

\$2.00

ISSN 0112-0980

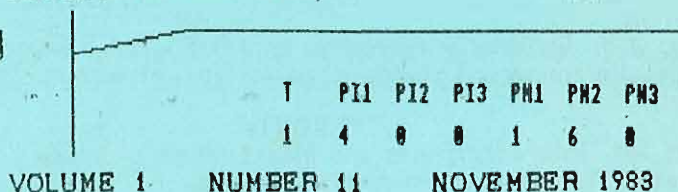
THE BBC MICROCOMPUTER USERS GROUP OF NZ INC

NEWSLETTER

Contents:

Editorial	2
Test Upper / Lower case	3
Speed Typist	4
Irrelevant Envelopes	5
Adventuring	6
Adventure soft & book review	7
Hall of Fame	8
M/c graphics III scrolling	9
Discs, Drives ...	10
PROClibrary	15
Entering listings	15
Quick lower case hint	15
Envelopes + listing	16
Problem page	21
Envelopes graphs	22
Prime Plingies	24
Last month's mistake	24
Mailbag	25
" - CBM64 vs Beeb	26
Software Library	26
Software Advert	27
Local Groups	28
Classifieds & Membership	28

36



EDITORIAL ?

The slave hits the keyboard again. Nothing happens. Drop Envelope. In Envelope: There is a note lying on the ground. Take note. Read Note: 'You can find out about discs this month and tapes next month. Happy adventuring.' reads the note.

By now you have some idea of the present issue, and things to come. (Like Christmas Carols?)

MAGAZINE NAME:

After passing all your suggestions round the committee, two rounds of voting were held on them. The final winner was Sam's, 'BEEBLET' (p.23 Sept). We will start using this at the start of Vol.2, in February, so you all have a couple of months to send in designs based on it, or register any most strenuous objections.

BEEBUG:

After the 31st November, we will NO LONGER accept subscriptions to BEEBUG, until the next financial year and renewal time. So if you wish to get BEEBUG through us, get your cheque to us FAST.

LOCAL GROUPS:

TAUPO:

Taupo is setting up a local group; contact David ph.84-215 for details.

HAWKES BAY:

HawkesBeeb intends meetings probably alternating each month between Hastings Boys' High School, and somewhere in Napier. Contacts:

 Kendall Napier 435-624

Bob Taradale 446-955

Mitch Hastings 778-235

Next meeting Tuesday 13th February at Colenso H.S. Napier, probably This has yet to be finalised.

What do you think of Hawksby, their logo? A cross between a hawk and magpie they say.

AUCKLAND:

December meeting usual date & times, talk on Torch and CP/M.

WELLINGTON:

combining with Wellington Micro Soc, on *** Thursday 8 December. *** Further details announced at November meeting, else phone contacts on Back page.

NEXT MONTH:

All sorts of interesting bits and pieces, some of which YOU haven't sent in yet!!! The slave-driven cracks the whip.

TESTING UPPER / LOWER CASE

Although the code to allow both upper and lower case entries from the keyboard, and treat them the same has been used a few times in the magazine (eg 'Drunk' line 2630, Jan issue), it has never been explicitly mentioned. Thanks to Brian of Ashburton for bringing it to our attention.

How often are you bothered by a program that insists you enter a capital 'Y', and promptly exits if you type in a lower case 'y'. This code accepts lower case AND upper case, and considers them the same.

If you wish to understand the code, you will need to consider the ASCII characters, and how they are stored in the computer. Each character is stored in a separate byte, and has a characteristic bit pattern. When these are examined, there is an obvious and simple difference between the upper case bit patterns, and those for lower case:

UPPER CASE:			LOWER CASE		
CHR	HEX	BINARY	CHR	HEX	BINARY
A	&41	0100 0001	a	&61	0110 0001
B	&42	0100 0010	b	&62	0110 0010
C	&43	0100 0011	c	&63	0110 0011
...
Z	&5A	0101 1010	z	&7A	0111 1010

You will notice that the only difference is that the third bit from the left is 0 for upper case, and 1 for lower case. So all that has to be done is to force this bit to become 0. How? We can either use the fact that you can describe the ASCII codes for upper case as being &20 less than lower case equivalents, or we can use the logical operator 'AND' with a suitable value (which is &DF, or 1101 1111 binary). (See User Guide p.205). The latter is the 'more sophisticated' method, but both will work.

SOME EXAMPLES IN CODE:

```
10 REPEAT
20 A%=GET
30 PRINT CHR$(A%)
40 B%=A%-&20           :REM or B%=A% AND &DF
50 PRINT CHR$(B%)
60 UNTIL FALSE
```

That will print all lower case input as upper case. In most applications, the GET and conversion will be done in the same line:

```
10 REPEAT
20 B%=GET AND &DF
30 PRINT CHR$(B%)
40 UNTIL FALSE
```

SPEED TYPISTS

Tim R. Wainuiomata

I was sitting down in front of my Beeb the other day and I realised that although I can touch type, my speed is very slow (something like 15 words per min). To rectify this I decided to write the program reproduced below. This is not a typing tutor mind you - simply a blank piece of paper to type on. The only thing it does do is provide automatic new-lines and keeps a tab on your typing speed.

```

10 REM *****
20 REM **
30 REM ** TYPING PRACTICE **
40 REM ** BY TIM RYAN **
50 REM **
60 REM *****
70
80 ON ERROR MODE 7:PROCend
85 words%=0:time%=0
90 MODE 3
100 PROCinstructions
110 MODE 0
120 PRINT TAB(10,0) "TYPING PRACTICE"
130 PRINT TAB(11,1) "WORDS TYPED SO FAR: 0000 IN 0000 SECONDS"
140 VDU 28,8,31,79,3
150 end_keys$=" !#$%&()'!*:-^[\_]]*+;<,.)?/:time%=TIME
160 still_flag/=FALSE
170 REPEAT
180 possible_end/=FALSE
190 A$=GET$:PRINT A$;
200 IF ASC(A$)=9 THEN PROCpause
210 IF ASC(A$)=13 THEN VDU 10
220 IF INSTR(end_keys$,A$)>0 OR ASC(A$)=13 THEN possible_end/=TRUE
230 IF possible_end/=TRUE AND still_flag/=FALSE THEN PROCword_done
240 IF possible_end/=FALSE THEN still_flag/=FALSE
250 UNTIL FALSE
260
270 DEF PROCword_done
280 words%=words%+1
290 IF POS>70 THEN VDU 13,10
300 VDU 5;MOVE 568,991
310 VDU 127,127,127,127
320 PRINT RIGHT$("000"+STR$(words%),4);
330 MOVE 768,991:VDU 127,127,127,127
340 PRINT RIGHT$("000"+STR$(TIME-time%) DIV 100),4);
350 VDU 4
360 still_flag/=TRUE
370 ENDPROC

```

```

300 DEF PROCinstructions
390 PRINT TAB(20,3) "*****"
400 PRINT TAB(20,4) "**                **"
410 PRINT TAB(20,5) "**          TYPING PRACTICE          **"
420 PRINT TAB(20,6) "**            BY TIM RYAN            **"
430 PRINT TAB(20,7) "**"
440 PRINT TAB(20,8) "*****"
450 PRINT "The idea of this program is to allow a person who has mastered their
r fingering,"
460 PRINT "say on a mechanical typewriter, to practise their skills without wa
sting reams"
470 PRINT "and reams of typing paper."
480 PRINT "When you have had enough, hit the escape key to see how well you ha
ve done."
490 PRINT "If you just want to pause temporarily to see how you are doing, hit
the TAB key."
500 PRINT "Any key typed afterwards will get you going again."
510 PRINT "If you want to type in lower case, make sure the CAPS LOCK and SHIF
T LOCK are"
520 PRINT "set correctly now."
530 PRINT "Simply press any key to start, then begin typing ....."
540 A%=GET$
550 ENDPROC
560
570 DEF PROCpause
580 timeZ%=TIME
590 A%=GET$:PRINT A%;
600 time%=time%+TIME-timeZ%
610 ENDPROC
620 DEF PROCend
630 ON ERROR OFF
640 VDU 26
650 PRINT TAB(11,10) CHR$141;"TYPING PRACTICE"
660 PRINT TAB(11,11) CHR$141;"TYPING PRACTICE"
670 PRINT TAB(0,15) "You typed an av. of ";INT(words%/(TIME-time%)*6000);" wor
ds per minute"
680 END

```

SOME IRRELEVANT ENVELOPES, JUST TO TRY:

```

ENVELOPE 1,1,0,0,0,0,0,0,126,-1,0,-5,126,0
ENVELOPE 1,133,8,4,0,3,1,1,126,0,0,-10,126,0
ENVELOPE 1,30,0,-30,6,1,6,13,0,0,-126,126
ENVELOPE 1,129;0,8,0,2,6,8;60;0,0,1,126,126
ENVELOPE 1,1,6,6,6,255,255,255,126,-1,0,-10,126,0
ENVELOPE 1,1,4,-4,10,20,10,10,127,0,0,-5,126,126

```

ADVENTURING:

Most people are very aware of the 'clobber-before-you-get-clobbered' type of arcade game. They require very fast reflexes, and an unlimited supply of 50cent coins. An adventure game is more leisurely, demanding less of your quick reflexes than of inventive intellect. As with playing chess, you need to consider possible reactions to your moves before you make them - it's no fun losing something when a bit of forethought would have avoided the loss, or foreseen an answer to a puzzle. Arcade games require colour, and lots of it. Adventure games usually don't.

Steve (Epsom): To the uninitiated, the appeal of a game consisting of a text-only dialogue between operator and computer may be difficult to understand. To those (like myself) who have spent literally hundreds of hours trying to solve the puzzles of Colossal Cave (Level 9's most popular adventure), there is no such difficulty! An adventure is a role playing game (RPG) where the player takes the role of an Adventurer (intrepid or otherwise), charged with a task or mission. It may involve collection of treasures or artefacts, the solving of riddles etc. The computer takes the role of 'mentor', describing locations, answering questions etc. The player instructs it in English with brevity eg GET / DROP / HELP LAMP, BOTTLE, TROLL etc. I have about 20 for the BBC ranging from superb (Acornsoft or Level 9) to mediocre (Computer Concepts). To anyone wanting an adventure to start with, I would recommend Sphinx (Acorn) or Colossal Adventure.

They also have educational value:

Tony T.'s children don't know that abbreviations are permissible, so they are learning spelling as well as extending their vocabulary through playing 'Sphinx' (Acornsoft). Also they learn mapping skills, as a map is drawn to show where they are or have been; and relationships between places. (Here there be alligators!)

Judy (Tauranga)'s favourite relaxation was Philosophers Quest. "I say 'was' because after a determined effort and with the help of my 13 year old son and friend we scored 236 points (out of 250). We are obviously missing one piece of treasure, and countless efforts and bright ideas have failed to find it. Can anybody help? One thing we did discover was that it is essential to 'save' games as we went."

Tony B. who reviewed 'Colossal Adventure' in August is busy trying to open the giant clam in it. (If he'd been listening to Brian Clarke & Alison Holst through October, he would have heard what to do with shellfish.)

Inventiveness and wit are required for these games, and you can profit from mistakes. For those of you unfamiliar with this type of game, some examples from Level 9's

DUNGEON ADVENTURE:

It's not what's there, it's what you've got: a doorway which you attempt to enter; to be told 'Only Archers Allowed'. Solution? Become an archer: find a bow. Sometimes you are not aware that you have passed such a test until you try another day to enter without some item you never figured out a use for.

In dungeons there is the perennial problem of light, or the lack of it. On exhausting my current light-source, I retreated to another room that I knew would light up, and got this reaction:

The octopus waves its arms and you can see. Thus "many hands make light work!"

You cannot see.

A ghoul emerges from the dark and eats you ...

The first time, when my light-source was strong, the octopus was amusing, but this time, ... so much for invention. I have since discovered how to renew my light source. Royden, keep trying - it's possible.

A game can stretch your imagination, sometimes by alluding to books too:

'A gigantic orc's head is carved into the cliff north of the river, its tongue forming a bridge over the waters to you. A ruined tower stands on top of the cliff.'

Orcs - Tolkien. Ever read The Sword of Shannara? Remember Skull Mountain? Later on you may find sticky passages from which you fall to the river below and drown. Sticky? With what?... The wight house is a pimple on its nose. Enough.

A game that has much to recommend it, with an 8-page booklet from Level 9 Computing in the UK (send '9.90 to them at 229 Hughenden Rd, High Wycombe, Bucks. HP13 5PG, England - approx NZ\$24).

BOOK:

'Creating ADVENTURE programs on the BBC MICRO' by Ian Watt. Pub. Interface publications / Addison Wesley.

Shows you how to go about creating your own adventures, but most of it deals with the coding, of 3 long programs. The author originally wrote these programs for a ZX80, and believes 16K RAM is almost essential. There are sections of code that make obvious his unfamiliarity with BBC BASIC; where he uses very complicated code that is not needed by the BBC. Not a PROC in sight either! I gave up. A great pity, as he does have some basic ideas on how to develop 'flat' scenarios, a room numbering system, what various objects could be used for, restrictions on them etc. Nevertheless at almost \$30.00 for 128 pages, I would rather recommend this book to ZX80 users than to BBC users. D&D players wouldn't get much from it either. Perhaps I've been spoilt by games that allow 8-compass points plus up and down, ...

8

that use 27K RAM for 200+ locations, ... but I expected much more from the title than there is; and of higher quality. Spaghetti anyone?

Steve would like to start a BBC Adventure sub-group. Anyone interested should write to Steve Williams, 61 Disraeli St, Epsom, Auckland 3.

HALL OF FAME:

Send in your High Score from any game for inclusion in this page each month. The top 3 scores per game will be published. (Adventure Games stay on until completed!)

PUBLISHER:	GAME	HIScore	SCORER
Acorn	Castle of Riddles	217	K. Family
	Monsters	21,200	Athol
	Philosophers Quest	236	Judy
	Planetoid	247,825	Athol
	Snapper	235,200	Susan
		32,668	Sarah (8yrs)
Prog.Pow	Killer Gorilla	81,000	Susan
Level 9	Colossal Adventure	'too busy	playing to note!'
	Dungeon Adventure	270	Anne



```
10 REM ** Scroll Checker by Tim **
20
30 OSBYTE=&FFF4:A%=0
40 DIM CHECK_SCROLL 20
50 P%=CHECK_SCROLL
60 [
70 OPT 3
80 \ PROGRAM to return value indicating
90 \ hardware or software scrolling
100
110 LDA #275
120 JSR OSBYTE
130 TXA
140 RTS
150 ]
160 SOFT%=(USR(CHECK_SCROLL) AND 0)
170 PRINT"SCROLL IS'BY: ";
180 IF SOFT% THEN PRINT"SOFTWARE" ELSE PRINT"HARDWARE"
190 END
```

TIM R.

Last time I introduced an ASSEMBLY program that drew a picture on the screen using VDU codes. It was a fairly simple program needing some redundant information but was intended to show that using OS is as fast as your own routines would be (possibly faster, the OS routines have been written by experts for as much speed as possible). The whole beauty of these routines is that they are so easy to use. Because parameters are usually passed one at a time, the routines should work just as fast from the other end of the Tube also. Fast processors working with the Beeb should be able to produce some phenomenal results. I really look forward to uses of the Beeb in this direction! (Especially with graphics.)

I had another program nearly ready for this issue, using interrupts, but in the end I decided to juice it up a bit for the bumper issue next month. Look for it; I think you will find it appealing. I've been given a pretty full listing of the VDU and *FX (i.e. OSBYTE) calls and I was surprised to see that there are a large number of undocumented ones, especially OSBYTE calls. There is in fact one for nearly all the numbers from 0 to 255 in OS 1.x. Only those specified as being available for OS 0.1 in the User Guide are expected to work with an OS 0.1. An interesting one I noticed was the ability to tell whether the CPU was doing a hardware or software scroll (as I discussed last time). To find out which is active type in the program below (note: this may or may not work in OS 0.1). The OSBYTE call is &75 and the program should print out hardware if there is no window defined at the moment, and software if you run it after defining a "window".

I guess some of you might be wondering what books would be helpful. There is a review next month in this magazine of two machine code books devoted exclusively to the BBC. For general reference I would recommend 'PROGRAMMING THE 6502' by Rodney Zaks. It is pretty expensive, but something is quite necessary if you are going to be doing your own experimenting. (Unless of course you have the whole language memorised to the last detail). The mnemonics are laid out well in Zak's book with address modes, and other necessary information easily readable. It would be a good idea to read the rest of the book too, to gain a full understanding of working of the 6502 CPU. Next month a bit on interrupts, and a rather curious result from a program "bug".

Listing on left.

DISCS, DRIVES...

Wellington's September meeting: talk by Bryan, notes by Jill.

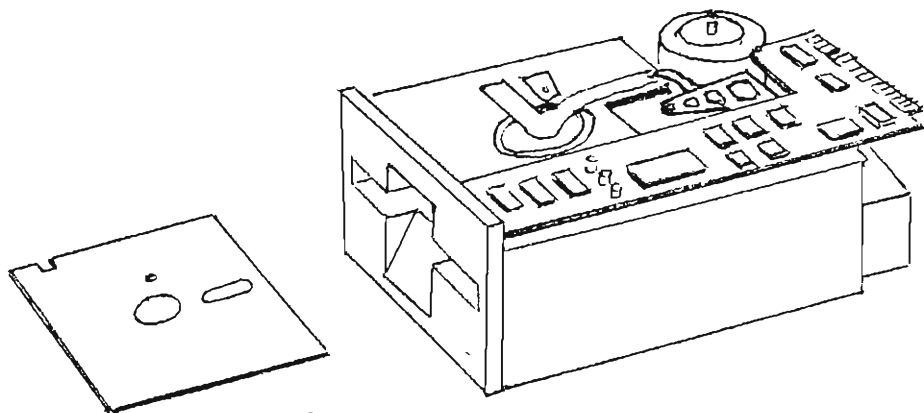
INTRODUCTION:

This was the main topic of the evening: an introduction to discs, what they looked like, what you could expect from them; followed by queries from the floor. Bryan's talk was about them in general; without specific reference to the BBC, showing a naked drive (ie out of its case), and illustrating points with diagrams - some of which are reproduced here.

DRIVES:

The normal drive, in its case, has a front slot for inserting the disc, and a lever to close the gate, thus clamping the disc firmly to the drive spindle, and the unit to be switched on. Inside the box, the main components are:

- 1.an array of chips which hold the internal logic of the system
- 2.a reading head
- 3.a motor which rotates the disc at a constant speed (300rpm)
- 4.a stepper motor to move the reading head to the correct point on the disc
- 5.connections to the computer and to the disc.



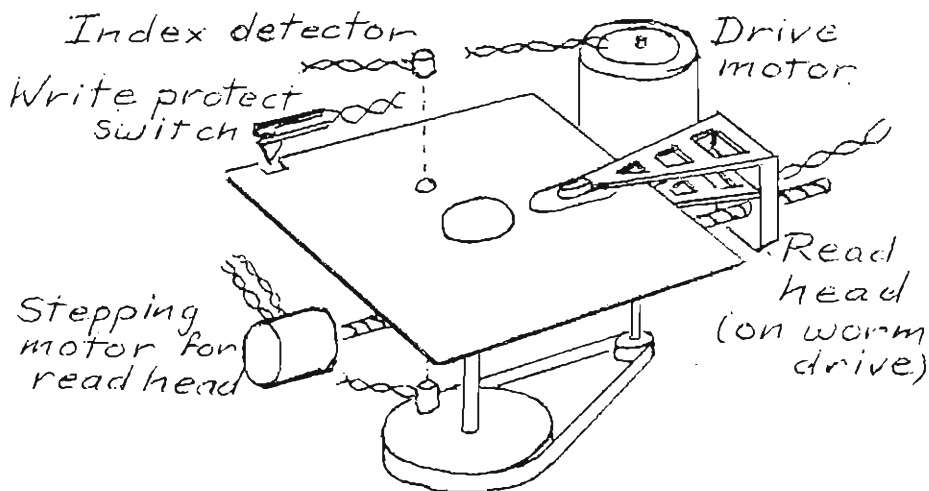
DISCS:

A disc is a very thin plastic sheet, encased in a protective cover 5.25 inches square. It has a large central hole through which the drive spindle passes. The 'disc' contains all its information in a series of concentric circles (usually 80), called tracks. It does not

have a spiral track like the more familiar gramophone record.

A part of the disc is exposed by an oblong slot in the outer casing. This should NEVER be fingered or otherwise damaged. When the gate is closed the reading head is lowered on to the disc at this slot. The stepper motor moves the head back and forth along this slot, to reach a track wanted on the disc.

There is also a small circular hole in the cover. When a hole in the disc passes this index hole, it allows a beam of light to pass from below up to a photoelectric cell, thus signalling the position of the start of each track.



Q. Is the disc always spinning?

A. Not usually. Most systems have an automatic shut-off to stop it when it's not needed.

Q. How long does it take to reach full speed?

A. Generally it is agreed that there is a finite time for it to stabilise, say about 2 revs or 350 milliseconds. The head must then move across to the right track, settle, then wait until the correct part of the track passes underneath it.

There is also a notch on the side of the disc cover. When this is open, information can be written onto the disc. When it is closed the disc is 'write protected', and can only be read from.

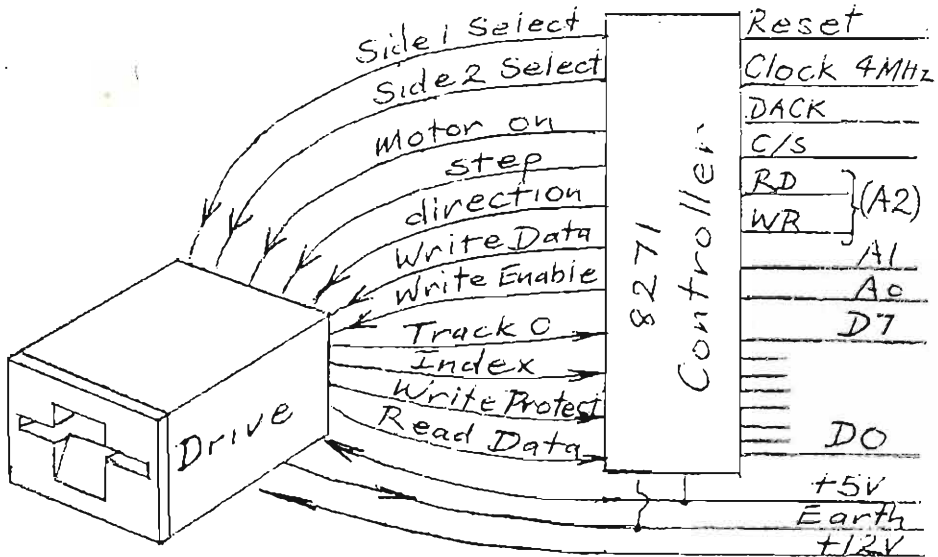
DISC CONTROLLER:

This is a large 'smart' chip inside the computer which controls everything between the computer and the disc drive. From the computer

it has links to the data bus, some address and control lines and a clock signal. It has 5 major registers, each of 8 bits. There are another 12 special registers, storing essential working information such as step timing, head settling time, or allowing a check on drive status.

Q. What is a register?

A. Register is a memory storage location, similar in action to the normal memory of the computer. Some registers are read only, some write only, and others are read or write as for normal memory. TO the disc drive, it controls such actions as select, step forward, write. FROM the disc drive it receives signals such as index, write-protected, track zero etc. It of course relays data to AND from the disc.



It is very complex, and newer models are constantly being produced. The BBC currently uses the 8271 chip. It fits many different computers. The controller operates under the direction of a program situated inside each computer (the 'Disc Operating System').

DISC SIZE:

SECTORS: On a disc each track is split into a number of equal portions called Sectors. These are variable between 10 and 18 on a track. On discs formatted by the BBC, each track has 10 sectors each of 256 useable bytes.

CAPACITY: A disc can have 40 or 80 tracks - normally 80 for drives sold in NZ - it can have 1 or 2 sides, and the computer can be linked

to one or more drives. Some computers can cope with 'double-bit-density', but the 8271 Disc Controller chip does not allow for this. Essentially, on a BBC, a single 40-track, single-sided disc holds 40x10x256 bytes i.e.100K. This can be progressively doubled by:

doubling the number of tracks (ie to 80)

doubling the sides

doubling the drives

Any combination can be used. Thus a dual drive, with double-sided 80-track discs, gives access to 800K bytes of storage. A program can talk to both sides of a disc and to discs in both drives of a dual system.

The BBC will support 2 double-sided drives, or 4 single-sided drives. But note that more power may be needed for the extra drives involved than the BBC's power supply unit can supply, especially if it is also supplying power to other units (eg printer, extra ROMs etc) as well.

Q. Which is better then, to use the Beeb's power supply, or have a PSU with the drive?

A. Use the BBC's until you overload it with other units as well!

Q. Can you turn a single-sided disc over and use the other side?

A. All discs are made by the same process. Quality control means that those with a probable fault in some way will not earn the description 'double-sided double-density', and will be sold with a lesser label - eg single-sided. As well, the write-protect notch and the index detector hole are on the wrong side. There is disagreement amongst the experts over the advisability of using a disc beyond its stated capabilities.

Q. What about the new 3" disc drives?

A. There is no industry standard, 2" ones are coming, and maybe even smaller ones too. Conclusion: stick with the (relatively) cheap and common 5.25" drives for a few years until things settle down.

Q. Are all disc drives compatible?

A. There are 4 basic makes of drive: Olivetti, MPI, Shugart and Tandon. The BBC can be connected to any of these. Appropriate connections are described in the Disc User Guide, and are also software selectable via the *FX255 call. There are also Mitsubishi slimline drives.

Q. Is there a list of errors you can get using discs?

A. Yes - in the Disc User Guide.

Q. How much memory do you lose with discs?

A. The Disc Operating System requires 3K, and takes this from your RAM, starting at PAGE &E00, ending at &1900. Most of this is buffer space for using files, so if you simply wish to load a program

that does not access files itself, PAGE can be reset from &1900 to &1100 before loading in the program from disc. This reduces the loss of computer memory to .75K. It is also possible to load in a program then run a routine to shift it down to &E00 before running the program itself.

Q. What is BOOT?

A. With early computers, the name given to the machine code instructions which had to be given manually in order to get the machine going, was 'bootstrap', i.e. the computer had to pull itself up by its bootstraps. In the BBC you can set up on disc a 'file' called !BOOT which is automatically called after switching on the computer when you press SHIFT and BREAK together. This !BOOT can CHAIN in some other program automatically for you (eg a menu, or your favourite User Define Keys).

FIRST IMPRESSIONS

WOW!! It's fast. What a saving in coffee too - no longer save to tape and make coffee while you wait!

No cassette block 0 bug (1.209); the time it took transferring most of 130+ tapes to disc!

Bad Program? It's quicker to reload the original back in than use trouble-shooter programs to retrieve part or all of it!

Menus make sense now, and save a lot of *, and CHAINing.

Another Users Guide to wander through with a whole host of new *commands and error messages.

Ours is set to boot on pressing BREAK, though usual is on SHIFT+BREAK. The problem with ours if the disc in the drive has an active !BOOT while you develop a program? Pressing BREAK when it is the only way to get out of a program means that the menu is CHAINED over it. Fine if the problem program was already saved ...

The ability to Lock files - especially once children have learnt that SAVE does something. There are several copies of programs with little-fingered alterations and peculiar names. None originally on the disc are lost because they are Locked, and can't be deleted easily. Dave (Ch'ch) points out the quick way to lock everything on a disc:

A. #. L (*ACCESS). If you are developing a program where all is locked, you have then to make a great effort to save a new version with an already-existing name.

PROC & FN LIBRARY:

We have tried to build up and publish your PROCs that may be transferred between programs. Please send them in.

Brian has suggested that we allocate set line numbers to PROCs etc to facilitate merging, along the following lines:

15000 Mathematical FNs
 16000 String FNs
 17000 Graphics FNs
 18000 File handling FNs
 19000 etc ...
 20000 DEF PROCs, with similar major subdivisions.
 30000 machine code or whatever ...

Makes a change from the Dewey classification!

If a library of standard FNs, PROCs, subroutines, redefined characters, Envelopes, machine code routines, etc were to be built, with different line numbers, it would be easy to merge into other programs. He also suggests that besides publishing these in the newsletter, a complete booklet and tape or disc of the current library be for sale to members.

Another possible way of arranging them could be in alphabetical order rather than subject order. How about more suggestions from you all? Brian (Ashburton) is prepared to put some work into this project.

ENTERING LISTINGS:

A lot of queries come from members, who have entered a program from another magazine, and found that when they try to RUN, nothing happens. PLEASE refer to our January newsletter p.5, and July p.22, as taking note of ON ERROR will often help.

Listings for this newsletter are printed out using LISTO7 - that is, there is an extra space put between the line number and the program, and extra spaces are inserted to show up the REPEAT ... UNTIL and FOR ... NEXT loops. These spaces do NOT have to be typed in! A line looking like:

10 NEXT I%
 should be typed in as:
 10NEXT I%

QUICK LOWER CASE:

Brian also says that if you are entering a program with lower case variables, you do not have to put the Caps Lock on and off - merely hold down SHIFT with Caps Lock ON, and out will come lower case, not upper case!!!

ENVELOPES

Or parts of them anyway.

Anne F.

The best explanation I've come across so far of these commands, comes in Jim McGregor & Alan Watt's book 'The BBC micro book; BASIC, Sound and Graphics,' pub. Addison Wesley NZ\$29.95. They devote 42 pages to music, SOUND and envelopes. This article is thus rendered obsolete for the rich (which I'm not!).

It will deal solely with the first 8 parameters of the Envelope command, and I recommend having the Users Guide handy (p.180ff & 244ff). Also, by entering the program given below, all the envelopes are presented in the order given here, with the same numbering. There are also some graphs attached, to illustrate some of the effects.

IS IT AUDIBLE?

If you wish to delve into sounds much, you will need to be able to get a respectable volume from the speaker. First, there is a screw near the speaker plug inside your Beeb which can be turned up or down to alter the volume. If this is insufficient, refer to p.7 of the February issue for details on rewiring the Beeb to fit an extension speaker or employ one in your monitor if there is one.

All envelopes here are accompanied by the same sound to demonstrate them:

SOUND1,N,50,20 i.e a pitch (P%) of 50 and a duration (D%) of 20 or 1 second.

The first parameter, 'N', is the number of the envelope. This number is the one used in the SOUND statement's second parameter. You can define up to 16 envelopes in a program, providing you are not using the RS423 port. If however, you are using the serial port in the same programme, you are limited to 4 envelopes at a time (N=1to4). Any other envelopes desired must redefine (overwrite) these 4 numbers.

The second parameter, 'T', is treated later and briefly, with the promise of more about it next month.

The following 6 parameters affect the pitch of a note, and as suggested by the setting out, there is a relationship between the PI and PN of the same number:

PI1	PN1
PI2	PN2
PI3	PN3

1) If all are zeroes, there is no variation in pitch as a note is played:

ENV.1,132,0,0,0,0,0,30,-4,0,-5,120,80

```

10 ENVELOPE1,129,0,0,0,0,0,30,-4,0,-5,120,80
20 ENVELOPE2,132,4,0,0,12,0,0,30,-4,0,-5,120,80
30 ENVELOPE3,132,4,-4,0,12,12,0,30,-4,0,-5,120,80
40 ENVELOPE4,132,4,-4,0,12,12,12,30,-4,0,-5,120,80
50 ENVELOPE5,132,0,-8,0,1,1,12,30,-4,0,-5,120,80
60 ENVELOPE6,50,16,-16,-16,6,3,3,127,0,0,-5,126,126
70 ENVELOPE7,178,16,-16,-16,6,3,3,127,0,0,-5,126,126
80 ENVELOPE8,129,16,-16,-16,6,3,3,127,0,0,-5,126,126
90 ENVELOPE9,1,16,-16,-16,6,3,3,127,0,0,-5,126,126
100 ENVELOPE10,1,0,-16,-16,0,3,3,127,0,0,-5,126,126
110 ENVELOPE11,1,4,0,0,1,6,0,126,-64,-64,-64,126,126
120
130 FOR I%=1TO11
140 PRINT"Envelope # ";I%
150 REPEAT
160 SOUND1,I%,50,20
170 UNTIL FNtest
180 NEXT I%
190 I%=11
200 PRINT"Envelope # ";I%;" Popcorn tune !"
210 PROCpopcorn
220 END
230
240 DEF FNtest
250 REPEAT
260 PRINT"Next envelope: Y/N? ";
270 *FX15,1
280 R%=GET$
290 PRINTR$
300 R%=CHR$(ASC(R%)AND &DF)
310 UNTIL R%="Y" OR R%="N"
320 IF R%="Y" THEN =TRUE ELSE =FALSE
330
340 DEF PROCpopcorn
350 RESTORE
360 REPEAT
370 READ P%,D%
380 IF P%(<)-1 SOUND1,I%,P%,D%*2
390 UNTIL P%=-1
400 ENDPROC
410 DATA 80,2,80,2,80,2,68,2,52,1,68,3,40,4,80,2,80,2,80,2,68,2,52,1,68,3,40,4
,80,2,96,2,100,2,96,2,100,2,88,2,96,2,88,2,96,2,80,2,88,2,80,2,68,1,80,3,88,4,-1
,-1

```

2) $PI1=4$, $PN1=12$ gives you a row of 12 notes, increasing in pitch by four. As the SOUND statement explains, an increase in the value of pitch of 4 means an increase of a semitone on the previous value. So this envelope describes an ascending chromatic scale with the use of only one sound statement:

ENV.2,132,4,0,0,12,0,0,30,-4,0,-5,120,80

3) $PI2=-4$, $PN2=12$ adds a descending chromatic scale to the above, by virtue of a negative step in the Pitch Increment. The number of steps, 12, ensures that the note finishes on the same pitch as it started:

ENV.3,132,4,-4,0,12,12,0,30,-4,0,-5,120,80

4) $PI3=0$, $PN3=2$ allows the sound to hold the pitch reached at the end of $PI2$:

ENV.4,132,4,-4,0,12,12,12,30,-4,0,-5,120,80

(This is more relevant when $T < 128$)

These parameters can be added together, split, and combined in different ways. So ENV.3 could have also been written using all 3 sets of parameters:

$PI1=4$	$PN1=12$	or:	$PI1=4$	$PN1=6$
$PI2=-4$	$PN2=6$		$PI2=4$	$PN2=6$
$PI3=-4$	$PN3=6$		$PI3=-4$	$PN3=12$

5) An envelope with a quick 'hiccup' at the beginning can be used to imitate a grace note:

ENV.5,132,8,-8,0,1,1,12,30,-4,0,-5,120,80

By varying these pitch changes, trills, turns and a variety of musical decorations can be effected.

There is one caution when mixing such envelopes in harmony - the overall pitch of a note is generally the pitch reached at the end of $PI3$ - which may not always be the one defined in the SOUND statement, which is where $PI1$ starts from.

By now, you may be wondering why T is such a large number, when you may usually have met it with a value of '1', and the User guide gives an acceptable range of 0 to 127. ' T ' is the length of time to be used by each step of PN within the envelope; counted in hundredths of a second.

The Users Guide then refers to bit 7 being on or off. It is set on by adding 128 to the chosen value of ' T '. So above, where $T=132$, in fact $T=4$ without auto-repeat. So $T=132$ is not allowing 1.32 seconds for each semitone in the above envelopes, but only .04 of a second.

The next four envelopes show the effect of altering this parameter

only.

Number 6 allows the PIs to cycle round slowly at the rate of half a second per step; number 7 stops the cycling round and only allows each step to sound once:

6) ENV.6,50,16,-16,-16,6,3,3,127,0,0,-5,126,126

7) ENV.7,177,16,-16,-16,6,3,3,127,0,0,-5,126,126

You may have noticed the variation in volume as the steps occur at different stages of the amplitude section too. This is an effect of the 'decay' part of the envelope. The sound produced seems much longer than the 1 second (D%=20) allowed in the SOUND statement, but when there is another note following the sound, the first will be cut short during the decay phase so it only sounds for the length specified by D%.

The only change to the next two envelopes is to set T to the minimum - number 8 is the same as 7 except that the time allowed for each step has been set to the minimum.

8) ENV.8,128,16,-16,-16,6,3,3,127,0,0,-5,126,126

9) ENV.9,1,16,-16,-16,6,3,3,127,0,0,-5,126,126

Number 9 is also set to the minimum, but allows the envelope to cycle round (auto-repeat) - siren-like wails can be made by this rapid cycling through steps at a far faster rate than allowed solely by SOUND statements. The notes making up this 'noise' come too quickly to be discerned individually.

10) A 'feature' of the Beeb is that when the value given for pitch is outside the range 0 to 255, it 'wraps around', i.e. pitch=260 in fact sounds like pitch=5. The envelope below is likely to demonstrate this effect when the pitch reached is outside this normal range.

ENV.10,1,0,-16,-16,0,3,3,127,0,0,-5,126,126

The sound starts at a pitch of 50 (set by SOUND), then drops by 16 (PI2&3) 6 times (PN2+PN3). $50-(16*6)=-46$ OR 209!

11) The Last Envelope:

ENV.11,1,4,0,0,1,6,0,126,-64,-64,-64,126,126

has its own tune too.

The Program:

This program can be used from immediate mode too. The envelope number used for the tune is variable, so by entering I%=(1to11) then PROCpopcorn you can hear the tune in any envelope you wish. Bet you've already thought of defining a function key to call the PROC too!

Alternatively, to hear it through the range of envelopes defined:
FOR I%=1TO11:PROCpopcorn:NEXT I%

You will also find that the effect of an envelope on a series of notes

(tune) makes them sound rather different from the sound you heard with a single note. This is an effect of the second half of the ENV.

THE GRAPHS: →

These are from graphs produced using the Envelope Editor from the Nov.'82 issue of Beebug. Printed out after each one are the current values of the relevant parameters.

The horizontal axis represents time in 100ths of a second. Time is set to zero at the extreme left. The next number is the length of time taken to reach the end of PN1. Referring to graph 3 (ENV.4):

The number 48 represents 48/100ths of a second, made up of 12 steps as specified by PN1, multiplied by the value of T (actually T MOD 128).

The next number (96) is the time elapsed from the beginning, to the end of PN2; the third (144) is total time to reach the end of PN3:

$$\begin{array}{rcl} T \times PN1 & 4 \times 12 & = 48 \\ (T \times PN1) + (T \times PN2) & (4 \times 12) + (4 \times 12) & = 96 \\ (T \times PN1) + (T \times PN2) + (T \times PN3) & (4 \times 12) + (4 \times 12) + (4 \times 12) & = 144 \end{array}$$

The vertical axis represents pitch. Pitch is initially set by P% in the SOUND statement - here P%=50. The envelope's influence is then shown:

The effect of PI1 & PN1 are to raise P% from 50 to 98 via 12 steps of 4:

$$[P\% + (PI1 \times PN1)]$$

Then PI2 & PN2 make their effect:

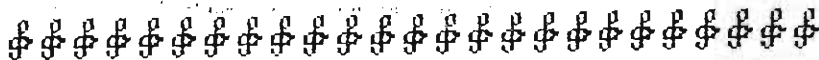
PN2=12; PI2=-4 so there are 12 steps, each decreasing the pitch reached at the end of PN1, by 4. This reduced the pitch to 50, whereupon PI3 & PN3 take up the change.

Here PI3=0 and PN3=12. This means that the pitch changes by 0 for 12 units of T. (ie not at all!).

POST SCRIPT:

Don't expect to make much sense of this at first reading, it does require dream time to sort out! Plotting your own graphs on paper can help sort out the relationships too. I am aware that I have been rather concise - not wanting to take up the whole of the mag for a month - so if you think you would like some of this expanded further, write in and ask, and I'll try to do so. Don't ask for an explanation of the rest of the envelope though - I'm still trying to figure out how that all fits together.

And for the 11th month ENV.11 graph HAD to appear on the cover !!



PROBLEM PAGE:

Some members have suggested we have one. Here it is. What is your problem? Must confess, the Editor expects people are too impatient to wait a month or two for an answer to their problem to appear in these pages, but of course ... not everyone writes in with their problem! Perhaps if it appeared here, AND if it got read, others would solve the same problem that bugs you.

The assumption: that someone reads the problem page and is moved to write in with the solution for it/them.

DEBUGGING

Do we offer a debugging service? Since this is usually asked by people keying in listings from other magazines, either there is a clerical error that they have made, or corrections are published in a subsequent issue of the magazine the listing came from. If such a program comes to the committee, it tends to take an age to find, simply because the mugginses on it are already very busy.

MODern BUG?

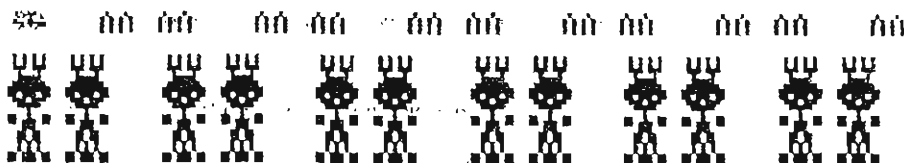
Anton and 'Battleship Stephen'

A program bug was discovered to be caused by the MOD function. Those unfortunate enough to have been taught MOD maths will know that there is no negative result from the mod operation. Time in hours (MOD 12) is never minus 2 o'clock. But! The Beeb does have and allow negative results:

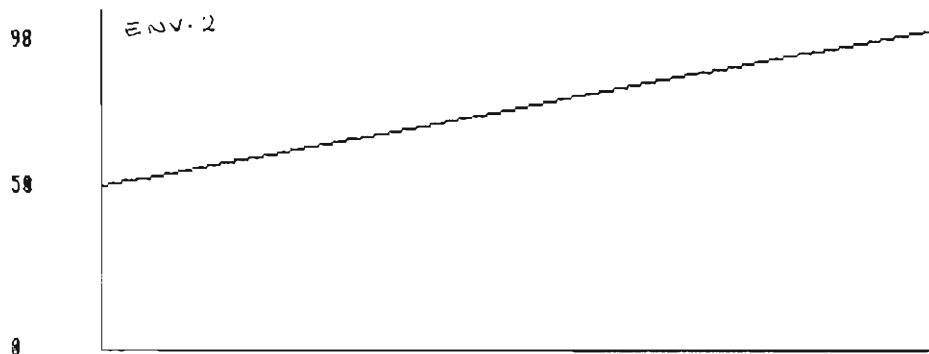
The Users Guide defines MOD in terms of DIV. It refers you to MOD when you look up DIV. But MOD is all about circular maths ... (snarl, snarl,...)

So, ... by writing $A \% MOD B\%$
you have the same effect as writing:
 $SGN(A\%)*ABS(A\%) MOD ABS(B\%)$

In other words, the sign of A% matters, but not that of B%.



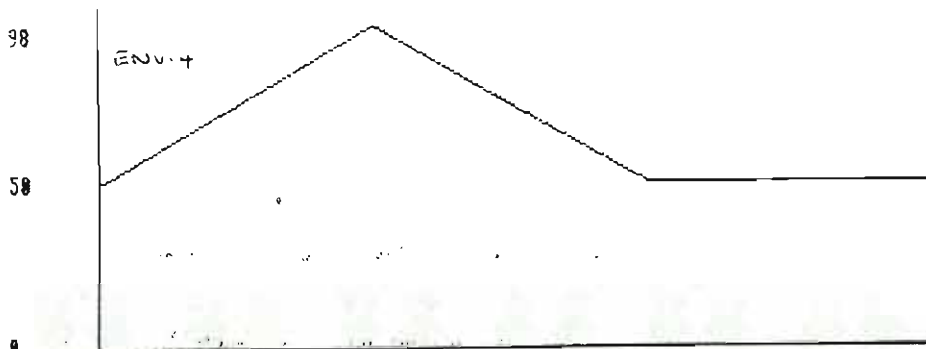
22



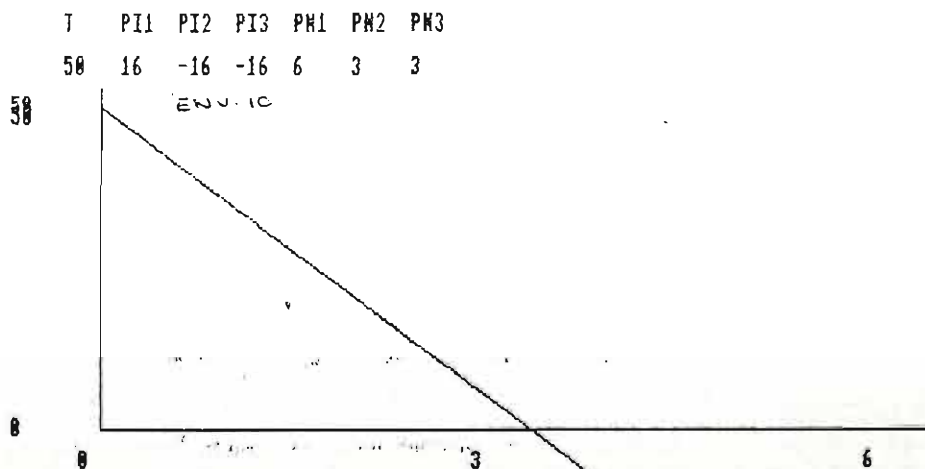
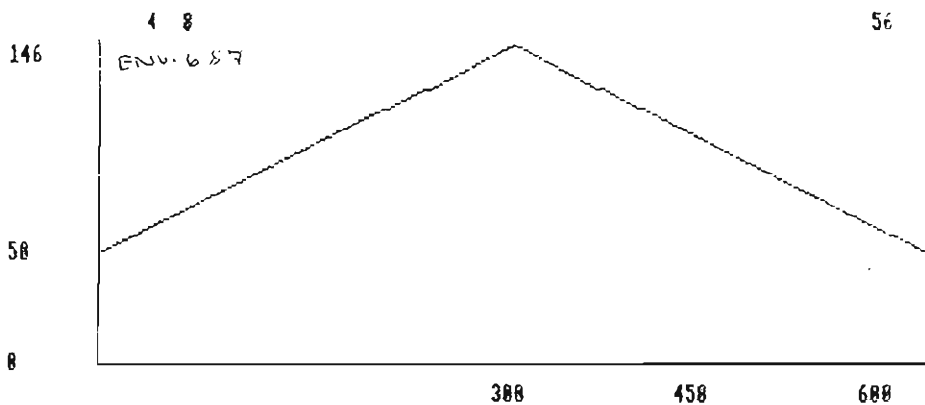
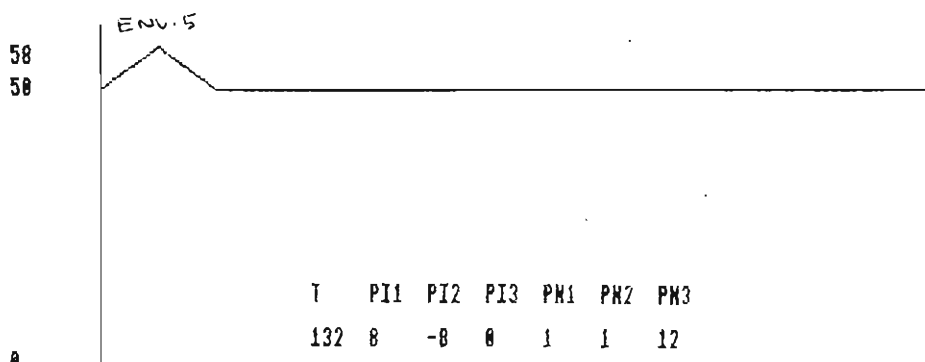
T	PI1	PI2	PI3	PH1	PH2	PH3	
132	4	0	0	12	0	0	484848



T	PI1	PI2	PI3	PH1	PH2	PH3	
132	4	-4	0	12	12	0	48 9696



T	PI1	PI2	PI3	PH1	PH2	PH3	
132	4	-4	0	12	12	12	48 96 144



T	PI1	PI2	PI3	PH1	PH2	PH3
1	0	-16	-16	0	3	3

WOW ! EH ?

PRIME PLINGIES ! ! ! !

Anton E. Wgtn

The prime number program given here cheats. The prime challenge (May issue) specifically disallows plings and queries, but I have found these to be the simplest, fastest and clearest way of solving the problem. The objection that indirection operators do not permit a reader to easily understand a program is usually quite valid, but here they simply allow the creation and marking of a very long line of 'cards' each representing an odd number. That's not obtuse at all. Any odd number (N%) can be found at address S%+(N%-1)/2. Points of note:

1. The method uses the sieve of Erastosthenes - see previous issues of the newsletter. Only one byte array is created and the lower bytes check the higher ones.
2. The array of bytes is not automatically zeroed; this is done at line 100. Contrast this with arrays of integers or reals.
3. As an alternative method, instead of zeroing, which is rather wasteful, use can be made of a repeating pattern every 15 bytes caused by divisibility by 3 or 5. Thus initialization simultaneously marks those most frequently marked off, and zeroes the rest. This method reduces the time taken down to 17.1 seconds. (Note the use of plings !!!)

Why bother? I extended the above method to check all primes to 100 million. It took 42 hours - without printing them. This enabled me to check a predictor P(X) of the number of primes under some value X.

The predictor was faulty and the new, improved predictor is:

$$P(X) = (1+1.6/(53+K*K))*X/(K-1)$$

where K=LN(X). I think it will take machine code to better that predictor.

For the method referred to in (3) above, replace lines 100-110 with the following:

```
100 FOR M%=S% TO S%+N% STEP 15
102 !M%=&10100:M%!4=&1000001
104 M%!8=&10000:M%!12=&101
106 NEXT
108 !S%=0
110 R/=2
```

A MISTAKE:

Last month, under Musical Medley (with apologies to Ian Birnbaum), the variable's name changed. The listing should have either I%'s or M%'s, not both. OK?

```

10 MODE7
20 DIM S% 24000
30 REPEAT
40   REPEAT
50     INPUT "Upper limit of primes ",N%
60     UNTIL N%>8 AND N%<40001
70     TIME=0
80     Z%=2505
90     N%=(N%-1)DIV2
100    FOR M%=S% TO S5+N%:?M%=0:NEXT
110    R%=8
120    REPEAT
130      REPEAT:R%=R%+1:UNTIL S%?R%=0
140      T%=2*R%*(R%+1)
150      P%=2*R%+1
160      FOR M%=S%+T% TO S%+N% STEP P%
170        ?M%=1:NEXT
180      UNTIL T%>N%
190    PRINT2;
200    FOR M%=S%+1 TO S%+N%
210      IF ?M%=0 THEN PRINT2*(M%-S%)+1;
220    NEXT
230    Z%=10:PRINT:PRINTTIME/100;" SECS"
240  UNTIL FALSE

```

MAIL BAG

DAVID, Greenhithe:

1. I've just finished a program (in old-style BASIC) that stores, retrieves and analyses 72x3 item data using disc. I'll shortly be adding correction facilities, using the random access instructions, but in Mode 0 (80 cols + underlining) I'm almost out of memory. Does anyone know of an existing inventory or income and expenses program that would let me analyse and print out up to 10 types of expense?

2. What is 1450 at the end of my address label?

1. Have a look at BBCsoft's 'Record Keeper', \$55.00. It will allow 4 data files PER disc, about 80 records, with 9 fields, each of 26 characters in length. The program can be altered to alter these.

2. Your zip code - wot the PO wants in exchange for cheaper postage rates.

3. David also offers a thumbnail sketch of the editor, with the suggestion of others sending in for publication, as well. No thanks - I've already had Sam (a Sam-ess actually, in Birkenhead) telling me I should have a black tooth. 'Well, greyish sort of. Discoloured' she offers. I don't think my ego could carry the burden!

Brian M. Tokoroa:

Recently I was asked to put my BEEB where my mouth was; I am not a braggart, but when the opportunity arises I will extoll the virtues of the BEEB. So I was given this little Bench Mark test and told how well others had fared viz:-

CASIO hand-held 1 min 11 secs

SHARP hand-held 2 min 36 secs

TANDY 2 min 36 secs

APPLE IIe 12 secs

CBM64 11.5 secs

Well, said I, I'll halve that time (meaning of course, the Beeb would), and punched in the following:-

```
10 TIME=0
20 X=.5
30 FOR Y=1 TO 100
40 X=(PI+X)^5 - INT((PI+X)^5)
50 NEXT Y
60 PRINT X
70 PRINT TIME/100
80 END
```

You can imagine my disappointment when up came the result: 20.8 secs. I immediately reran the program in disbelief that the Beeb had failed to live up to my expectations; Lo and Behold! back came the same answer! What a let-down, to have to say to my CBM64 friend 'Your's is better'. I could hear him now, as he quietly let the world know,

AT LAST MY 64 HAS DONE SOMETHING BETTER THAN THE BEEB
YAAHOO!!!

But, this can't be so, thought I, and looked to see what the Beeb had done wrong. It is always mistyping or misspelling. Twice, thrice, I searched in vain, each time ignoring the obvious error. Then I *Saved The Day!* The silly machine had entered

```
30 FOR Y=1 TO 1000 instead of 1 TO 100.
```

Face saved; and even better than I had said - the Beeb didn't just halve the time, but knocked it back to less than a fifth at 2.08 secs.

Not bad considering it can't even type or spell!

Ed: REM digital calculator known since before mankind: fingers! Rather more error-prone as suggested here. Makes a change from the Apple-bashing tales usually told too.

SOFTWARE LIBRARY:

Athol K. in Wgtn is interested in setting up a 'swap' library of magazine programs - along the lines of you sending a listing in (debugged) that you've typed in, and swap it for another. There would be a small charge for this too. 35c stamps don't come cheap!! Nor do envelopes.

SOFTWARE

for the BBC model B microcomputer.

ELECTRONICS DESIGN PACKAGE NO. 1

SUITABLE FOR THE ELECTRONICS HOBBYIST,
TECHNICIAN OR ENGINEER

All parts of this Program are accessible.

This package will aid in the design of:

1. Resistor combinations to give specified values.
2. Potential dividers.
3. Values of resistors for power supplies based on the National Semiconductor series of 3 terminal adjustable voltage regulator integrated circuits.
4. Inverting and non-inverting amplifier feedback resistor values.

In each of the above cases where appropriate you have the option of displaying simple resistor combinations using E12 standard values that give the closest value to that required.

Supplied on cassette, 1200 Baud.

To purchase, send \$20 with order to:

IDEX INTERNATIONAL LTD
P.O. BOX 6070 HAMILTON

USER GROUP MEETINGS

AUCKLAND:-meets 2nd Wednesday of the month, 8.00pm, VHF clubrooms, Hazel Ave, Mt Roskill. ph. Dave 770-630 x 518(bus) or Kerry 695-355

HAMILTON:-last Wednesday of the month 5pm Waikato Tech B-block staffroom. Ph. Peter, Hamilton 393-990 or Alison, Morrinsville 6695.

HAWKES BAY:-Alternate months Hastings & Napier. Contact Kendall Napier 435-624, Bob Taradale 446-955, Mitch Hastings 778-235.

TAUPO:-Contact David 84-215.

WELLINGTON:-meets LAST Thursday of the month, 7.30pm, Correspondence School Staffroom (1st floor), Portland Cres, Thorndon. ph. Anton 286-289 or Warren 787-005.

CHRISTCHURCH:-fortnightly workshops, Tuesdays 7.00pm at Hagley High School. ph. Michael 582-267.

OTHER CENTRES:- let us know to publish details here.

CLASSIFIEDS & DEALS:

Cassette leads to allow motor control (ie 7-pin DIN to 3 jacks): \$5.00 plus 50c p&p. ph. Kevin Wgtn 845-243 or write c/- Box 9592 Wgtn.

TAPES: C-10s, boxed, leaderless - available from the User group in multiples of 20 \$24.00 plus \$2.00 p&p.

THE NEWSLETTER:

CONTRIBUTIONS:- most welcome. Listings should be sent in on tape or disc. (Tape - 2 copies, one at 300 baud please) A 35c stamp gets your tape / disc returned too. Please include written explanation of listing; text-only material welcome too.

DEADLINES:- For a particular issue, the last day for material is the last day of the month prior to the month of publication.

ADVERTISING:- Rates are \$20 per half page, deadline as above.

Copyright (c) THE BBC MICROCOMPUTER USERS GROUP OF NZ INC.

Published monthly and mailed to financial members.

MEMBERSHIP:

Membership of the users group is on payment of an annual subscription from April to March. For the 1983/4 year it is: \$20.00, plus a \$5.00 joining fee for new members, plus \$5.00 for Auckland & Wellington members as a local Branch levy, which entitles attendance at meetings as detailed above. Renewing members get all back issues of the current year; New members get ALL back issues back to v.1 #1 Jan '83.

THE BBC MICROCOMPUTER USERS GROUP OF NZ INC

P.O.Box 9592

Wellington

Reproduced by ROBERTS PRINTING Co. Ltd, Wgtn.