume control of the 3T07 should have been left, in the above operation, at the level you use when loading into the micro, i.e. about half-way down the scale. Press PLAY and RECORD on the stereo deck and PLAY on the 3T07 and the program will be copied. Now try it by loading it into your micro. I have yet to find a program that you cannot copy by this method for either the BBC or the Atari machines.

On now to method two—downloading direct. Most programs of any consequence nowadays are written in machine code. They are often loaded by the command ★RUN having been recorded by the command ★SAVE"PROG"SSSS FFFF EEEE (open your user guide at page 392 and read section F carefully). Once loaded the computer jumps to the special execution address. To make a copy you need to know the program length, the start address and the execution address. To get this information do not ★RUN the program—★LOAD it instead. On completion of loading control will return to you and the program length will be indicated on the screen. Let's take a fictional program called ASTROCRASH as an example.

Type ★LOAD (ret) the program will load and on completion will report ASTROCRASH 2C 2C12. The 2C12 is the program length—write it down. Next you need the start address. This information is stored at locations 2F4 (low byte) and 2F5 (high byte). Type P.?&2F5. The response may well be E. Do the same for location 2F4. The response may be 0. Putting the high byte and low byte together gives 0E00. In this case "Page" (it usually is). The execution address is stored at locations 2F9 (high byte) and 2F8 (low byte). Peeking these locations will give you another hex number, say, 11 high byte, C3 low byte. Putting these two together gives 11C3 as the execution address. Try out your new-found knowledge. Type CALL&11C3 and the program you ★LOADed will burst into life. Press the BREAK key, type OLD(ret) and you will get the response Bad Program. Now make your back-up copy. Type ★SAVE"ASTROCRASH"E00+2C12 11C3 (see page 392 again) you will get the response 'RECORD THEN RETURN" just as you would normally. When the program has saved, try loading it in the recommended fashion, i.e. ★RUN etc. You now have a back-up copy. Certain programs you will notice are CHAINed. In some cases there are two or more programs in succession on the tape, the first loading the second, the second loading the third, etc. Do not be put off. ★LOAD (ret) and watch the screen, the first section of the program will load and the recorder will stop. Remove the cassette (don't rewind it or move the tape position) insert your back-up tape, peek the locations and ★SAVE it using the same filename as displayed on the screen. Remove your back-up tape (don't rewind etc.) insert the tape to be copied and ★LOAD the next section. Keep going until all sections are transferred. Try your back-up using the CHAIN"" command.

These methods work with all the software I have yet encountered and the information is being made available not in the expectation that everyone will be pirating and selling software, but in the hope that people will make back-up copies purely for their own protection.

Finally an appeal. If anyone knows how to make back-up copies of software on disc for Atari computers, I would be very interested.

Phil Hirst

Teletext counterpart. For instance all spaces in the program are replaced by ¼ and that alone makes the program hard to read. What if you want ¼? What you then have to do is put "double-bar" ¾.

Reaction about Prestel is mixed but the general opinion on it is that it failed whereas Teletext made it. The "re-launch" of Prestel is due very soon and hopefully that will increase the usefulness of the service. It is indeed more expensive but offers a much larger service than Teletext. It will probably always have more telesoftware on it than Teletext although the extent to which telesoftware will appear on Ceefax is as yet unknown.

We will try to keep you up to date with information on Prestel and Teletext with regards to telesoftware and the BBC Micro.

competition 3

Once upon a time someone in LASERBUG wrote a program. The end of it appears below:

```
>LIST
32763 REM THIS IS THE PROGRAM
32764 REM THIS IS THE PROGRAM
32765 REM THIS IS THE PROGRAM
32766
32767
     0 REM THIS IS THE COPYRIGHT
>
```

Now this person only has OS EPROM 0.1. The competition that is being set to you is how is the last line produced? The listing is completely genuine and in case you are interested there is another line 0 at the beginning of the program.

Send your answer (if you have one) to us at the usual address marking the envelope COMPETITION. The closing date is 15th December.

oddspot

The *Oddspot* program this month comes from Henry Connell of Pinner, Middlesex. It is longer than usual but we thought you wouldn't mind a little bit of extra keyboarding.

```
>L.

  10 MODE0:VDU23;8202;0;0;0;
  20 VDU19;6;0;19;388;0;
  30 X=1280:Y=0
  40 REPEATMOVEX,0:DRAW0,Y
  50 X=X-25.6:Y=Y+20.48
  60 UNTILX<=0:X=0:Y=1024
  70 REPEATMOVEX,1024:DRAW1280,Y
  80 X=X+25.6:Y=Y-20.48
  90 UNTILY<=0:X=1280:Y=1024
 100 REPEATMOVEX,1024:DRAW0,Y
 110 X=X-25.6:Y=Y-20.48
 120 UNTILX<=0:X=0:Y=0
 130 REPEATMOVEX,0:DRAW1280,Y
 140 X=X+25.6:Y=Y+20.48
 150 UNTILY>=1024:REPEATUNTIL0
```

If you have a short program (around 10 lines) that performs something "impressive" in either graphics or sound then please send it to us, marking the envelope ODDSPOT. A small prize goes to all programs we publish.

wordprocessor update

Firstly I must apologise to all those people who tried last months's program and found at times it wouldn't work. What you may have found is that once you have completed a screen and try to copy it out the program stops altogether. You might have also found a cure for this was to press f3 at the beginning. The reason for this was I was using the LASERBUG peek routine (from Issue 3) for both the f3 function and to dump the screen. When I dumped the screen I forgot to set the accumulator (A%) to 135 although the f3 key did have this. The listing below does cure this (line 1135).

The remainder of the listing below performs an added function to our wordprocessor on function key 7 (which was previously unused). This is another kind of delete key, the difference being when you delete something the rest of the line closes in.

We would welcome any comments you have on this program as well as any added features you have thought of or even written.

Please RENUMBER the program after you have made the alterations below otherwise any future additions will not make sense to you. Also remember to change the version number to 1.2.

Paul Barbour

```
>L.255
  255 *KEY7!!!!N
>L.1135
 1135 A%=135
>L.1510,
 1510 DEFPROCback_space
 1520 A%=135
 1530 X=POS:Y=VPOS:VDU9
 1540 FORX1=X+1 TO79
 1550 L(X1)=(USR(&FFF4)AND&FF00)/&
100:VDU9
 1560 NEXT
 1570 PRINTTAB(X,Y);
 1580 FORX1=X TO78
 1590 VDUL(X1)
 1600 NEXT
 1610 VDU32,32
 1620 ENDPROC
```

This article is all about string handling in one form or another. Twenty commands are described below and the only real assumption made is that you understand string variables (p.63). All page references that are made refer to the new User Guide.

Firstly let us define a string (A$) to be LASERBUG:

```
>A$="LASERBUG"
```

If you then enter:

```
>PRINTA$
LASERBUG
```

the word LASERBUG simply appears at the beginning of the next line. If you wanted a blank line before and after the word then you have to add a ' which generates a new line (like pressing RETURN):

```
>A$="LASERBUG"
>P. 'A$'

LASERBUG

>
```

If you wanted the word moved away from the edge by 10 characters you could redefine A$ thus:

```
>A$="          LASERBUG"
```

but this method is rather wasteful. A much better way is to say:

```
>PRINTTAB(10)A$
          LASERBUG
```

TAB stands for tabulator such as you would find on a typewriter. Its method of operation is the same and starts the word 10 character places away from the edge of the screen. SPC is similar to this command and further information on both can be found on pages 360 and 354 respectively. A screen is divided up into "fields" and in a MODE7 screen there are 4 fields which start at character positions 9, 19, 29 and 39. You can start print directly onto these fields by using the comma. For instance:

```
>PRINTA$,A$,A$
LASERBUG   LASERBUG   LASERBUG
```

The first word is printed at the edge of the screen and the following ones at the beginning of each appropriate print field (see p.68). These can be altered by changing the variable @% but that is out of the scope of this article. If you use a semi-colon after a variable no new line is generated and hence the cursor continues at the very next print position:

```
>PRINTA$;
LASERBUG>
```

The above descriptions are all very brief and I make no apologies for this. They are all straightforward and need little explaining. Should any further information be required this should be obtained from the manual.

Instead of assigning a string in a program you can get the user to input it directly:

```
>INPUTA$
?LASERBUG
```

You can also enter several pieces of data at once by separating the answers with commas:

```
>INPUTA$,B$,C$,D$
?LASERBUG,BEEBUG,ACORN USER,BEEBON
```

However, supposing you want to enter the whole line as one string:

```
>INPUTA$
?LASERBUG,BEEBUG,ACORN USER,BEEBON
>PRINTA$
LASERBUG
>
```

As you can see the computer still thinks you are trying to enter several pieces of data at once and assigns A$ to only the letters before the commas. Hence you must use another form of INPUT called INPUTLINE. This enters everything that you type in:

```
>INPUTLINEA$
?LASERBUG,BEEBUG,ACORN USER,BEEBON
>PRINTA$
LASERBUG,BEEBUG,ACORN USER,BEEBON
>
```

Instead of waiting for something to be entered and RETURN pressed you can just get one character from the keyboard. This can be done firstly with a command called GET$. The command takes the form A$ = GET$ and the computer will wait until a key is pressed. Whatever key is pressed will appear as A$. The other version of this command is INKEY$(time). This will wait for "time" hundredths of a second for a key to be pressed i.e. A$ = INKEY$(100)—would wait for one second.

This is again very brief and is taken to be "common knowledge". The remainder is what this article is really about.

First we will look at the string handling commands LEFT$, RIGHT$, MID$, LEN and INSTR. Most computers have all of these commands (except INSTR which is only normally found on much larger computers).

To demonstrate these functions we need to define a string:

```
>A$="BBC MICROCOMPUTER MODEL B"
```

The syntax for LEFT$ is B$ = LEFT$(A$,number). What this function does is to look at A$ and give "number" characters of this string. For example supposing we want to get the word "BBC" out of the above string. This is 3 characters away from the left of the word:

```
>PRINTLEFT$(A$,3)
BBC
```

RIGHT$ performs the same function except from the other side. For example "MODEL B" is 7 characters away from the right of the string and so:

```
>PRINTRIGHT$(A$,7)
MODEL B
```

MID$ is similar to these functions but acts from the MIDdle of the string. The syntax for this command is B$ = MID$(A$,position, number_of_letters). The position is the number of characters away from the left hand edge to start at (the first position is 0) and "number_of_letters" is the number of characters to be taken from this point. To get the word "MICROCOMPUTER" this starts at the 4th position (5 characters along) and is 13 letters long itself. The command required is:

```
>PRINTMID$(A$,4,14)
MICROCOMPUTER
```

With these three commands you can dissect any string you want and to good effect. Now supposing you wanted to know the LENgth of A$. All you need to enter is:

```
>PRINTLEN(A$)
25
```

INSTR is a very powerful command and on other micros a routine has to be written to do this. The command stands for IN STRing and what it does is search for a small string inside a larger one. Supposing with our string we wanted to find out whether it was referring to a Model A or B. What we could do is use A$ as the main string and search whether the words "MODEL A" are present in it:

```
>PRINTINSTR(A$,"MODEL A")
0
```

(There are easier ways to do this). What INSTR actually gives is the position of the smaller string inside the larger one. If you wanted to find out where the word "MICRO" started in A$ you would use:

```
>PRINTINSTR(A$,"MICRO")
         5
```

Note that when using this command the first position is 1 and not 0 as with the other commands. For details on these commands individually see p.283/4, 341, 298, 285/6 and 280/1. Also note that there is a bug in the current operating system when using INSTR in some cases (see p.281).

All letters on computers are stored in its memory in numeric form with the numbers referring to a specific code known as ASCII. This ranges from the letter A which is coded as 65 to the letter Z which is 90 (symbols also have ASCII codes, full details are given on p.490/1). Codes less than 32 are special control codes and are irrelevant as far as this article is concerned. Using the ASCII code you can either print up a letter by referring to its code or find out its actual code by referring to its letter. We said above that the ASCII code for the letter "A" is 65. What we need to do is print the CHARacter string of this code:

```
>PRINTCHR$(65)
A
```

On the other hand if you wanted to know what the ASCII code of the letter "Z" was you could use:

```
>PRINTASC("Z")
         90
```

Let us suppose that for some reason you wanted to enter a number as a string in a program:

```
>INPUTA$
?10
>PRINTA$
10
>PRINTA$*A$

Type mismatch
```

This is fine but as is shown above you cannot perform any numeric functions on the string. If you want to turn a string that holds a number into a numeric variable you must use the command VALue:

```
>A=VAL(A$)
>PRINTA
         10
>PRINTA*A
         100
```

If A$ contains letters then the value given by VAL is 0:

```
>A$="LASERBUG"
>PRINTVAL(A$)
         0
```

This situation can be looked at conversely. Say we had a numeric variable A which held the number 173. Like the above case you cannot perform any string functions on it:

```
>INPUTA
?173
>PRINTA
         173
>PRINTLEFT$(A,2)

Type mismatch
```

To turn this into a STRing you use the function STR$:

```
>A$=STR$(A)
>PRINTLEFT$(A$,2)
17
```

STR$ is affected by the value of @% but again that is out of the scope for this article.

The final string handling function we are going to look at is EVAL although this is rather different from the ones above. Supposing in a program you wanted to enter an equation. On most other micros you have to change a line in the program to enter the equation but on the Beeb you can enter the equation as a string and make it give a value by using EVALuate. Supposing we had five different equations, all in X, and we wanted to know their value when X was 1 and 5. First we need a short program:

```
>L.

    10 INPUTA$
    20 X=1
    30 PRINT"When X=1, f(X)=";
    40 PRINTEVAL(A$)
    50 X=5
    60 PRINT"When X=5, f(X)=";
    70 PRINTEVAL(A$)
    80 GOTO10
>RUN
?SIN(X)
When X=1, f(X)=0.841470985
When X=5, f(X)=-0.958924274
?COS(X)
When X=1, f(X)=0.540302306
When X=5, f(X)=0.283662185
?TAN(X)
When X=1, f(X)=1.55740773
When X=5, f(X)=-3.38051501
?LN(X^2)*0.5
When X=1, f(X)=          0
When X=5, f(X)=1.60943791
?(LOG(X)*LN(X))*SIN(X)
When X=1, f(X)=          0
When X=5, f(X)=-1.07874074
```

Any equation can be entered and it can refer to variables in the program. The equation must of course have a correct syntax—this could be checked with the other string commands outlined above.

I have tried to be as brief as possible throughout the whole article as there is a great deal to explain. I hope I have not left you too confused about the use of the commands but the best way to understand them fully is to experiment.

Have fun experimenting and don't get too strung up!

Paul Barbour

what monitor

When you buy a computer that can be hooked up to a television you soon find that you become either frustrated or the most hated member of the family. Frustrated because the wife wants to watch Coronation Street while you want to finish off your Space Invaders program or the most hated member because you did finish off your program during her Coronation Street. Hence most people find a new viewing system i.e. a second TV or a monitor their top priority.

This month we take a look at pure monitors—one a black and white and one RGB. The Black and White monitor we are using is a 12 inch high resolution model made by Crofton of Twickenham. The RGB is the official BBC Monitor—the 14 inch Microvitec.

This article is aimed primarily at the novice and hence I will try to avoid all technical terms. Firstly let me explain the difference between a monitor and a normal domestic television. The television signal that your television receives is specially coded so that it may be transmitted over the air. When your television receives it it decodes it into its pure form so that a picture can be produced. Your computer can produce this signal in its pure form from the outset but a normal TV cannot understand this. Hence the computer codes the signal and sends it out through the UHF Out. Because of all the coding/decoding the picture produced is not very good as you can see. Of course the system is fine for normal television pictures but when you are using a computer you notice the limitations of this system. A monitor can accept the pure signal directly and so the computer gives out this pure signal through the video out. The BBC Micro as standard gives out this signal as Black and White only, although you can alter this to give colour (see LASERBUG Issue 1). A colour picture is produced by mixing three different colours—red, green and blue. An RGB output can control these colours directly and produces the clearest and brightest pictures. Most televisions have only a UHF Input although some now have a Video Input as well. You can also buy some TV's converted for RGB and we hope to be reviewing some of these over the coming months. The Video and RGB systems are in actual fact simpler than the UHF one because they need less circuitry. However as Video and RGB are nowhere as much in demand as UHF higher prices still remain.

A black and white monitor normally costs around £100 although you can obviously get it cheaper if you look around. If you want to be able to read MODE0 clearly and want the optimum definition then you should look for a high-resolution model. One question I am quite often asked is how do I tell if it is Hi-res? The first thing to look for is whether or not it can cope with 80-columns. If it can then it is bound to be Hi-res. Some people get worried when they read that a monitor can use 80 × 24 text as MODE0 is 80 × 32. If a monitor has a good enough resolution to cope with 80 characters then it is bound to be able to cope with 32 lines. Also you should remember that your BBC Micro has an overall resolution of 640 × 256. Most Hi-res monitors have a resolution of about 800 lines in the centre. Towards the edges this resolution is generally poorer— about 600 lines. This is all within acceptable limits.

With a black and white monitor you obviously lose colour but instead get six shades of grey. I say six because when using B/W the colours red, blue and black all come out as black. While we are talking about shades there are generally three different types of monochrome monitors available—Black & White, Green and Amber. The most common type is black and white and is preferred by many people. The green monitors, in my opinion, are easier to use over long periods and are my preference. The amber monitors look nice but cannot really be used for too long at once if you want to avoid eyestrain.

Monitors give a very clear and steady picture and hence it is normally advisable to turn off the interlace with ★ TV0,1. If you spend a great deal of time at your computer for certain applications, e.g. word-processing, then you are much better off with this kind of monitor. Generally I find the clearest pictures are obtained by setting the background colour to 2 and the foreground to 3. If however you are more interested in other applications of your BBC Micro, particularly ones that use colour, then you really need an RGB monitor.

The monitor we are looking at for this article is the official BBC Monitor. This is around £300. Unfortunately the resolution of this monitor is not up to standard but is fairly close and gives an acceptable MODE0 result. The result is much better if you reverse the screen, i.e. background 1, foreground 7. The quality of colour you get is unbelievable if all you have been used to is a normal TV. With the Microvitec you do need a ★ TV255 to move the picture down a bit but other than that it works fine. There is not really a great deal I can say about this monitor except that it produces good results. A pure monitor

such as this one will produce better results than a converted TV especially as far as resolution is concerned.

If you want to use your computer to its full capabilities then you need a good colour monitor such as the Microvitec. If you do a great deal of word-processing then a Black and White (or Green or Amber) monitor is essential. If you want a colour monitor but are not willing to spend £300 on something that cannot be used for normal viewing then you need a converted TV/RGB monitor. These normally can accept both signals although the RGB will not be as good as on a plain monitor. If you are happy with a poorer picture and if you can get a second colour TV into the bargain then any normal television will be fine, although the Sony range is arguably the best. A normal TV is likely to cost ½ the price of a converted one.

The other alternative open to you is converting your own TV to RGB. However THIS IS VERY DANGEROUS AND SHOULD ONLY BE ATTEMPTED BY A COMPETENT TV ENGINEER.

In the next few months we will be reviewing some converted TV's for you. Otherwise there was an article elsewhere in this month's magazine about converting receivers.

Paul Barbour

book review

Since the last Bookreview *no new books have appeared. Rather than have no review at all we decided to review a book which was published some months ago.*

30+ Programs for the BBC Microcomputer by Chris Evans
71 pages   £5.00 for the book, £4.00 for cassettes (2)
CJE Microcomputers

This book was published back in May of this year when the BBC Micro was still fairly new. Hence then, this book would have contained very useful programs and information. By the time you are reading this it will be November and you would have known the information in this book for several months. When it was written however the book would have been a great deal of use. The question is has the book anything to offer now?

The book contains 31 programs covering Educational, Games, Graphics, Music, Scientific and Utilities programs. These are also available on cassette for £4. As well as this there is an 8 page hints and tips supplement. As the main part of the book is the programs I will look at those first.

The first eight programs are all games. They all use the teletext mode, most of them with colour. The games are typical of those you would find on the older, less adaptable computers, e.g. PET. The games are all fairly simple but use some of the more advanced features of the BBC Micro to run. Using MODE7 none of the games are visually exciting but when you consider the time that the book was written they aren't too bad.

The next two programs are described as graphics/educational. They draw a pyramid/cube rotating about its axis. The following programs are two graphics ones that draw a spiral/circle in different colours and then rotates them. If you remember the sample screen dump we printed in the last issue the next program from the book draws this function only in two colours. Next is a program that creates patterns by drawing concentric rectangles. Feeling patriotic? The next program draws the Union Jack. The next two programs use the instant graphics methods (see elsewhere in this issue) to move lines and circles. There are two more graphics programs before we continue. One draws surprisingly good patterns by rotating a polygon without rubbing it out (something like a spirograph) and the other draws Moire patterns (similar to Persian Carpet from the User's Guide).

The next three programs all demonstrate sound. The first allows you to make up your own tunes, the next two play a couple of tunes—God

Save The Queen and The Sailor's Hornpipe. The two tunes are fine but not very exciting.

Mathematics is the topic for the three following programs covering Best Fit's, Integration (by summation) and Standard Deviations. Next come two utilities, a fast inder for cassette recorders without a counter and a soft key set-up program (see LASERBUG, Issue 2).

The next few are described as educational. First is a program called Screenplay, which must win the prize for the most pointless program of the year! Flashcards is the next one which simply puts words on the screen for the child to try and recognise. The third one adapts this method to form a spelling program. The final program is a simple file handling system using the CFS.

The other thing you get with this book, as mentioned previously, is a Hints and Tips Supplement. Here you are told about using a cassette recorder (and cassettes and leads), how to merge programs, how to use a printer and a loose sheet containing useful data for quick reference, i.e. VDU codes, tokens, character set and details of the screen modes.

In the book all program listings are well commented. On the tape all programs are recorded at both 1200 and 300 baud which is a feature I would recommend to all software houses.

The book itself is fairly well written and if I had reviewed it in May when it was written I would have thought much more of it. Now however, I am afraid that it is simply too out of date to contain anything new in it. The programs in it are still very good and I think if you are new to the BBC Micro this book might give you some help. Other than that, in all honesty, I cannot think of a use for it.

Paul Barbour

programmers corner

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 would grind to a halt if you didn't write to us so if you find out something that would make all our lives easier, 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.

Supposing you wanted to define A$ as I wish to subscribe to LASERBUG, the magazine for the BBC Micro. You couldn't just say:

```
>A$="I wish to subscribe to "LASER
BUG", the magazine for the BBC Mi
cro."

Syntax error
>
```

To put a quote mark inside a quotation you must specify " by putting "" i.e.:

```
>A$="I wish to subscribe to ""LASE
RBUG"", the magazine for the BBC
Micro."
>PRINTA$
I wish to subscribe to "LASERBUG",
  the magazine for the BBC Micro.
```

VDU statements can be a pain at times because they require so many values, e.g. VDU19 requires 6 values when only 3 are relevant:

```
>
    10 VDU19,2,5,0,0,0
```

Some people have found out that the last two zeros can be compacted but few people actually understand what they are doing:

```
>
    10 VDU19,2,5,0;0;
```

To take our example of VDU19, this requires 5 bytes after it, i.e. VDU19,b1,b2,b3,b4,b5. b1 is the colour to alter and b2 is the colour you are changing it to. Also a byte cannot be any higher than 255. As we explained in Issue 3 (★ FX Part II, p.9) the computer represents numbers greater than 255 in the form of high byte/low byte (N DIV256/N MOD256). We can use semi-colons to send two bytes at once which saves space. If this is all gibberish to you, let me explain all this in English.

Let us suppose we wish to compact the command shown below:

```
>VDU19,13,4,0,0,0
>P.~13
            D
>P.~4
            4
>VDU19,&40D;0;0;
>P.&40D
          1037
>VDU19,1037;0;0;
```

What you first have to do is convert the two bytes (b1,b2) into hex. Then you reverse their order (because the computer requires low byte first in this case), compact the final few zeros you end up with the penultimate command. It does not have to be in hex however as the final command shows. This technique of compacting VDU commands can be applied to any of them. VDU23 is the longest of them all and so below is an example applied to this:

```
>L.
    10 VDU23,224,255,35,67,37,53,14
3,235,85
>P.~255
            FF
>P.~35
            23
>P.~67
            43
>P.~37
            25
>P.~53
            35
>P.~143
            8F
>P.~235
            EB
>P.~85
            55
>L.
    10 VDU23,224,&23FF;&2543;&8F35;
&55EB;
```

David Wood of Harrogate writes that he is totally confused with text windows and would like a simple PROCedure to make the setting of them easier. Below is the PROCedure David required:

```
L.
    10 DEFPROCwindow(TX,TY,BX,BY)
    20 VDU28,TX,BY,BX,TY
    30 ENDPROC
```

As you can see there is little point in making up a procedure. David's main problem is that he doesn't understand the syntax of VDU28. If you remember that the window you are defining is a square, the first 2 values

for the PROCedure is the co-ordinates of the top, left-hand point and the other 2 the co-ordinates of the bottom, right-hand point, i.e:

(TX,TY)
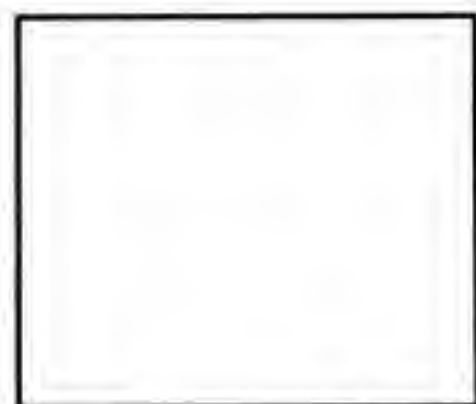
```
┌─────────────┐
│             │
│             │
│             │
│             │
│             │
└─────────────┘
       (BX,BY)
```

If you remember this then a PROCedure is unnecessary.

```
>_=3
£=9
```

Do the above statements look funny to you? If they do, don't despair. A fact you may not know is that the symbols __ and £ can be used as variable names. This fact is not of any startling significance but if you have a value that you want to hide away, who would think of looking at those two as variables?

Next is a little hint on using the joysticks (apart from all those elsewhere in this issue). One criticism of the joysticks is that they are too big and bulky for a child to handle. They should be used with the button facing towards you but try turning them around—the hand grip part is wrong but children will find them easier to use. (*Note:* The How To Use Joysticks article in this issue assumes that you have turned them around.)

The average person's computer system consists of a computer, monitor/TV and cassette recorder. Some people also have a printer. Hence most people tend to use a large gang-socket for their micro system. Also the same people tend to get confused when they pull out the socket for say the printer or even just turn it off and the computer suddenly crashes. Computers are very sensitive to their power supply and any sudden changes in this can cause a complete crash. This can normally be cured by a hard-break although the program is quite often corrupted beyond help. If you are in the habit of playing around with the power in any way (pulling out plugs/turning off peripherals) whilst using a gang-socket the computer should always be on a separate power supply (i.e. a different lead).

Following on from earlier about VDU statements, there is another way of making them shorter. As far as the computer is concerned 255 is identical to − 1, 254 is identical to − 2 and so on. Hence:

```
L.
    10 VDU23,224,255,255,255,255,25
5,255,255,255
```

can become:

```
>L.
    10 VDU23,224,-1,-1,-1,-1,-1,-1,
-1,-1
```

One final thing before we close *Programmers Corner* for this month. We are very grateful to all those people who send us in hints and tips but please make sure that the hints and tips sent in are genuine. To take one example, somebody wrote in to us about a "special" form of SOUND:

```
>SOUND0,-10,&93576,25
>P.&93576MOD8
         6
>SOUND0,-10,6,25
```

All this is is a multiplied version of channel 0,6 as you can see.

If you have any hints and tips that are original (i.e. have not already been published) about the BBC Micro, please let us know as this page will only work if we hear from you.

Paul Barbour

compiler review

The very first compiler for BBC BASIC was written by Jeremy Rushton and is published in the form of a ring-bound folder containing a listing of the program. It retails at £4.95 and the reason it is so cheap is that you have to type in the program yourself.

Here we are reviewing version 3.0 of this compiler which to my knowledge is the only one that has been published. The first thing when using this compiler is to type it all in which is no mean feat. The listing is 13.1k long and covers 605 lines! Once you have entered the listing the next problem is debugging it—not that the original listing has any errors in it but it is extremely unlikely you could type it all in without one mistake. Debugging takes almost as long (if not longer if you are not used to typing so much) as entering the program and it is essential in my opinion to use a printer to help you. Without one the task of debugging is made 10 times harder.

Once you have correctly entered the listing to enter your program you must change PAGE to a value above that of the compiler—&6000 was found to be a good place. The compiler is not able to compile normal BBC BASIC but only a subset of it.

The first difference you notice about the actual language is that no loops (FOR...TO...NEXT & REPEAT...UNTIL) are allowed. Below is a normal program for a loop and the version required by the compiler. However despite the actual number of commands the compiler version is 70% faster than the normal one. ★ FX commands are catered for as is VDU. Variables can only be integer numeric and a single letter with their range being between 0 and 65535. Pseudo-negative numbers are possible and this is explained in the instructions. POINT and RND are not available in their normal forms. POINT(X,Y) becomes (X)READ(Y) and RND(X) where the number required is between 10 and 50 becomes (10)RND(50). LET must be used at all times and a "not equal to" comparison is available using a hash. One extra command called OFF is added to the language to turn the cursor off.

The other problem with this compiler is its size. Because it uses 13k + memory for your program + memory for the compiled version of the program your use of MODEs is very limited. MODE7 is easily available but you start having problems with any others.

Despite these drawbacks the compiler does work very fast. The machine code the program generates is not very compact but as the instructions say "...neither the compiler itself nor the code it generates are particularly compact. It is very fast, though, and I think this makes up for it."

The compiler does work well and it does work fast but its real use I cannot see. The capability for graphics is very limited as the use of screen modes is very limited. As far as mathematics are concerned the numbers it can handle are just not sufficient to make it worthwhile. It is still well written and gives you an insight into the world of compilers but the practical applications of it are limited. Also the thought of typing in a 13k listing must seem rather daunting to most people and so a cassette with the program on might be a good idea for the publishers to consider.

LANGUAGE: BBC Basic
SPEED: Very fast
LENGTH: 13k (listing only available)
PUBLISHERS: Interface Publications
PRICE: £4.95

```
>L.
    10 REM * COMPILER VERSION *
    20 LETA=0
    30 LETA=A+1
    40 IFA#1000THENGOTO30
    50 END
```

```
>L.
    10 REM * TEST PROGRAM *
    20 FORA=1TO1000
    30 NEXT
```

The program below generates TV Test Signals for the setting up of colour TV Convergence and linearity, as well as providing colour bars to aid the setting of chrome circuits. The program provides a crosshatch pattern and a dot pattern, these are used for the setting of static and dynamic convergence. A large circle is provided to enable correct setting of linerity. In addition to these and to stimulate flagging interest a less-boring-circle is also generated. In view of the very high cost of crosshatch generators and colour bar generators the program will soon prove its worth.

Phil Hirst

```
?LIST
   10 REM     TV TEST SIGNAL GENERAT
OR
   20
   30 REM       Written by Phil Hirst
   40
   50 REM      Modified by Paul Barbo
ur
   60
   70 REM            Version 2.1
   80
   90 REM               23/9/82
  100
  110 REM (C) LASERBUG - October 1
982
  120
  130 M%=7:GOSUB830
  140 PROCmenu
  150 *TV0,1
  160 M%=2:GOSUB830
  170 IFA$="C"THENPROCcross
  180 IFA$="D"THENPROCdots
  190 IFA$="B"THENPROCbars
  200 IFA$="R"THENM%=1:GOSUB830:PR
OCcircle
  210 IFA$="L"THENPROCless_boring_
circle
  220 IFA$="E"THENMODE7:END
  230 REPEATUNTILINKEY(-99):RUN
  240 DEFPROCcross
  250 VDU19,7,0,0,0,0
  260 FORX%=32TO1248STEP64
  270 MOVEX%,0:DRAWX%,1024
  280 NEXT
  290 FORY%=32TO992STEP64
  300 MOVE0,Y%:DRAW1279,Y%
  310 NEXT
  320 VDU20
  330 ENDPROC
  340 DEFPROCdots
  350 VDU19,7,0,0,0,0
  360 FORX%=32TO1248STEP64
  370 FORY%=32TO992STEP64
  380 MOVEX%,Y%
  390 DRAWX%,Y%+4
  400 NEXT:NEXT
  410 VDU20
  420 ENDPROC
  430 DEFPROCbars
  440 X%=0
  450 DATA7,3,6,2,5,1,4,0
  460 REPEATREADcolour%:GCOL0,colo
ur%
  470 VDU19,colour%,0,0;
  480 MOVEX%,1024
  490 MOVEX%+160,1024
  500 PLOT85,X%,0
  510 MOVEX%+160,0
  520 PLOT85,X%+160,1024
  530 X%=X%+160
  540 UNTILcolour%=FALSE:VDU20
  550 RESTORE:ENDPROC
  560 DEFPROCmenu
  570 PRINT" "CHR$157;TAB(6)CHR$14
1;CHR$129"TV Test Signal Generator
" " "CHR$157;TAB(6)CHR$141;CHR$129
"TV Test Signal Generator"
  580 PRINT" "CHR$157'" "CHR$157;T
AB(11);CHR$132;"(C) LASERBUG 1982"
' " "CHR$157'" "CHR$157;TAB(9);CHR$
133;"Written by Phil Hirst"
  590 PRINTTAB(2,7);CHR$129;"Press
:";CHR$134;CHR$136;"'C'";CHR$137;"
for crosshatch"TAB(9,8);CHR$130;CH
R$136;"'D'";CHR$137;"for dots"TAB(
9,9);CHR$135;CHR$136;"'B'";CHR$137
;"for colour bars"TAB(9,10);CHR$13
2;CHR$136;"'R'";CHR$137;"for circl
e"
  600 PRINTTAB(9,11);CHR$133;CHR$1
36;"'L'";CHR$137;"for less boring
circle"TAB(9,12);CHR$131;CHR$136;"
'E'";CHR$137;"to end"
  610 *FX15,1
  620 C$="CDBRLE":REPEATA$=GET$:UN
TILINSTR(C$,A$)>0
  630 PRINTTAB(8,15)CHR$131;CHR$15
7;CHR$132;"To exit all options  ";
CHR$156;'TAB(8)CHR$131;CHR$157;CHR
$132;"press the SPACE BAR  ";CHR$1
56
  640 PRINTTAB(0,18);CHR$134;CHR$1
57;CHR$132;"NOTE: You may have to
wait a second  ";TAB(0,19);CHR$134
;CHR$157;CHR$132;" or two for the
pictures to appear  "
  650 TIME=0:REPEATUNTILTIME=500:E
NDPROC
  660 DEFPROCcircle
  670 VDU19,7,0,0,0,0
  680 VDU29,640;512;:MOVE0,512
  690 FORR%=0TO360STEP8
  700 DRAWSINRAD(R%)*511,COSRAD(R%
)*511
  710 NEXT:VDU20:ENDPROC
```

```
   720 DEFPROCless_boring_circle
   730 colour%=RND(7)
   740 VDU29,640,512; :MOVE0,512
   750 FORI%=0TO360STEP8
   760 GCOL0,colour%:VDU19,colour%,
0,0,0,0
   770 DRAWSINRAD(I%)*511,COSRAD(I%
)*511
   780 DRAWSINRAD(I%+8)*511,COSRAD(
I%+8)*511
   790 DRAW0,0
   800 colour%=colour%+1:IFcolour%=
8THENcolour%=1
   810 NEXT
   820 REPEATVDU19,RND(7),RND(7),0,
0,0:UNTILINKEY(-99):RUN
   830 MODEM%:VDU23;8202;0;0;0; :RET
URN
```

meeting place

We want to encourage the setting up of local groups under the general banner of LASERBUG. All it needs is for willing people to allow us to print their names and addresses (or telephone numbers) along with when their meetings take place to allow other people in turn to contact them with a view to either attending meetings or helping to organise them. It could be as small as a couple of people in someone's front room or a larger affair at the local church hall or school.

In case you are worried about lack of support from our 1982 Questionnaire 70% of people said that they would attend local meetings in their area, 27% of people said that they would probably attend and only 3% said that they were not interested. On a slightly more disheartening note you will probably end up doing a great deal of the work yourself as only 20% of people said that they would be willing to help organise things.

Generally we want to leave this up to you. Of course we are willing to give advice and help you where we can but the main idea is to get local groups functioning by themselves. We would like to see a whole network of local groups holding regular meetings so that anyone in possession of a BBC MICRO can get together with other owners/users wherever he lives.

We at LASERBUG feel very strongly that belonging to a local user group is the best way to get the most out of your computer. As well as meeting new friends you can get the most out of your computer. As well as meeting new friends you can get the benefit of other peoples discoveries—and share your own with them. If there is not one near you then why not start one yourself?

If you have started or are planning to start your own group then please let us know. Why not (assuming you have attended one of these meetings) write a short article on your experiences. We would be pleased to publish it on a Club Report page.

The present list of local user groups is:

John Claydon,	North London BBC Microcomputer Users Group and Education Workshop, Bounds Green Junior School, Park Road, N11. 01-889 5446. Meetings held on the second Sunday of each month at 2.00. Fee of approx. £1/meeting to cover costs.
G. W. Goodacre,	CHELMERBUG, 34 Quilp Drive, Chelmsford, CM1 4YA. Formal meetings held on the first Wednesday of each month at a local school. Informal meetings in-between in members homes.
Nick Lamb,	23 Gaywood Close, Caister-on-Sea, Great Yarmouth, Norfolk, NR30 5RD. 0493-728442.
Norman Lambert,	Orpington Computer Club, 11 Vinson Close, Orpington, Kent, BR6 0EQ. Meetings held every Friday evening at a local church hall.
Dr. P. Voke,	39 Ligo Avenue, Stoke Manderville, Aylesbury, Bucks. HP22 5TY. 0296-613974.

To be included in this list please write to us at our usual address giving full details and mark the envelope MEETINGS. a free affiliation scheme is available to all user groups and those interested should get in touch:

competition 1 results

Firstly I must thank all of you who made the effort to attempt to answer our competition. To refresh your memories the idea was to produce the most efficient program that printed out the Fibonacci sequence. Efficient was defined in terms of (i) size and (ii) speed and hence two different programs were asked for.

Unfortunately there was only one winner to the competition. We challenged you to beat 48 bytes in length and a run-time of less than 3.58 seconds. Out of the many entries Mr. A. D. Coombe of Ammanford, Dyfed, won the "size" part with the following program which was 46 bytes long:

```
>L.
    1 TIME=0:A=0:B=1:REPEATC=A+B:P
RINTC:A=B:B=C:UNTILC>1E38:PRINTTIM
E
>
>
>PRINTTOP-PAGE
        46
>
```

A free 3-month subscription goes to Mr. Coombes. For the "time" part of the competition we had several entries but none that beat our attempt below, which runs in 3.39 seconds:

```
>
   10 TIME=0:A=0:B=1:ONERRORPRINT'
'TIME:END
   20 C=A+B:PRINT;C; :A=B:B=C:GOTO2
0
```

Before we have a flood of letters, the above program is in line with the rules we set—sorry but you have to think smart to win.

Again, thanks to all the people who entered. Don't forget the competition in this month's magazine. Winners of Competition 2 will appear next month.

Paul Barbour

LASERBUG is edited by Paul Barbour.

The Contributors for this month were Paul Barbour, Dr D E Susans and Phil Hirst.

Final copy date for issue 8 (January) is Monday 6th December. Printed by Prestige Press (UK) Ltd, East Street, Chesham, Bucks. LASERBUG, 10 Dawley Ride, Colnbrook, Slough, Berks, SL3 0QH.

Membership: £12 for 1 year (£18 overseas), £6 for ½ year (£9 overseas).

Advertising rates available on request.

Next month's magazine will be a Christmas Special. Being Christmas it will probably be a bit later than usual but then again, at Christmas you won't mind so much (I hope!)

A very special Christmas program will be one of the bigger features in our *Seasonal Program* spot and also as a treat we will be printing a rather good games program—all you arcade freaks look out!

All the usual articles will be there of course, plus a few different ones. However, that is all I will disclose about the next issue—what would Christmas be without surprises!!!

Paul Barbour

One thing I must say is that I am very disappointed with REM. A few pages reviewing the BBC's games, a few more pages giving delivery dates, one page on telesoftware and a single page intro is not my idea of a newsletter.

At LASERBUG we are negotiating about putting programs on both Prestel and Ceefax to increase the service we provide and so many more people can get the benefit of our programs. The only programs that will ever be broadcast however are those from old issues of LASERBUG, no programs in the current issue will be released.

Paul Barbour

LASERBUG
10 Dawley Ride,
Colnbrook,
Slough, Berks,
SL3 0QH.

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...

LASERBUG SPECIAL OFFERS

As an added service to all readers of LASERBUG we present some special offers. These are available to anyone as long as the coupon below (or a photocopy of it) is sent with your order (not applicable for Dracal).

COVERS FOR YOUR BBC MICRO ONLY £3.25
5% DISCOUNT ON ALL GOODS FROM DRACAL
BACK COPY AND PHOTOCOPY SERVICE
NEW CLASSIFIED ADS – PRIVATE AND COMMERCIAL

LASERBUG Dust Covers are manufactured for us by a well known company and are made to extremely high quality. For full details on the covers see LASERBUG Issue 5. Price: £3.25 inclusive

Dracal (North West) Limited have kindly agreed to give all LASERBUG members a 5% discount on their goods. Dracal manufacture 3 products which might be of interest to you. The first is their special graphic design paper which were developed to assist the designing of high resolution displays. Prices: 1-4 £4.95 each; 5-9 £4.50 each; 10+ £3.95 each. Second is a stand which will allow you to place a TV/Monitor on your computer. Price: £29.50. The third item is a Decimal/Hex scale rule for £6.50. All the prices quoted for Dracal do not take into account the 5% discount. We are not acting for agent for Dracal and accept no responsibility for their services. The prices quoted for them are correct and to the best of our knowledge. Dracal can be found at P.O. Box 130, Warrington, WA1 4QB. Phone Padgate 815419

Missed out on a copy of LASERBUG ??? If so then you require our back copy service. Back copies are available for the price of £1.25 each and are available for all magazines printed to date except issue 1. Brief details of the magazines are given below.

Issue 2 (16 pages) – Sound/Envelope, *FX Part 1, Teletext Part 2, User Definable Key & Pontoon
Issue 3 (24 pages) – Programmers Corner, *FX Part 2, Epson Screen Dump, Soft Review, Telesoftware
Issue 4 (16 pages) – Hardspot, What Printer?, Bookreview, Oddspot, Pixel Power, Moving Things, Four In A Row
Issue 5 (32 pages) – Wordprocessor, Machine Code, Disassembler, Seikosha Screen Dump, Circles Galore, Conversions for
             RGB Input

Also please either include 50p P & P for the first magazine and 25p for each subsequent one or include an A4 SAE of appropriate size and postage

For those magazines which we have run out of stocks on we offer a photocopy service. This is also available if there is a particular article or program you wish photocopied instead of buying the whole magazine. Price: 10p/page + postage as for backcopies. For issue 1 therefore which is a 16 page magazine please send £1.60 + postage

Want to take out an ad in LASERBUG but can't afford full advertising rates? Then take out a classified ad now. Rates: 10p/word (minimum charge 15 words) classified/£5 per column centimetre semi display.

Please send me ..... LASERBUG Dust Cover(s) @ £3.25 each
             ... LASERBUG Back Copies of issue(s) .... @ £1.25 each
             ... LASERBUG Issue 1(s) .... (Photocopies only) @ £1.60 each
             ... Photocopie(s) of LASERBUG articles @ 10p/page *
             ... Classified ad(s) *

NAME .... ADDRESS ....

....

I enclose a cheque/PO for .... made payable to LASERBUG

PLEASE ALLOW 28 DAYS FOR DELIVERY

*Please enclose details on a separate sheet