AUCKLAND COLOUR GENIE USERS GROUP

P. O. Box 27-387,
AUCKLAND. 4.

ISSUE NO: 10 **NOVEMBER, 1984**

Secretary's Report

Last month we had another great meeting, with a record 73 people turning up. Everyone who arrived at the meeting now has a nametag, hopefully left with us to bring along each meeting night. It was a great help to be able to look at the nametag of the person we were talking to if we didn't know them.

Olwen opened the meeting at a quarter to eight, and Andy gave another of his terrific talks on programmable graphics. Further on, you will find the notes of his last talk (now a couple of months ago) with this newsletter. Sorry it is a bit late, but Andy has hardly been in the country over the last couple of months.

Also elsewhere in this newsletter is part three of Allan Clarke's lesson on Assembly Language for Beginners.

****

Chris Bishop is now our official Newsletter Editor. For the last few months I have been doing the newsletter, and Chris (who works at Rank Xerox) has been photocopying it for me. I have then distributed it (with the help of my family in addressing, folding, stamp licking etc.) However, I have found that it was just too much trying to cope with the newsletter as well as everything else I was trying to do, so Chris has kindly stepped in. He has several plans in mind - one is to start up a 'swap shop' where interested people wishing to swap programs they are sick of, can get send their programs to him and he will publish the details in the newsletter. However, he stresses that your tapes must be originals - either Rakon tapes or ours, and he will keep an eye out so that no-one sells the same tape twice!

I would like to thank Ernie Roots at this point, for the marvellous job he made of Newsletter Editor, until his boss thought that his time was better spent 'on the job'. Ernie is now dividing his time between Auckland and Wellington, and just hasn't got the time available any more. But thank you Ernie, you did a great job.

****

Our first lot of royalty payments were made in the middle of September. Nearly $200.00 was paid out to three different companies, and all have written back expressing gratitude for the payments. James Brier, whose programs Bank Robber and Invaders we are selling, said that our sales are 'marginally above' the English sales. Tomorrow Computers, who sell the Colour Quest adventure series, also were 'surprised and pleased' at the number of programs we are selling. The next payment will be in December. A nice Christmas present for the English companies.

****

You will find 'Software Lists' now available. The country members should all have them with their last (No 9) newsletters. I do apologise for the delay in sending the October newsletters, but I wanted to get this list typed up and photocopied so that it could go out at the same time, and I just didn't realise how much time it was going to take me.

These lists will be updated from time to time. You will find pages in your newsletter that can be taken out and added to the software list. We already have another few programs to add to the list now. Also, a couple of programs that are brand new might have a bit of a delay in copying, while Ken gets the original from me, and makes up a master.

New software for this month is:

A10 Bomber $10.00

This program, according to Gum's review is a lot better than 'Scramble' - which we already have. It has several screens and skill levels.

****

Still no sign of the Gum magazine. Apparently, the September issue has been scrapped, and a 'twice-as-big' September/October issue will be sent to us 'some time over the next couple of weeks'. Hopefully it will arrive in time for the meeting. I am writing this at the end of October, so I have got two weeks yet before I need to panic! It better be here!!

Ken and Andy have offered to re-align the heads of tape recorders of those that are having trouble with loading. Just bring your recorders along to the meeting, and (time permitting) they will realign them on the spot.

MUSIC

Those of you that have purchased this tape so far, can you please change Line 134 to say Colour 10 instead of Colour 16 as it is at present. Those purchasing after 20th October will have this changed already. This program, on arrival from England, unfortunately slipped through without being checked (naughty, naughty!).

You will find a review on this program from Keith McGill, who picked up this error. He thought, until he went through the program, that the graphics were pretty poor. If you don't want to change this line and re-save, just send your tape back to us, and we will do it for you.

****
*** There will be a small Christmas party next month. Can you all ***
*** bring a plate each please, and the drinks will be 'on the house' ***
*** so to speak. As we are in a church building we will have to keep ***
*** it toned down, but we do want to celebrate the coming Christmas ***
*** season. ***
****

Secretary's Report

STATE OF ACCOUNTS

In future I will include details on our monthly bank balance, just to keep you up to date with finances.

Balance last month (as at end of September)			1689.89
Deposits for October from software sales		+	926.90
Less Monies Paid Out This Month:-			2616.79
royalties	97.03		
400 blank tapes	495.60		
photocopy paper	15.00		
nametags	37.85		
rent	15.00		
stationery	13.80		
postage	96.70		
software etc. for resale	458.05	-	1229.03
			$1387.76
Monies invested @ 9% with United Bldg Society		-	1000.00
Balance in cheque account			$ 387.76

With the $1000.00 invested this month, we now have a total of $1500.00 with the United Building Society on call at 9%.

If anyone wants to look at 'the books' do feel free to ask me. I bring them along each time to the meetings.

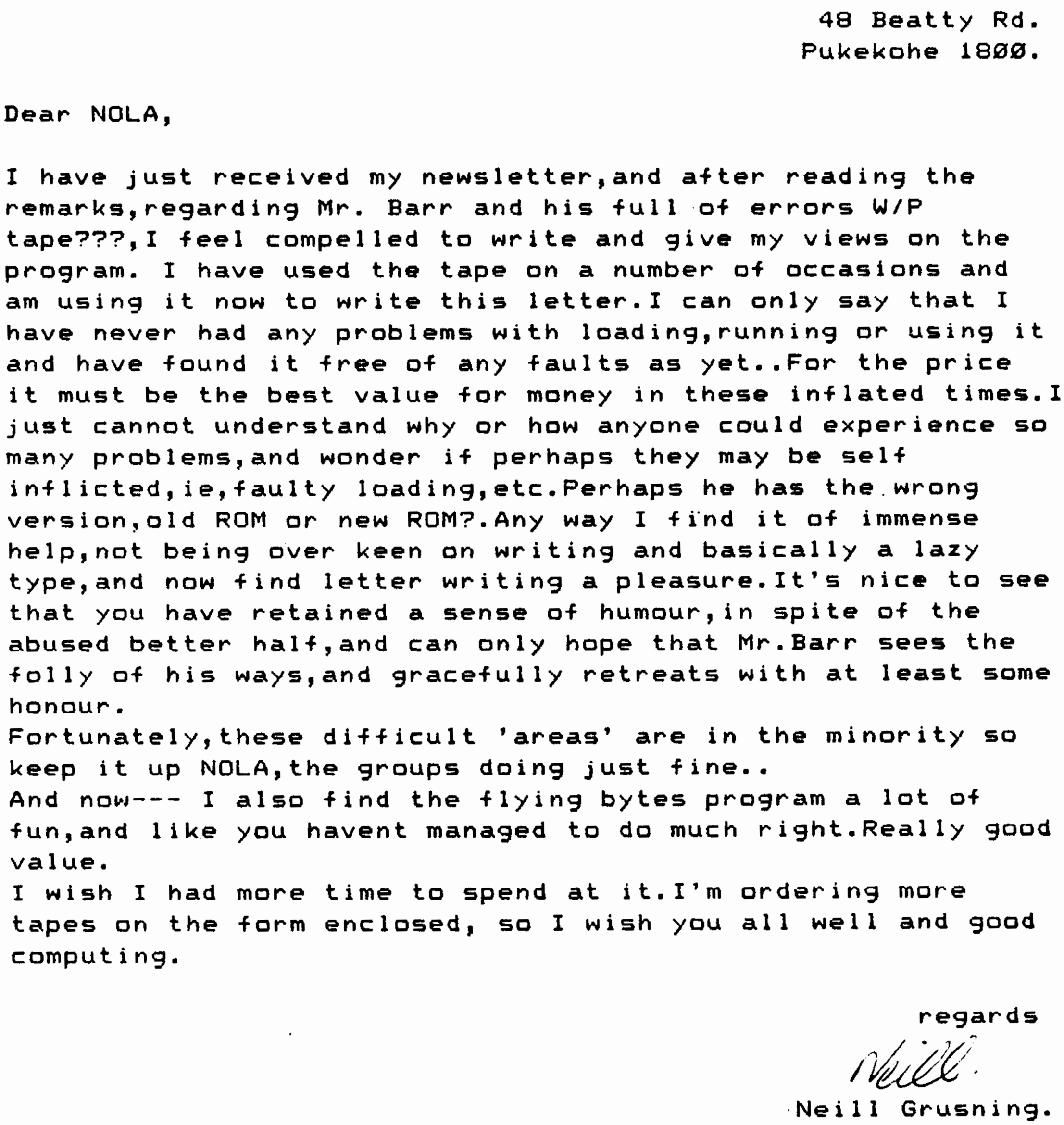
- Nola Huggins

MEMBER'S REQUESTS

from KERRY REID
He needs a joystick routine that he can add to his programs as he has a set of joysticks and would like to adapt more of his programs to work with them. Can anyone help?

from GENE RIGNEY
He would like to know if there are any other members that are Primary School teachers. He would be interested to find out the use, if any, of their Genie in the classroom. He writes and uses Drill-type programmes, and occasionally uses the Genie for word processing, but other ideas would be welcome. Anyone out there?

I have had a lot of phone calls and messages about the Stirling Barr/word processing tape episode. However, I would like to include this letter in our newsletter, as it seems to sum up the general feeling.



48 Beatty Rd.
Pukekohe 18ØØ.

Dear NOLA,

I have just received my newsletter,and after reading the remarks,regarding Mr. Barr and his full of errors W/P tape???,I feel compelled to write and give my views on the program. I have used the tape on a number of occasions and am using it now to write this letter.I can only say that I have never had any problems with loading,running or using it and have found it free of any faults as yet..For the price it must be the best value for money in these inflated times.I just cannot understand why or how anyone could experience so many problems,and wonder if perhaps they may be self inflicted,ie,faulty loading,etc.Perhaps he has the wrong version,old ROM or new ROM?.Any way I find it of immense help,not being over keen on writing and basically a lazy type,and now find letter writing a pleasure.It's nice to see that you have retained a sense of humour,in spite of the abused better half,and can only hope that Mr.Barr sees the folly of his ways,and gracefully retreats with at least some honour.
Fortunately,these difficult 'areas' are in the minority so keep it up NOLA,the groups doing just fine..
And now--- I also find the flying bytes program a lot of fun,and like you havent managed to do much right.Really good value.
I wish I had more time to spend at it.I'm ordering more tapes on the form enclosed, so I wish you all well and good computing.

regards
Neill
Neill Grusning.

---

from ROSS KAY

```
E.G. 25Ø1 BANK ACCOUNT
CHANGES TO PROGRAMME
Line 25Ø Change 32Ø to 52Ø
Line 36Ø Shift FOR J=1 TO 1ØØ: NEXT J To after-(Y/N)";
LINE 42Ø Remove << Son >>
Line 15ØØ Add space after pay and befor "
Line 199Ø Add space after value and before "
Line 233Ø Change 224Ø to 2235
ADD line 2335 Print @ 6Ø8, "RECORDING DATA."
ADD line 23Ø5 Print @ 32Ø, CHR$(3Ø):Print @ 44Ø, CHR$(3Ø):
    Print @ 52Ø CHR$(3Ø): Print @ 6ØØ,CHR$(3Ø)
```

EDITORS NOTE

Those of you at the last meeting will recall that due to increassed membership and demands for software Nola was not able to give enough time to the newsletter as well. So as I have been involed with the letter for awhile the commitee asked if I would take it on fully.
So any articles,queries or corespondance regarding the newsleter should be now sent to me at
6 JENANNE PLACE GLENFIELD AUCKLAND

When space permits we will have a feature called SOFT SWAP SHOP (try saying that fast three times). The idea being that if you are tired of a certain GAMES program (I don't think yu can get fed up with a utility), then let me know what you want in its place and I will print it. It will then be up to other members to contact you direct to arrange the swap

This feature is on a trial basis and if we find people swapping copies of tapes it will be stopped. If anyone offers you a tape which is a copy plêase tell one of the commitee. Remember the club exists for your benefit and if it folds through abuse and cheating YOU are the ones that will suffer.

CHRIS BISHOP

SOFT SWAP SHOP

PAUL SEYB:- Swap ANDROMEDA for any adventure game without graphics or SAUG

CHRIS BISHOP:- Swap METEOR for any action game

Addresses & phone No's available from membership list

Chris Draper of Taupo, has sent in the following article on the **TOOLKIT** which is available from Moonshine. Before ordering though, make sure you are able to get them.

Toolkit is an EPROM based program designed to aid the user in writing and debuging machine code. It can also copy Colour Genie software,and load/convert TRS-80 tapes. Comming in two EPROMS, which fit a standard Cartridge board,(this is available also), Toolkit plugs into the expansion interface on the rear of the computer.

To enter Toolkit, use the command "CALL C000" ,and Toolkit will introduce itself. The commands, which can all be quickly mastered, are all one letter entry, followed by any relevant data. The program is well error-trapped so new users will not find themselves in difficulty over miss-entered commands or, where possible, missing data.

Here is a brief summary of the commands:

COMMAND	DESCRIPTION
Breakpoint	Set breakpiont at specified address
Ascii	Dump memory in ascii formatt from specified address
Binary	Maths & hex/decimal/binary converter
Cancel	Cancel Printer
Dump	Dump memmory in hex formatt
Edit	Write and alter programs in RAM
Find	Locate specified byte
Goto	Transfers control to anywhere in memory
Hunt	Locate specified 16-bit address
Input/init	Input from port/Initialse Toolkit
Jump Calc	Calculate jump relative displacements
Keyread	Write Text into memory
Load	Load System/basic or TRS80 formatt Tapes
Move	Move any memory block
TRS-GENIE	Convert TRS L2 basic to Colour Genie Basic
Output	Output to port
Place	Place a tracepoint
Quit	Quit to basic
Redefine	Redefine user area
Symbol	Symbolic dump (disassembler)
Trace	Trace operation of a program
User	Activate user routine
Verify	Compare two blocks of memory
Write	Write Genie/TRS80 system tape
Exchange	Swap two blocks of memory
Zero	Fill any block of memory with sopecified byte

I've been using Toolkit for several months and I'm delighted with it, its a credit to its designers at Modec Instruments. All the instructions are easy to use, but I still advise you to keep a copy of the Z80 instruction set handy if you intend to use Toolkit to its fullest capabilities.

Toolkit is available from:

Moonshine Computers Ltd.
70 Vicoria St,
Lower Hutt.

REVIEW of MUSIC (Gumboot Software)

This tape is pretty basic, in more senses than one. It could be useful, it could be fun in the right hands, although it is open to some criticism from the teaching methodology side - no positive reinforcement, no immediate correction of errors, no scoring.

The options are:
(1) PIANO
The action and sound is more like an organ. Uses Keys 1-8 for an octave (choice of high, middle or low) with Q W R T Y for the black notes. A bit restrictive if your desired tune exceeds an octave, unless you have nimble fingers.

(2) WRITE / LOAD / SAVE PROGRAMMES
Quite useful. Limited in its ability to create varied time values and very slow on "save".

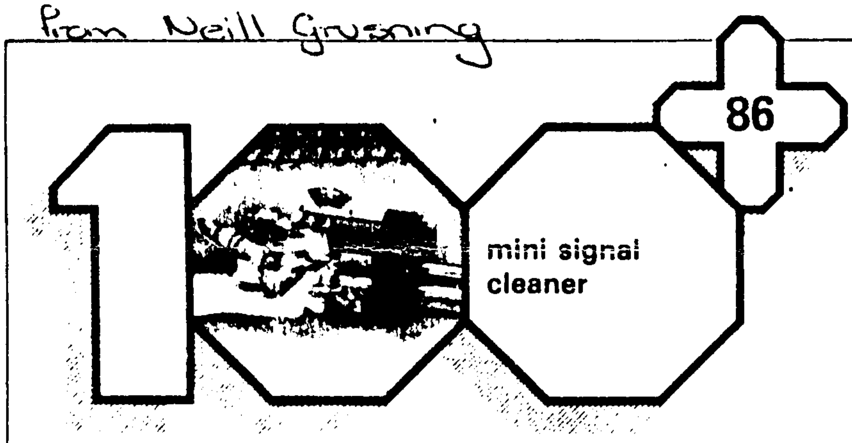
(3) MUSIC TEST
Quite good aural training. Does not score you. Two levels of difficulty.

(4) PLAY THAT TUNE
A "Simon Says"-like test of musical memory. Doesn't score you, and doesn't point out your mistakes or repeat correct sequence. Any number of notes between 1 and 10 can be selected.

- Keith McGill

From Neill Gruening

ELEKTOR. july/august 1984

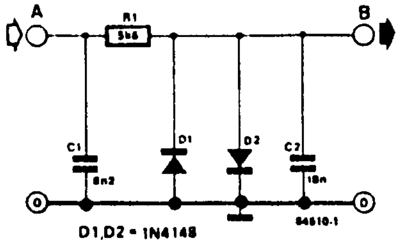


mini signal cleaner

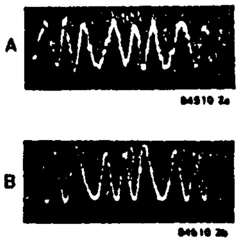
FSK filter for computers

A problem well known to personal computer users is the difficulty of swapping cassette tapes containing software. One of the main reasons for this is the setting of the read/write head in the cassette recorder. This should be at 90° with respect to the tape but in practice this is not always the case, with the result that loading a program from a 'strange' tape causes problems. When using FSK (Frequency Shift Keying) the signal cleaner here provides a very marked improvement. The time spent searching for the correct signal level is then greatly reduced. As the filter requires only five components there should be no problem finding a space for it within the case of any computer.

The layout of the circuit is not at all complex. The signal passes first through the low-pass filter, consisting of R1 and C2, which has a cut-off frequency of about 1600 Hz. In frequency shift keying a '0' or '1' is recorded on the tape as a sinusoidal signal (with frequencies of 1200 and 2400 Hz respectively) so this filtering removes all the 'rough edges' (figure 2a) from the signal. The result is shown in figure 2b. The two diodes limit the amplitude of the output signal to about ±600 mV.



D1,D2 = 1N4148



ASSEMBLY FOR BEGINNERS - PART THREE

I am even more amazed than you that the last set of programs worked as they were written at a great height (over the Pacific) without any opportunity to check them out on a Genie of any kind. It may be too optimistic to expect the programs to work for a third time as I am still Genie-less but now at a great distance (19,000km), nevertheless, here goes.

A useful utility for a BASIC programme is the ability to save and restore a particular format on the screen. This can be done in BASIC but tends to be rather slow and can detract from the enjoyment of a game. This machine code routine is simple, fast and can be incorporated into a BASIC program.

As before, load up your monitor (G-MON is assumed for these articles) and type in the following under the Modify command starting at 7F00:

```
21  00  44  11  00  B0  01  C0  03  ED  B0  C9
```

(See previous articles if you are not sure of what to do).

Exit (X) from the Modify code and return to BASIC (S). Now enter the command CALL7F00. Nothing much seemed to happen? What this routine does is to shift a copy of the screen data into memory using the powerful instruction LDIR (load, increment and repeat).

21 00 44	LD HL,4400H	Load the start of text screen into HL register.
11 00 B0	LD DE,B000H	Load the destination address into DE register.
01 C0 03	LD BC,03C0H	Load No. of screen bytes into BC register
ED B0	LDIR	Copy screen into memory at B000
C9	RET	Return to BASIC.

It is all very well saving the screen but how do we get the screen back? Simply by reversing the values in the HL and DE register pairs. Type CALL BCD0 to re-enter G-MON and type in the following under Modify starting at 7F17:

```
21  00  B0  11  00  44  01  C0  03  ED  B0  C9
```

Exit from the Modify mode, return to BASIC (S) and type call 7F17. You should now have the screen back just as it was when you saved it.

If you play around with the save-and-restore routine as it is you will find that it does not save the colours if you alter the colour layout of the screen between saving and restoring. We have to save the colour information as well. Although there is a better way to do this, we will use the same routine for now as these articles are for the beginner rather than a dazzling display of obscure elegance.

The current screen colour information is held at F000H. So we have to alter the address going into the HL register pair from 4400H to F000H. We also have to up the value going into the DE register from B000H to B400H. Now reenter GMON and the full program can be entered. REALLY lazy hackers will only fill in the missing bits. Starting at 7F00 the complete program is:

```
21  00  44  11  00  B0  01  C0  03  ED  B0
21  00  F0  11  00  B4  01  C0  03  ED  B0  C9
21  00  B0  11  00  44  01  C0  03  ED  B0
21  00  B4  11  00  F0  01  C0  03  ED  B0  C9
```

The Assembly notation is not written out in full as it only repeats the basic module using different sources and destinations. Besides, it increases the possibility of editorial typing mishaps.

The LDIR instruction takes the value in the byte pointed to by the H6 register and transfers it into the byte pointed to by the DE register. It then decreases the value held in the BC register by one and increases the contents of both the HL and DE registers by one. It keeps repeating the whole process until the value held in the BC register falls to zero. And it does it fast!

BASIC programmers can incorporate this routine thus:

```
FOR I = 0 TO 45:REM In Initialisation routine.
READ K:POKE&H7F00+I,K
NEXT
DATA  33,0,68,17,0,176,1,192,3,237,176
DATA  33,0,240,17,0,180,1,192,3,237,176,201
DATA  33,0,176,17,0,68,1,192,3,237,176
DATA  33,0,180,17,0,240,1,192,3,237,176,201
```

The commands CALL7F00 or CALL7F17 can now be used in your program whenever you want to save or restore a screen. This version will only save one screen but could be modified to save more than one by altering the destination value for each screenful.

- Allan Clarke
(E&OE)

The following article is one that should have appeared a few months ago, after the last talk that Andy gave at the meeting. However, he has been away flying ever since, and has only just been able to prepare this for the newsletter on his return.

CREATING LARGE HEADINGS USING BLOCK GRAPHICS

Firstly lets look at the capabilities of the Genie and the graphics available.

Our switch on the CG goes to CHAR1 mode. This is the first of 4 possible character sets. The ASCII numbers start at 128 and only the ones above this change ie 0-127 are always the same graphic character.

CHAR 1	all programmable graphics
CHAR 2	192 to 255
CHAR 3	128 to 191
CHAR 4	128 to 255

Codes 128-191 are not available directly from the Keyboard. Codes 192-255 are available from the keyboard but will only print on the screen as long as you are in the correct character mode i.e. 2 or 4. The computer will though, store the blank ones (ASCII codes) and if the correct CHAR mode is selected then change the blanks to the appropriate graphics shape. I will only use the defined characters at this stage.

So it is important you have the correct CHAR set selected.

Now The Headings
There are a number of ways of creating large block type words or more importantly graphics shapes.

I will demo the methods you can use with the word 'hello'.

Lines 70-140
Let's look at the first method. Decide on the shape you want and the graphics blocks you are going to use. It pays to plot this out on graph paper or the like. Now make up data statements for each horizontal section e.g. 5 lots of data. When writing this into the program put it at the end of the program in a subroutine using for/next loops to make up a string of characters for each line.

Let's look at the program.

LIST 70-140

5 data lines then for next loops to form Z$(1) to Z$(5).

Now when you want to print the graphic shape print the five Z$'s one below the other. Using this method, once the strings are made up the shape can be printed anywhere.

The second method you could use is to make up a large Print@ statement with the shape inside it. As a line can only have 255 characters you have to be frugal in character usage in making up the shape.

Let's look at lines 210-220.
There are a couple of drawbacks here. The shape can only be printed in the same X slot all the time. Y direction doesn't matter but X will and cause the top of the shape to be misaligned. Also can only use 1 colour for the complete shapes. Only limited to small shapes because of the 255 character limit per program line.

```
10 CHAR4:CLEAR550:CLS:GOSUB120:GOSUB270
20 PRINT:PRINT"SELECT DEMO (1) (2) (3) (4) (5)"
30 S$=INKEY$:IFS$=""THEN30
40 S=VAL(S$):S=INT(S)
50 ONSGOSUB300,70,150,210,250
60 GOTO20
70 DATA202,32,32,202,32,202,202,202,32,202,32,32,32,202,32,32,32,205,202,202,224
80 DATA202,32,32,202,32,202,32,32,32,202,32,32,32,202,32,32,32,202,32,32,202
90 DATA202,202,202,202,32,202,202,202,32,202,32,32,32,202,32,32,32,202,32,32,202
100 DATA202,32,32,202,32,202,32,32,32,202,32,32,32,202,32,32,32,202,32,32,202
110 DATA202,32,32,202,32,202,202,202,32,202,202,202,32,202,202,202,32
115 DATA228,202,202,225
120 CLS:FORQ=1TO5:Z$(Q)="":FORA=1TO21:READX:Z$(Q)=Z$(Q)+CHR$(X):NEXTA:NEXTQ
130 FORX=1TO5:PRINTZ$(X):NEXT
140 RESTORE:RETURN
150 CLS:PRINT@410,"J   J JJJ J    J    MJJN"
160 PRINT@450,"J   J J    J    J    J   J"
170 PRINT@490,"JJJJ JJ   J    J    J   J"
180 PRINT@530,"J   J J    J    J    J   J"
190 PRINT@570,"J   J JJJ JJJ JJJ dJJb"
200 RETURN
210 H$="J   J JJJ J    J    MJJU
           J   J J    J    J    J   J
           JJJJ JJ   J    J    J   J
           J   J J    J    J    J   J
           J   J JJJ JJJ JJJ dJJb"
220 CLS:PRINT@410,H$
230 RETURN
250 CLS:GOSUB210:FORX=1TO5:CHAR1:FORZ=1TO100:NEXTZ:CHAR4:FORZ=1TO100:NEXTZ:NEXTX
260 RETURN
270 N$=Z$(1)+CHR$(26)+STRING$(21,8)+Z$(2)+CHR$(26)+STRING$(21,8)+Z$(3)
                    +CHR$(26)+STRING$(21,8)+Z$(4)+CHR$(26)+STRING$(21,8)+Z$(5)
275 RETURN
290 CHAR4:RETURN
300 CHAR1:GOSUB400:PRINT@778,"CHAR '1'":GOSUB450:CHAR4:CLS
310 CHAR2:GOSUB400:PRINT@138,"CHAR '2'":GOSUB450:CHAR4:CLS
320 CHAR3:GOSUB400:PRINT@778,"CHAR '3'":GOSUB450:CHAR4:CLS
330 CHAR4:GOSUB400:PRINT@910,"CHAR '4'":GOSUB450:RETURN
400 CLS:X=128
405 COLOURRND(8):FORZ=1TO12:PRINTTAB(Z*3)CHR$(X);:X=X+1:IFX=256THENRETURNELSENEXT
410 PRINT:PRINT:GOTO405
450 PRINT@880,"PRESS ANY KEY TO CONTINUE"
460 I$=INKEY$:IFI$=""THEN460ELSECLS:RETURN
```

J= MODSEL @
M= MODSEL C
N= MODSEL V
so does
U= MODSEL V
d= MODSEL Z
b= MODSEL X
sorry printer
problems again

The 3rd way is to use print lines for each line of the shape, keying in the shapes required in each line. Once again some sort of design will need to be drawn up prior to entering a program line. This is quite a simple method that allows colour changes for each line but is difficult to shift around the screen.

Let's look at lines 150-200.
Now, an interesting effect but one which is easily achieved.

We know that changing CHAR will either form on or off various characters.

Okay, let's use this to make the word 'hello' flash. By going to a sub-

routine which alternately changes char sets the 'hello' will flash or blink. Great for attention getters. Just alter the timing loops to change the flash rate. Note that characters below 128 ASCII remain visible on the screen.

- Andy Russell

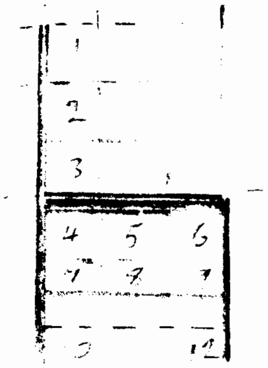
PROGRAMING GRAPHICS BLOCKS

Last time we looked at using the progammed graphics in the computer. To recap-

CHAR1.	128-255	programmable
CHAR2.	128-191	programmable
	192-255	key fronts
CHAR3.	128-191	graphics set
	192-255	programmable
CHAR4.	128-255	all preprogrammed

With that knowledge in hand then as we saw last time by changing CHAR sets the blocks of graphics can be displayed on or off and if programmed blocks are used also the shape of things can be changed. Remember the characters below 128 are alphanumeric and are the same all the time regardless of CHAR set chosen The total No of charecters that can be programmed is 128 (128-255)

The first step is to draw the desired shape on graph paper. Try to make up the shape with preprogrammed blocks. Remember not to use blocks from different CHAR sets as they may not all be available at the same time leaving gaps in your design. If you intend programming all your shapes then stick to CHAR 1. Otherwise CHAR 2 or 3 are the best as there is a mix available between programmable and preprogrammed blocks. Lets program some blocks to make up this shape.



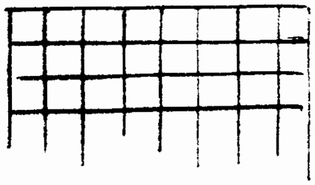
One of the problems with block graphics is block overlap. You have to be careful of entering block positions on the screen.

12 blocks are required for this shape.

Now draw each individual block on an 8/8 grid :

To help it is handy to have the grid marked with values as shown

```
1
2 6 3 1
8 4 2 6 8 4 2 1
```



All that is required now is to start at the top of each block and working horizontally add up the values of active sections of each row

In block 1 the first two segments are active in each row so they all equal 192. Lets look at block 7the numbers add up as follows:-

ROW	1	192
	2	192
	3	192
	4	255
	5	255
	6	192
	7	192
	8	192

Now we need to place all these numbers into the RAM. The RAM area starts at F400H and stops at F7FF. This area will hold 8 bytes of imformation for CHAR 128-255.
It pays to put each CHAR as a seperate DATA statment. So blocks 1,2,3 & 10 would each be.

```
DATA192,192,192,192,192,192,192,192
```

As some blocks are the same you only need 8 lots of data statements. Remember now to note the ASCII No for each CHAR. This depends on where in the RAM you poke the data. Lets put it in at F400H onwards. Write a FOR/NEXT loop to poke the data in.

```
FOR X=0 TO 63:READA:POKE &HF400TX,A:NEXT
```

Now CHR$(128) starts at F400 and finishes at F407H. CHR$(129) starts at F408H and goes on to F40FH etc.
Now all you need to do is incorporate this into your program and use the graphics as required. Of course a graphics program such as the one Chris Bishop has written wil greatly assist with the oring task of number entering but until you understand what you are doing I suggest the method I have shown.

ANDY RUSSELL

```
10 CLS:CLEAR100:GOSUB100
20 CLS:CHAR1:COLOUR4
30 GOSUB190
40 PRINT@379,CH$
50 FORX=1TO2000:NEXT
60 COLOUR8:PRINT@248,STRING$(25,197):PRINT@768,STRING$(25,197):FORX=1TO12
70 PRINT@248+40*X,CHR$(197):PRINT@272+40*X,CHR$(197):NEXTX
80 FORX=1TO20:CHAR2:FORZ=1TO50:NEXTZ:CHAR1:FORZ=1TO50:NEXTZ:NEXTX
90 RESTORE:GOTO10
100 DATA192,192,192,192,192,192,192,192
110 DATA255,255,255,255,192,192,192,192
120 DATA255,255,255,255,0,0,0,0
130 DATA255,255,255,255,3,3,3,3
140 DATA192,192,192,255,255,192,192,192
150 DATA0,0,0,255,255,0,0,0
160 DATA3,3,3,255,255,3,3,3
170 DATA3,3,3,3,3,3,3,3
180 FORX=0TO63:READA:POKE&HF400+X,A:NEXTX:RETURN:REM CHRS 128-135
190 DATA128,26,8,128,26,8,128,26,8,129,130,131,26,8,8,8,132,133,134,26,8,8,8
200 DATA128,32,135
210 CH$="":FORX=1TO26:READA:CH$=CH$+CHR$(A):NEXTX:RETURN
```

```
1 LPRINT CHR$(27);"N";CHR$(6)
2 LPRINT CHR$(27);"Q";CHR$(70)
3 LLIST
4 END
8 CHAR4
10 RANDOM:CLEAR (20):DIMP(4,8),S$(4),R$(5):X=49
15 'CONVERTED TO COLOUR GENIE BY                    DON  EDWARDS
20 COLOUR12:CLS:PRINTCHR$(23):PRINT§129,"?     WHODUNNIT     ?": FOR I
=9 TO 29STEP2:PRINT§40+I,"?":PRINT§200+I,"?":NEXT I
21 PRINT§89,"?";TAB(29)"?";:PRINT§169,"?";TAB(29)"?";
22 I$=INKEY$:'CLEAR KEYBOARD
25 COLOUR4:PRINT§642,"PRESS <I> FOR INSTRUCTIONS":PRINT§778," PRESS <B
> TO BEGIN"
27 GOSUB 4000:I$=INKEY$:IFI$="B" THEN 90 ELSEIFI$<>"I"THEN 27
30 COLOUR12:CLS:PRINT§10,"W H O D U N N I T ?":PRINT
32 COLOUR4:PRINT"THE OWNER OF HUNTLEY MANOR HAS BEEN     MURDERED.   I
NITIAL INVES";
34 PRINT"TIGATIONS SHOW THAT THE MURDERER WAS ONE OF THE FIVE   GUESTS
 STAY";
36 PRINT"ING AT THE MANOR AND THAT THE MURDER TOOK PLACE IN ONE OF THE
 ";
38 PRINT"6 ROOMS IN  THE WEST WING BETWEEN THE HOURS OF  1-PM  AND 9-P
M.    ";
40 PRINT"YOUR JOB AS INSPECTOR IS TO DEDUCE  WHO  MURDERED  THE HOST,"
42 PRINT"WHERE THE MURDER TOOK  PLACE AND AT WHAT HOUR.";
44 PRINT"   YOUR SCORE DEPENDS ON HOW LONG IT TAKES YOU.  IF A SUSPECT
 ";
46 PRINT"MENTIONS THE HOST THEN YOU HAVE FOUND THE  ROOM      (UNLESS T
HE'RE ";
48 PRINT"LYING). YOU  CAN QUESTION  AND GUESS AS OFTEN AS YOU LIKE.  I
F A"
50 PRINT"SUSPECT SAYS  THEY SAW SOMEONE THEN THEY MUST HAVE BEEN IN A
";
52 PRINT"ROOM NEXT DOOR      (NOTICE WINDOWS IN FLOOR  PLAN)."
54 PRINT"   ALTHOUGH THE INNOCENT SUSPECTS WILL   BE HONEST, THE KILLE
R ";
56 PRINT"MAY LIE TO COVER HIS/HER CRIME.":COLOUR1:PRINT§931,"PRESS <B>
 TO BEGIN:";
58 I$=INKEY$:IFI$<>"B"THEN 58
90 FOR A=0  TO 4:READS$(A):NEXT:FOR A=0TO5:READR$(A):NEXT
95 DATAKEN,MARY,PAUL,SUE,JOHN,CONSERV,KITCHEN,LIVING,DINING
97 DATATROPHY,GAMES
100 DATA153,233,313,393,473,422,412,402,82,93,103
110 FORI=0 TO 4:READSS(I):NEXT I:FORI=0 TO 5:READRR(I):NEXTI
314 COLOUR6:CLS:FORI=0TO5:PRINT§RR(I)+1,R$(I);:NEXTI
320  COLOUR2:PRINT§0,CHR$(129);:FORX=1TO30:PRINT§X,CHR$(209):NEXTX
325 PRINT§10,CHR$(129):PRINT§21,CHR$(128):PRINT§31,CHR$(128)
330 FORX=40TO440STEP40:PRINT§X,CHR$(213):NEXTX
335 FORX=50TO450STEP40:PRINT§X,CHR$(213):NEXTX
340 FORX=61TO461STEP40:PRINT§X,CHR$(212):NEXTX
345 FORX=71TO471STEP40:PRINT§X,CHR$(212):NEXTX
346 PRINT§200,CHR$(131):FORX=201TO230:PRINT§X,CHR$(210):NEXTX
347 PRINT§280,CHR$(129):FORX=281TO310:PRINT§X,CHR$(209):NEXTX
350 PRINT§480,CHR$(131):FORX=481TO510:PRINT§X,CHR$(210):NEXTX
355 PRINT§490,CHR$(131):PRINT§501,CHR$(130):PRINT§511,CHR$(130)
360 POKE17498,32:POKE17509,32:POKE17617,32:POKE17618,32
365 POKE17629,32:POKE17630,32:POKE17658,32:POKE17669,32
370 POKE17697,32:POKE17698,32:POKE17709,32:POKE17710,32
380 POKE17818,32:POKE17829,32
390 COLOUR4:PRINT§73,"GUESTS:";:COLOUR6:FORI=0TO4:PRINT§SS(I)+1,S$(I);
:NEXT
```

```
400 COLOUR13:PRINT§560,"HOST MURDERED BETWEEN 1 AND 9 PM."
460 FORK=0TO4:P(K,0)=RND(6)-1:NEXT:FORL=1TO8:FORK=0TO4
520 A=RND(6)-1:IFA=P(K,L-1)THEN520
530 P(K,L)=A:NEXTK,L:M=RND(5)-1:T=RND(9)-1:R=P(M,T)
610 COLOUR15:PRINT§553,:PRINT§560,"INSPECTOR, WHO DO YOU WISH TO QUEST
ION"
612 I$=INKEY$:IF I$=""THEN 612
615 S=5:FOR I=0 TO 4:IFI$=LEFT$(S$(I),1) THEN S=I
617 NEXT I:IF S= 5 THEN 612
620 S1$=S$(S):PRINT§805,;S1$:FOR I =1 TO 100:NEXTI
650 COLOUR13:GOSUB 3000:PRINT§560,"DO YOU WISH TO QUESTION ";S1$" ABOU
T *       "
660 PRINT"1- WHEREABOUTS AT A CERTAIN TIME"
670 PRINT"2- TIME SUSPECT WAS IN A CERTAIN ROOM"
680 PRINT"3- KNOW WHO WHERE OR WHEN"
685 I$=INKEY$:IFI$=""THEN 685
690 A=VAL(I$):IFA<1 ORA>3THEN 685ELSEC=C+1
700 GOSUB 3000
710 ONAGOTO 720,990,1200
720 PRINT§600,S1$;",WHERE WERE YOU AT ? O'CLOCK?       "
730 I$=INKEY$:IFI$=""THEN 730
740 T1=VAL(I$):IFT1<1ORT1>9 THEN 730
750 PRINT§600,S1$;", WHERE WERE YOU AT";T1;"O'CLOCK ?        "
760 T1=T1-1:R1=P(S,T1):IF S<>MTHEN 860
790 IF RND(4)<1 THEN 910
800 R1=RND(6)-1:IF RND(8)=4 THENT2=10 ELSE T2=0
820 GOTO 890
860 T2=T1:IFR1<>RTHEN 910
880 IF RND(10)=1 THEN 910
890 IF T2<T THEN PRINT"THE HOST WAS STILL ALIVE."
900 IF T2>T THEN PRINT"THE HOST WAS ALREADY DEAD"
910 PRINT"I WAS IN THE ";R$(R1);" ROOM."
920 FOR K=0 TO 4:IF K=S THEN 960
940 IF P(K,T1)=R1PRINT"I WAS WITH ";S$(K);". ";
950 IF ABS (R1-P(K,T1))=1 PRINT"I SAW ";S$(K);". ";
960 NEXT K:GOTO 610
990 PRINT§600,S1$;", WHAT TIME WERE YOU IN <ROOM> "
1000 I$=INKEY$:IF I$=""THEN 1000
1010 GOSUB 2000:IF X=6THEN 1000
1015 PRINTR$(X)
1020 IFS<>M THEN 1110
1050 T1=RND(9):GOTO 1090
1070 IFT1=F PRINT"I WAS NOT IN THAT ROOM.":GOTO 610
1090 PRINT"I WAS IN THAT ROOM AT";T1;" O'CLOCK":FORX=1TO100:NEXTX:GOTO
 610
1110 K=0:FORB=0 TO 8:IF P(S,B)<>X THEN 1170
1140 IFK=0PRINT"I WAS IN THAT ROOM AT";B+1;:K=1:ELSE PRINT",  ";B+1;
1170 NEXT:IFK=0PRINT"I WAS NOT IN THAT ROOM."
1180 GOTO 610
1200 C1=C1+1
1220 GOSUB 3000:PRINT§560,"DO YOU KNOW: 1- KILLER, 2- ROOM, 3- TIME  4
- BAFFLED "
1230 I$=INKEY$:IF I$=""THEN 1230
1240 A=VAL(I$):IF A<1ORA>4 THEN 1230
1260 ONAGOTO1265,1410,1340,1550
1265 IFH=1THEN C1=C1-1:PRINT"YOU ALREADY KNOW THE KILLER IS "S$(M)".":
FORX=1TO400:NEXTX:GOSUB3000:GOTO 610
1270 COLOUR5:PRINT"  THE KILLER IS ?  ";
1272 I$=INKEY$:IFI$=""THEN 1272
1280 X=5: FOR A=0 TO 4:IFI$=LEFT$(S$(A),1) THEN X=A
1285 NEXT A:IFX=5 THEN 1272
```

```
1287 PRINTS$(X)
1300 IFM<>X THEN 1530
1310 PRINT"  YOU HAVE THE KILLER INSPECTOR.":H=1
1315 COLOUR3:PRINT§SS(M),"*";
1316 FORX=1TO900:NEXTX:GOSUB3000
```

```
1320 IFH=1ANDH1=1ANDH2=1 THEN 1470 ELSE 610
1340 COLOUR13:IFH1=1 THEN C1=C1-1:PRINT"YOU ALREADY KNOW IT WAS AT";T+
1;"O'CLOCK":GOTO1316
1350 COLOUR5:PRINT"  TIME OF MURDER ?";
1360 I$=INKEY$:IFI$=""THEN 1360
1365 T1=VAL(I$):IFT1<1ORT1>9 THEN 1360
1367 COLOUR13:PRINTT1;"O'CLOCK":FORX=1TO500:NEXTX
1370 IFT1-1<>T THEN 1530
1375 GOSUB3000
1380 PRINT"  INSPECTOR YOU HAVE THE RIGHT TIME.":H1=1
1385 COLOUR3:PRINT§250,T+1"O CLOCK"
1390 COLOUR13:PRINT§560,"HOST WAS MURDERED AT";T+1;"O'CLOCK.        ";:
GOTO1316
1410 GOSUB3000:IFH2=1 THEN C1=C1-1:PRINT" YOU ALREADY KNOW IT WAS IN T
HE ";R$(R);"   ROOM.":GOTO 610
1420 COLOUR5:PRINT"ROOM OF MURDER ?  ";
1422 I$=INKEY$:IFI$="" THEN 1422
1430 GOSUB 2000:IFX=6 THEN 1422
1440 COLOUR13:PRINTR$(X)
1450 IFX<>R THEN 1530
1455 COLOUR3:PRINT§RR(X)+43,"*";
1460 COLOUR13:PRINT§680,"  INSPECTOR,  YOU HAVE THE RIGHT ROOM.":H2=1:
GOTO1316
1470 CLS:PRINT§121,"YOU'VE SOLVED THE MURDER CASE INSPECTOR"
1505 S=100-C-8*(C1-3): IFS<0 THEN S=0
1507 S%=S
1508 BL$="":IFC1-3>1THEN BL$="S"
1510 PRINT"IT TOOK YOU ";C;"QUESTIONS AND ":PRINTC1-3;" BLUNDER";BL$;"
."
1512 CM$="TERRIBLE.":IF S>30 THEN CM$=" NOT VERY GOOD."
1513 IFS>49 THENCM$="NOT TOO BAD.":IFS>62 THEN CM$="MEDIOCRE."
1514 IFS>75 THEN CM$="REASONABLY SWIFT.":IFS>85 THEN CM$="A NEW AMOUNT
 OF SWIFTNESS."
1515 PRINT:PRINT" THAT GIVES YOU A SCORE OF ";S%;"%.":PRINT: PRINT" TH
ATS ";CM$
1520 GOTO 1590
1530 PRINT"  INSPECTOR,  YOU ARE HIGHLY INCOMPETENT  TRY AGAIN."
1540 FORX=1TO900:NEXTX:GOSUB3000:GOTO 610
1550 PRINT§600," TOO BAD INSPECTOR.":PRINT" THE FACTS ARE:"
1570 PRINTS$(M);" KILLED THE HOST AT";T;" o'CLOCK IN   THE ";
1580 PRINTR$(R);"  (ROOM)"
1590 COLOUR1:PRINT:PRINT" DO YOU WANT A NEW CASE ?";
1595 I$=INKEY$:IFI$="Y"THEN RUN ELSE IF I$<>"N" THEN 1595
1600 CLS:END
2000 X=6:FOR A=0 TO 5:IFI$=LEFT$(R$(A),1) THENX=A
2010 NEXT:RETURN
3000 PRINT§560,CHR$(31)
3010 RETURN
4000 COLOUR12:POKE 17408+X,32: FOR J=1 TO 10: NEXT J:POKE 17408+X,63
4010 IFX< 69 THEN X=X+2:RETURN:ELSE IFX>209 THEN X=X-2:RETURN
4020 IF(X=209)OR(X=169)OR(X=129)OR(X= 89)THEN X=X- 80
4030 X=X+40:RETURN
```

RANDOM MORSE CODE GENERATOR from Dick Collins

This is a program which generates random morse code (sound) in 5 character groups. It produces characters in the range A - Z and 0 - 9.

The speed is set by inputting a number e.g. 30 for 9 words per minute.

Pitch change is in line 2000. Play (1,4,4,15) can be: Play (1,4,7,15) for a better (higher) note.

```
1Ø REM RANDOM MORSE CODE GENERATOR by R.J.COLLINS,PARAPARAUMU
2Ø INPUT"SPEED (15-17wpm;2Ø-12wpm;3Ø-9wpm;1ØØ-SLOWER THAN LEARNER)";S
25 A=Ø:B=-1
3Ø RANDOM
4Ø A=RND(36):FOR Z=1 TO (3*S):NEXT Z
5Ø B=B+1:IF B=5GOTO1ØØØELSE6Ø
6Ø ON A GOTO 5ØØ,5Ø1,5Ø2,5Ø3,5Ø4,5Ø5,5Ø6,5Ø7,5Ø8,5Ø9,51Ø,511,512,513,514,515,516
,517,518,519,52Ø,521,522,523,524,525,526,527,528,529,53Ø,531,532,533,534,535
5ØØ TD=S:GOSUB2ØØØ:TD=S*3:GOSUB 2ØØØ:GOTO3Ø
5Ø1 TD=S*3:GOSUB 2ØØØ:TD=S:GOSUB 2ØØØ:GOSUB 2ØØØ:GOSUB 2ØØØ:GOTO 3Ø
5Ø2 TD=3*S:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:TD=3*S:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO  3Ø
5Ø3 TD=3*S:GOSUB 2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOTO 3Ø
5Ø4 TD=S:GOSUB2ØØØ:GOTO3Ø
5Ø5 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:TD=3*S:GOSUB2ØØØ:TD=  S:GOSUB2ØØØ:GOTO3Ø
5Ø6 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO3Ø
5Ø7 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
5Ø8 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
5Ø9 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
51Ø TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
511 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
512 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
513 TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO3Ø
514 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
515 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO3Ø
516 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
517 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO3Ø
518 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
519 TD=S*3:GOSUB2ØØØ:GOTO3Ø
52Ø TD=S:GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
521 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
522 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
523 TD=S*3:GOSUB2ØØØ:TD=S;GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
524 TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
525 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
526 TD=S:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
527 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
528 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
529 TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:TD=S*3:GOSUB2ØØØ:GOTO3Ø
53Ø TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
531 TD=S*3:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
  532 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø:REM 7
533 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
534 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:TD=S:GOSUB2ØØØ:GOTO3Ø
535 TD=S*3:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOSUB2ØØØ:GOTO3Ø
1ØØØ B=-1:FOR G=1 TO (6*S):NEXT G:GOTO 3Ø
2ØØØ PLAY(1,4,4,15):FOR D=1TO TD:NEXT D:PLAY(1,4,4,Ø):FOR G=1 TO S:NEXT G:RETURN
```