LYNX USER GROUP

MAGAZINE
Vol. 1 Issue 4.

CONTENTS
~ ~ ~ ~ ~ ~ ~ ~

It has come to my attention that some people have certain "moans" concerning the contents of our magazine, also that others are loaning their magazine to friends rather than encouraging them to join the USER GROUP. I must point out that because there are different "kinds" of owners, a balance of content is quite a difficult task. If you have any suggestions concerning what you would like to see included then PLEASE tell me. Some of the criticisms are: a) Too many reviews, b) Not enough games, c) Not enough CP/M material, d) Too heavy on M/C code articles, e) More on M/C for beginners, etc. I will try to answer these comments:-

a) A lot of people want reviews, because this is the only way that any software can be assessed, as most of it is totally unavailable at any retail outlet. (Hopefully this will change).

b) One point I must make is that a very high proportion of the group are capable of writing code, and are simply not interested in games. However I will try to include at least one game per issue from now on.

c) Concerning CP/M, i.e. 128K owners, I have to point out that you are at present, in a minority. There are as many 128K owners as those who are "into" FORTH! Also I can only publish what I receive!

d) This ties up with b)! As I said it is difficult trying to achieve a balance!

e) This is one area which I personally would be most interested in. However in spite of being promised material from three sources, articles on LYNX M/C, nothing has appeared! So......

Finally to those who loan their magazine to friends or even supply copies of the mag. to others, you are cheating yourself (you paid money for it)! The LYNX is NOT in the position of other micros, i.e. 10s or 100s of thousands owned, where a certain amount of copying doesn't greatly affect the overall "health", WE ARE IN A MINORITY! There is little support for the LYNX at present, certainly the trade and software houses are VERY reluctant to have any dealings with the LYNX. This also applies to copying software of any form. As an example, great effort was put into obtaining and writing JETSETWILLY from SOFTWARE PROJECTS. Bear in mind that software is only "put" on a computer if it is COMMERCIALLY VIABLE and not because the machine is great or you personally regard it as the best micro around. It is ALL to do with MONEY!
SUPPORT YOUR LYNX and the LYNX USER GROUP.
For these reasons, I am therefore amending the aims to read:-

5. LUG is now affiliated to the ACC.
6. Owners who want to get in touch with their nearest members, only upon written request.
7. Deleted.

To change the subject, several people have asked when is the next show. As there is so little new material for the LYNX at present (although various "things" are in the pipeline), there is little point in having a second show until new products or software become available. However I can say that "things" are coming, and the estimate is, subject to solving certain problems, that a second show is likely nearer to Christmas, probably in November or maybe the early part of December.

Due to certain "parties" and a mix-up at CAMPUTERS liquidation, valuable data and documentation have been lost. What I can say however is that a honary member (ex-CAMPUTERS) who wrote the hardware manual for the 96K LYNX, has agreed to re-write the manual again for his benefit (money) and for the benefit of other owners. So please be patient everything for the LYNX will appear in due course.

Following on from the above, one member is now working "flat out" to re-generate an ARABIC LYNX and is co-operating with both ANSTON TECHNOLOGY Ltd and the linguist who originally worked with CAMPUTERS.

As I was present at the PCW show at OLYMPIA in September, there was still a lot of interest in the LYNX, both from owners and the trade from here and overseas. ABERSOFT who produced CAMFORTH and the first PAC-MAN game for the LYNX, have stated that they are still interested in doing more software. We also require a volunteer, as INTERCEPTOR SOFTWARE want 6 games taken over from the COMMODORE 64. Amongst these is a good graphic game called ARABIAN NIGHTS, so.. if you are familiar with 6502 code and would like to earn some money, please get in touch immediately! LEVEL 9 have a "glut" of COMPASS which they would like to sell, please mention LUG and you should get a good discount. What was demonstrated at the show, was the prototype SIDEWAYS ROM board plus some of the new graphic commands for ✗SUPER LYNX✗, at present only on disk. Also an extra feature of fast screen printing, also to go in ✗SUPER LYNX✗. To give you some idea, the normal printing in 40 columns under LYNX BASIC is approximately 300 characters/second, what was demonstrated, was printing in 32 columns at 2000 characters/second and even this can be improved to over 4000 characters/second. All this is still on the 96K using a clock of 4MHz! When this is transferred to the 128K LYNX, it means that with a clock of 6MHz, screen printing will be at over 6000 characters/second! Although slightly bugged at present, we now have a new DRAW routine which is 3 times faster than the original. Stay with your LYNX and your patience WILL be rewarded. A new flight simulator was also shown, this is DELTA WING from CREATIVE-SPARKS SOFTWARE, this should be readily available for CHRISTMAS.
R B JONES

MP165 Dot Matrix Printer
....

This is a fast high resolution dot matrix printer, which is only imported by MICRO-PERIPHERALS Ltd. What will be of interest to LYNX owners, is that it appears to be compatible with every model of the LYNX. No patches!

Although not quite in the league of a cheap printer, nevertheless it is a perfect copy of the EPSON FX80, providing all the controls and text characters of that infamous printer at much lower cost. All the EPSON codes are catered for even the range of characters from other countries. It has NLQ quality at 75 chars/sec and a fast operation of 165 chars/sec hence its name! A sample of the character set is reproduced opposite:-

The matrix is either on a 9x9 basis or on a 17x17 high resolution mode. It was quite pleasing to see an accurate dump of the screen graphics using the EXT VPRINT command on the 128K without having to fiddle about with setting patches. It worked from switch-on. There are two DIL switch sets provided, one which is accessible from the rear (although rather fiddly to get at) for most of the normal configuration requirements and a second one which is primarily intended to be set up for other countries' character sets. The printer is only CENTRONICS parallel standard but thanks to its operating speed, compares very well with a serial line printer. Both tractor and friction drive are provided for and also a LIGHT/DARK control.

Other aspects of its specification are:-
Line feeds in 1/6", 1/8", 7/32", n/72, and n/216 increments, reverse line feed is also catered for.
Character sets in High speed PICA, High density PICA, ELITE, Condensed and Proportional modes, as well as normal, italic, double width etc.
Head life quoted at 100,000,000 operations!
Paper width from 4" to 10".
International character sets are USA, FRANCE, GERMANY, UK, DENMARK, SWEDEN, ITALY, and SPAIN as well as ASCII.

A very comprehensive manual is supplied, which gives all details of obtaining the character sets under software control plus details of the CENTRONICS connector, timing diagrams and even a block diagram of the circuitry.

As I obtained this for another member, I was unable to put it completely through all its features, it would have taken too long, but it left me with envy (I can't afford it yet!) and I can only strongly recommend it if you are in the market for an excellent piece of engineering at a reasonable price. 10/10.
R B JONES

```
;<=>?@ABCDEFGH
'9:;<=>?@ABCDE
=>?@ABCDEFGHI
;<=>?@ABCDEF(
>?@ABCDEFGHIJK
@ABCDEFGHIJKLMN(
JKLMNOPQRSTUVWXYZ[\]^
;=>?@ABCDEF(
9:;<=>?@
;9:;<=>
>?;;<=>?@
;<=>?@
<=>?@AB
<=>?@AB
>?@ABCD
DEFGHIJKLMN(
ABCDEFGH
JABCDEF
;CDEFGH
WXYZ[\]^_`abcdefghijk.
OPQRSTUVWXYZ\]^
HNOPQRSTUVWX>
```

HOME ACCOUNTS SYSTEM by RAD SYSTEMS
....

This is a cassette based accounts system. The instructions consist of a single A4 sheet of paper telling you how the program loads in two parts: i) SET UP ACCOUNTS FILES and ii) HOME ACCOUNTS SYSTEM. The program is designed to help you keep a month by month record of your finances.

After loading Set Up Accounts Files, which takes about three minutes, the screen displays the instructions. The tape recorder is set up for RECORD and then "any key is pressed". If you have a remote facility on your recorder, then the recorder starts automatically, otherwise you will have to do this yourself. The screen instructions tell you which month is being created. On completing each month a beep is sounded and the next month file is formed, each taking about one minute to SAVE. It is possible to list this program and on doing so many more instructions can be found. There should be some mention of this facility in the written instructions. All the REM statements can be deleted and this would greatly speed up the LOADing time.

Having set up the Accounts Files you then have to LOAD "HOME ACCOUNTS", which takes about four minutes. The first page of instructions asks you to enter the month, and at the bottom of the screen there is a command line, which also shows Help and Exit. The month file is then loaded, and the screen shows four headings; ENTRY, DETAILS, DEBITS, CREDITS and a BALANCE. All screen inputs are entered on the command line when prompted to do so. This enables them to be edited before they are included in the accounts. If you are in difficulty, a HELP screen can be produced by typing "H". Over a hundred entries may be made. To enable this number of entries to be viewed the screen may be scrolled up or down by using the arrow keys, or by typing "T" for top and "B" for bottom. Additions and Deletions may be made to any item. As each entry is made the Balance keeps a running total. Once all the data is entered, you Exit the program and are prompted as to whether you wish to SAVE the data or not. If you do, the tape recorder has to be set and the file is SAVEd, and the program ends.

Generally I thought this program lacked style, the presentation was poor, there being no sleeve to the cassette and a lack of documentation. The program worked well, and appears to be bug free. If you have a very large budget each month or own a disk drive perhaps this program will be of use. However for the average home, I feel using a calculator would be a quicker way of working out your finances. Apart from these criticisms, the program was well written in BASIC and being able to copy the tape is a great help. There are many useful routines in the program for a programmer to use. The cost was £7.95 and overall I award the program 7/10.
A.R.BRISTOW.

"GETTING THE MOST FROM YOUR LYNX"

Books devoted solely to the LYNX are so rare that the appearance of this one comes as a pleasant surprise. As LYNX buffs are only to well aware, it is unlikely that there will be many more books to add to the LYNX's library. Therefore owners are virtually certain to buy this present one, but is the book really worth having? If the title is an accurate indicator, it is! "GETTING THE MOST FROM YOUR LYNX" implies a wide and complete coverage of the machine's capabilities, but the slimness of the volume (128p) raises immediate doubts.

A quick glance at the chapter headings reveals that much of the book is devoted to LYNX BASIC. Much that is covered in the CAMPUTER's USER MANUAL is repeated, but with many of the features of the language being treated in greater detail. Short programs illustrate their use, e.g. FOR......NEXT loops, and commands such as STR$ and TEXT, absent from the USER MANUAL, are described, as well as multi-dimensional arrays.

Chapter 6 entitled Graphics and Sound, is one of the most absorbing sections, and includes sections on high resolution and user defined graphics. Methods of filling rectangles and triangles with colour are desribed, and the drawing of a chess board and creation of chess pieces are used to illustrate some of the techniques available. The value of the PROTECT command is made very clear in the chess graphics program and the effect that this instruction has on the subsequent use of the various colours is also explained.

As for the use of sound, BEEP is explored quite thoroughly and a useful table of frequencies and wavelengths of musical notes is presented. On the other hand, the SOUND command is dealt with inadequately. Examples show how to store "sounds" in the memory and recall them, but the way in which the required parameters are arrived at, is not explained.

A special chapter is devoted to hints and tips on such things as scrolling, altering the auto-repeat rate (and even switching it off), novel ways of listing programs, writing to the alternative green bank, retrieving programs after NEWing them and recovering from crashes. Needless to say, connecting the machine to TV sets, monitors and cassette players is also described and hints are given on overcoming LOADing problems and faulty TV displays. Brief information on add-ons and how a computer works, is also included in seperate chapters.

Most of the material in the book applies to the 48K LYNX but the final chapter concentrates on the 96K and 128K versions. Short accounts of their extra capabilities are given, along with a description of their new commands.

Presumably discussion of the LYNX monitor and the monitor commands have been omitted for want of space. This is unfortunate in a volume whose title implies comprehensiveness, but just as serious, if not more so, is the lack of an index. In the LYNX USER MANUAL, the Contents list serves as a reasonable alternative but this is not so in the book under review. Indeed the way in which the subject matter is arranged makes an index imperative if this new book is to be used as a work of reference.

In spite of the afore-mentioned shortcomings Steven Jedowski and Penguin Books are to be congratulated on producing a book for a machine which, sadly, many other authors and publishers have ignored. It is well worth having. A number of useful and interesting matters are included which have not appreared in the previous books and will point the reader in some of the right directions for ... getting the most from their LYNX.

A Rendall.

TREASURE ISLAND

The first part of this graphical adventure only takes a short time to load, this auto-runs and produces a picture of Captain Long John Hook with his treasure chest beside him. The computer starts to play "What shall we do with the drunken sailor?" as well, although slightly off key. The LYNX asks you if you require instructions; if you do, it gives an explanation of the game and the four verbs which you can use:- "take", "drop", "use", and " " (do nothing)! This although it seems very limited, is also quite flexible. The instructions also give you one or two very useful clues. After this you are taken back to the first screen again and asked to load the second program.

The second part is also quite quick to load, it auto-runs and you are at the first location in the adventure. The adventure is part graphic, part animated and these on the whole are quite reasonable, however they could not be called brilliant by any means! The puzzles are for the most part, quite good with one or two very good ones.

There are, as far as I know, about nineteen or twenty locations though I have only been able to get to sixteen of them. (How do you get past the green fog?). I bought the game from Phoenixx Software and it cost £2.99 which, in my view is very good value indeed. hence for value=9/10 and overall=7/10.

R Harries-Harris.

LINE WRAPPER
....

I bought a Parallel Printer interface for my LYNX but was not wholly satisfied with the software provided, so I disassembled the print routine to see how it worked. One of my gripes had been that upon LISTing a long line on my printer the print head reaches the right hand end, then hammers away, producing a neat black dot on the rubber roller, instead of wrapping the line over. To my surprise I find that this facility has been included in the software, but not properly implemented - all it needs is a few bytes of machine code to rectify this.

The following routine works very well:-

```
F5 C5 3A C1 61 47 3A C2 61 BF 20 05 3E 0
D CD 26 9F C1 F1 C3 26 9F
```

This routine is fully relocatable so it can be used from a CODE line or entered through the monitor. If the latter method is used it can be tacked onto the bottom of the print routine, the whole thing dumped to cassette and then loaded as a package. The routine above simply compares two of the bytes used by the parallel print routine and issues a carriage return when they are the same. To initialise it &6202 has to point to the start of the routine, and &61C1 initialised to the required length. This can be done by POKEing the value in or modifying the parallel print loader which initialises this byte to 80 anyway. -See NILUG Issue 6 "Interfacing non-standard printers." for instructions on how to break into the software.

I don't know if the routine will work on a 96K LYNX but would be interested to know.
J.S. Colombo.

FOR SALE
....

LYNX Technical Information
....

The SERIAL PORT

There are two distinct serial systems on the LYNX computers. The 48K and 96K machines have simple systems that run at 2400 Baud and are fixed at one start bit, one stop bit, no parity and runs with eight bit data without handshaking. The 128K Laureat machine however runs a software configurable serial system that does support handshaking. Its maximum baud rate is 78.4 Kbaud, which should be fast enough for most applications.

The systems on the 48K and 96K machines are driven by a 6402 UART device which is socketed on the main board. Therefore it is possible to remove this device and insert a "piggy-back" board to allow full use of the UART's flexibility regarding data protocols.

The final point on this matter is to note that the clock for the UART passes through the links at the rear right hand quarter of the main board. The link should normally be on the pins nearest the front of the computer. The rearward pin pairs provide 4800 and 9600 baud data rates. Therefore if the only changes required of the serial system is that it should run at a lower baud rate, then it is possible to effect this by inserting a dividing network between these pins.

Z80 PORTS

The Z80 I/O ports on the LYNX computer have been organised in such a way as to allow certain locations to be used for home enthusiasts projects without fear of interfering with Computer's peripherals. To this end, port locations 840 to 84F have been reserved for the hobbyist.

The operation of the I/O ports warrant some explanation. When an OUT instruction is performed that does not correspond to certain internal ports, the RD and EWR (early write) lines at the expansion connector are not activated and the address bus is not multiplexed. Therefore it is not necessary for the port decoding logic to either demultiplex the address bus nor for it to observe the RD and EWR signals. Note that the low byte of the address bus consists of A0 to A6 and A13, not simply A0 to A7.

One point that should be noted as a result, it is not possible for external READ and WRITE ports to occupy the same physical addresses. It is suggested that for systems requiring both input and output ports, A0 may be used in place of the missing RD/WR signals.

In general it is anticipated that the A13 line will be ignored due to the requirement that the C register be set before an OUT instruction, if it is to carry useful information.

Note that during memory access cycles, both the RD and EWR signals are active but that the EWR signal is an Early WRite signal and is not the WR signal direct from the CPU.
Simon Roberts.

The EXT COMMAND

The following article was to appear in LYNX USER (3), which of course was never produced. Although there was no supplied sample program, it is to be hoped that at least some people in the group will find the following information of interest. (Editor).

The following demo program shows how to add new EXT commands to LYNX BASIC.

EXT commands are tokenised by the EXT token (&41) followed by a second byte which can have any value. EXT commands are divided into 4 groups according to the value of the second token byte:-

&00 to &3F	Available to users.
&40 to &7F	3rd internal ROM (&4000 to &5FFF).
&80 to &BF	2nd XROM (&C000 to &DFFF), disk utilities.
&C0 to &FF	1st XROM (&E000 to &FFFF), includes Disk commands.

The 3rd internal ROM contains routines for handling the EXT token, so except on the 48K LYNX, the user need not supply these.
The user must supply the following:-

1) A list of ASCII command words.
2) A list of addresses for syntax checking.
3) A list of addresses for execution.
4) A 3-word block containing addresses of lists.

Requirements for the ASCII list 1) are:-
(a) The first byte gives the number of words.
(b) The first character in each word has BIT 7 set.
(c) The list is terminated by a byte &80.

The block 4) is pointed to by a word in RAM.
Each group of tokens has its own RAM pointer, i.e.:-

Group	
Group &00 to &3F	&61E6
Group &40 to &7F	&61E8
Group &80 to &BF	&61EA
Group &C0 to &FF	&61EC

On power up, all 4 pointers are set to 0. Before a group of EXT commands can be used, the appropriate pointer must be filled in with the address of the 3-word block. Thus on the 96k and 128K LYNXes, the pointer at &61E8 is set up immediately after power up as part of the 3rd ROM initialisation. Similarly the pointer at &61EC is set up when the disk ROM is called (i.e. XROM (RETURN)).

The demo program shows how to set up three user EXT commands, namely:-

EXT ALPHA	Tokenised as 41 00
EXT BETA	Tokenised as 41 01
EXT GAMMA	Tokenised as 41 02

Addr.	Obj code	Mnemonics	Comments
5000	21 07 50	LD HL,BLOCK	;Routine to
5003	22 E6 61	LD (&61E6),HL	;intialise
5006	C9	RET	
			;Addresses of lists
		BLOCK:	
5007	0D 50	DW ASCII	
5009	1D 50	DW SYNTAX	
500B	23 50	DW EXECUT	
			;List of command words

ASCII:-
&500D	03	DB 3	;Number of ;words
500E	C1 4C 50 48	DB &80 "A","LPHA"	
5012	41		
5013	C2 45 54 41	DB &80 "B","ETA"	
5017	C7 41 4D 40	DB &80 "G","AMMA"	
501B	41		
501C	80	DB &80	;Terminator

;List of syntax checking

SYNTAX:-
501D	29 50	DW SYALPHA
501F	29 50	DW SYBETA
5021	29 50	DW SYGAMMA

;List of execution

EXEC:-
5023	29 50	DW EXALPHA
5025	29 50	DW EXBETA
5027	29 50	DW EXGAMMA

Syntax routines can be written by the user, or if the syntax is the same as for a system command, then the routine in the system ROM can be used.

SYALPHA:-
        code.....
SYBETA:-
        code.....
SYGAMMA:-
        code.....

Routines to execute must be supplied by the user.

EXALPHA:-
        code.....
EXBETA:-
        code.....
EXGAMMA:-
        code.....

From the above it is apparent that the LYNX had the potential to support probably the biggest BASIC the world has ever seen or is ever likely to be seen. If you work it out, it means that under the token techniques described above, the LYNX could support 65,536 BASIC commands and functions! This would of course mean a vast ROM capacity, even more than XXXSUPER LYNXXXX supports (128K increase). It might be interesting if someone would care to suggest (including the existing ones) even a 1000! PHEW!
R B JONES.

FINAL SCORE
....

```
100 DIM N$(10)(89),C$(80),B$(80),D$(80),
E$(80),C(80)
110 LET t=0
120 RANDOM
130 PROC read(80,1)
140 PROC six(80,1)
150 PROC six(40,2)
160 PROC read(64,21)
170 PROC six(64,3)
180 PROC six(32,4)
190 PROC six(16,5)
200 PROC six(8,6)
210 PROC semi
220 PROC final
230 PRINT "Do you want another run ? N/Y
"
240 LET K=GETN
250 IF K=89 OR k=121 THEN  RUN
260 END
70 DEFPROC six(n,r)
280 PROC draw(n)
290 PROC print(n,r)
300 PROC pause
310 FOR x=1 TO n
320   LET C(x)=0
330 NEXT x
340 PROC results(n)
350 PROC pause
360 ENDPROC
370 DEFPROC draw(n)
380 LET C$=""
390 FOR I=65 TO 64+n
400   LET C$=C$+CHR$(I)
410 NEXT I
420 LET B$="",D$="",E$=""
430 WHILE LEN(C$)>1
440   LET R=RAND(LEN(C$))+1
450   IF R=1 THEN  LET E$=B$+LEFT$(C$,1)
,D$=""
460   ELSE  LET B$=B$+MID$(C$,R,1),D$=LE
FT$(C$,R-1)
470   LET E$=RIGHT$(C$,LEN(C$))-R'
480   LET C$=C$+E$
490 WEND
500 LET B$=B$+C$
510 ENDPROC
520 DEFPROC print(n,r)
530 CLS
540 PRINT "Draw for round "r;" follows"
550 FOR I=2 TO n STEP 2
560   LET y=ASC(MID$(B$,I-1,1)),z=ASC(MI
D$(B$,I,1))
570   PRINT RIGHT$(N$(y-64),9);" versus
" TAB 20;RIGHT$(N$(z-64),9)
580   IF I=42 THEN  PROC pause
590 NEXT I
600 ENDPROC
610 DEFPROC pause
620 PRINT @ 27,230;"Press a key to cont
inue"
630 LET X=GETN
640 CLS
650 ENDPROC
```

```
660 DEFPROC results(n)
670 LET f=0
680 FOR I=2 TO n STEP 2
690   LET y=ASC(MID$(E$,I-1,1)),z=ASC(MI
D$(B$,I,1))
700   IF C(y-64)=1 THEN  GOTO 830
710   LET H$=N$(y-64),A$=N$(z-64)
720   LET d=(VAL(LEFT$(H$,1)))-(VAL(LEFT
$(A$,1)))
730   PROC score(-d,t)
740   LET h=s
750   PROC score(d,1-t)
760   LET a=s
770   IF a>h THEN  LET N$(y-64)="Z"
780   IF h>a THEN  LET N$(z-64)="Z"
790   IF a=h THEN  LET f=f+1
800   ELSE  LET C(y-64)=1,C(z-64)=1
810   PRINT  TAB 9;RIGHT$(H$,9); TAB 19;
h; TAB 22;RIGHT$(A$,9); TAB 32;a
820   IF I=42 THEN  PROC pause
830 NEXT I
840 LET t=0
850 IF f>0 THEN  PROC replays
860 ELSE  PROC sort(n)
870 ENDPROC
880 DEFPROC replays
890 PROC pause
900 PRINT "Replays follow"
910 LET t=1
920 PROC results(n)
930 ENDPROC
940 DEFPROC score(b,v)
950 LET s=-6
960 FOR J=1 TO 11+b+(v=0)
970   LET s=s+RND
980 NEXT J
990 LET s=INT(s*1.4+(1.4*(v=0)/2))
1000 IF s<0 THEN  GOTO 950
1010 ENDPROC
1020 DEFPROC sort(n)
1030 LET A=1
1040 FOR k=1 TO n
1050   IF N$(k)="Z" THEN  GOTO 1070
1060   LET N$(A)=N$(k),A=A+1
1070 NEXT k
1080 ENDPROC
1090 DEFPROC semi
1100 PRINT "Draw for semi-finals"
1110 PRINT RIGHT$(N$(1),9);" versus ";RI
GHT$(N$(2),9)
1120 PRINT RIGHT$(N$(3),9);" versus ";RI
GHT$(N$(4),9)
1130 PROC pause
1140 LET D=RAND(5),E=RAND(5)
1150 IF D=E THEN  GOTO 1140
1160 LET F=RAND(5),G=RAND(5)
1170 IF F=G THEN  GOTO 1160
1180 PRINT @ 33,30;"SEMI-FINAL SCORES"
1190 PRINT @ 27,60;RIGHT$(N$(1),9); @
57,60;D; @ 66,60;RIGHT$(N$(2),9); @ 9
6,60;E
1200 PRINT @ 27,80;RIGHT$(N$(3),9); @
57,80;F; @ 66,80;RIGHT$(N$(4),9); @ 9
```

```
6,80;G
1210 IF E>D THEN  LET N$(1)=N$(2)
1220 IF G>F THEN  LET N$(3)=N$(4)
1230 PROC pause
1240 ENDPROC
1250 DEFPROC final
1260 LET D=RAND(5),E=RAND(5)
1270 PRINT @ 42,20;"FINAL SCORE"
1280 PRINT @ 27,40;RIGHT$(N$(1),9); @
57,40;D; @ 66,40;RIGHT$(N$(3),9); @ 9
6,40;E
1290 PROC pause
1300 IF E=D THEN  PROC final
1310 ENDPROC
1320 DEFPROC read(m,u)
1330 FOR i=u TO m
1340   READ N$(i)
1350 NEXT i
1360 ENDPROC
1370 DATA 3Bristol R,3Millwall ,3York Ci
ty,3Bradford ,3Rotherham,3Gillingham
1380 DATA 3Bristol C,3Doncaster,3Hull Ci
ty,3Brentford,3Derby    ,3Reading
1390 DATA 3Walsall  ,3Preston  ,3Burnley
 ,3Wigan    ,3Bolton   ,3Newport
1400 DATA 3Orient   ,3Swansea  ,3Lincoln
 ,3Bournmouth,3Plymouth ,3Cambridge
1410 DATA 4Hereford ,4Chestrfld,4Bury
 ,4Peterboro,4Darlngton,4Blackpool
1420 DATA 4Colchestr,4Port Vale,4Tranmer
e ,4Hartlepol,4Crewe    ,4Stockport
1430 DATA 4Southend ,4Chester  ,4Mansfie
ld,4Swindon  ,4Wrexham  ,4Torquay
1440 DATA 4Scunthorp,4Exeter C.,4Aldersh
ot,4Rochdale ,4Halifax  ,4Nrthmpton
1450 DATA 5Weldstone,5Enfield  ,5Worcest
er,5Boston   ,5Altrncham
1460 DATA 5Barnet   ,5Barrow   ,5Dartfor
d ,5Runcorn  ,5Kiddrmnsr,5Bath City
1470 DATA 5Dagenham ,5Weymouth ,5Scarbor
o ,5Nuneaton ,5Telford
1480 DATA 5Northwich,5Maidstone,5Ketteri
ng,5Frickley ,5Gateshead,5Yeovil
1490 DATA 5Fareham  ,5Gosport  ,5Watrlov
le,5Gravesend
1500 DATA 5Crawley  ,5Hastings ,5Minehea
d ,5Dortchstr,5Trowbridg,5Taunton
1510 DATA 1Arsenal  ,1Tottenham,1Sheff W
ed,1Man.Utd. ,1Nottm.F. ,1Everton
1520 DATA 1Newcastle,1West Ham ,1Sundrla
nd,1Chelsea  ,1Southmptn
1530 DATA 1Aston V. ,1West Brom,1Ipswich
 ,1Q.P.R.   ,1Liverpool,1Norwich
1540 DATA 1Coventry ,1Luton    ,1Leicest
er,1Watford  ,1Stoke
1550 DATA 2Portsmuth,2Blackburn,2Oxford
 ,2Birmngham,2Brighton ,2Shrewsbry
1560 DATA 2Barnsley ,2Man City ,2Leeds U
td,2Grimsby  ,2Fulham
1570 DATA 2Charlton ,2Wimbledon,2Oldham
 ,2Carlisle ,2Sheff Utd,2Middlsbro
1580 DATA 2Huddrsfld,2Wolves   ,2C Palac
e ,2Notts C. ,2Cardiff
```

THE CUBE by D.NESS

A RUBIC Cube game which makes use of the LYNX's colour graphics but which can also be played on a monochromatic display.

The display of the cube consists of two isometric views, the left-hand view showing the front, top and left-hand faces and the right-hand view showing the rear, bottom and right-hand faces from the same viewing point. That is, the insides of the faces as though the cube was transparent.

In order to identify the move required, the cube is divided into 3 slices in the X, Y and Z directions as given below. +ve rotations are clockwise when viewed in the direction of the arrows shown.

FIG.1.
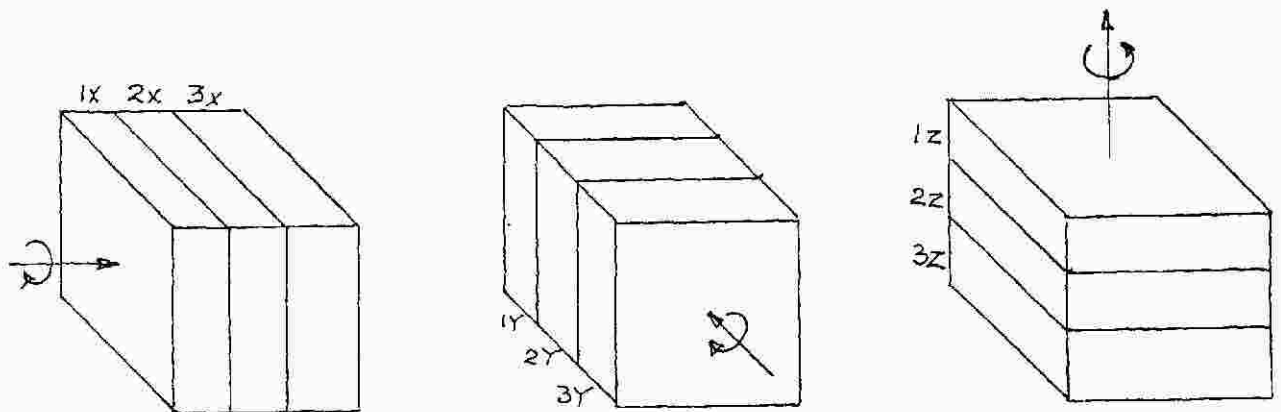
FIG.2.   ACTUAL SCREEN DISPLAY
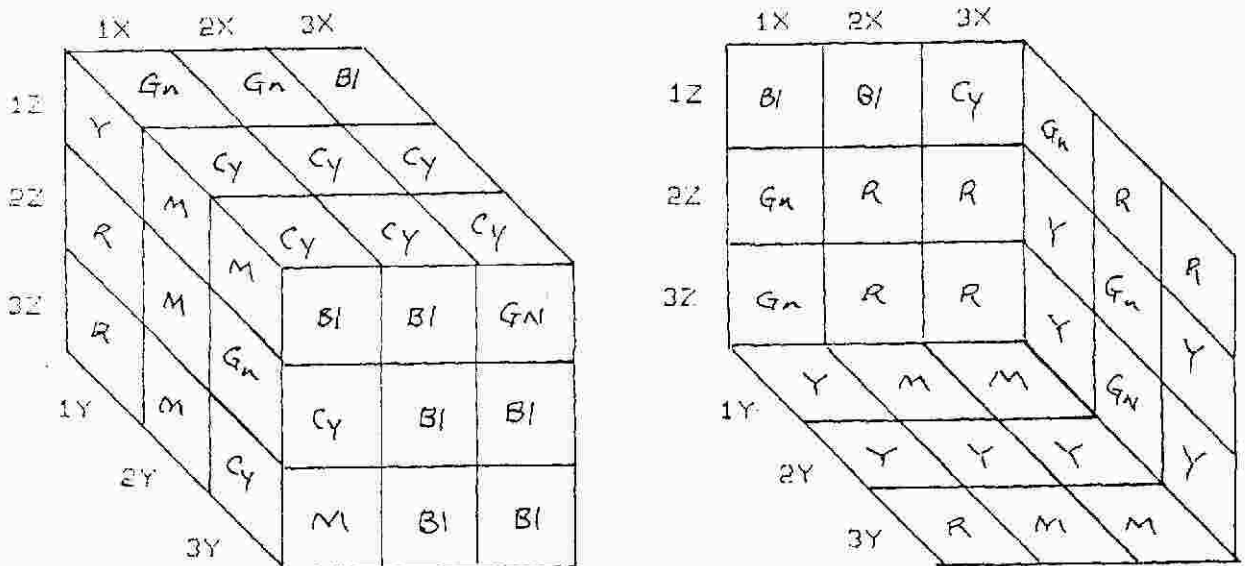
The level of difficulty chosen controls the number of random moves made by the computer from the initial correct position. There is a replay option available which shows the computer's and the player's moves. There are 18 possible moves that can be input at any time during the play. Each slice can be moved in a +ve or -ve direction. Moves are input by entering a three character string such as for example -3Z.
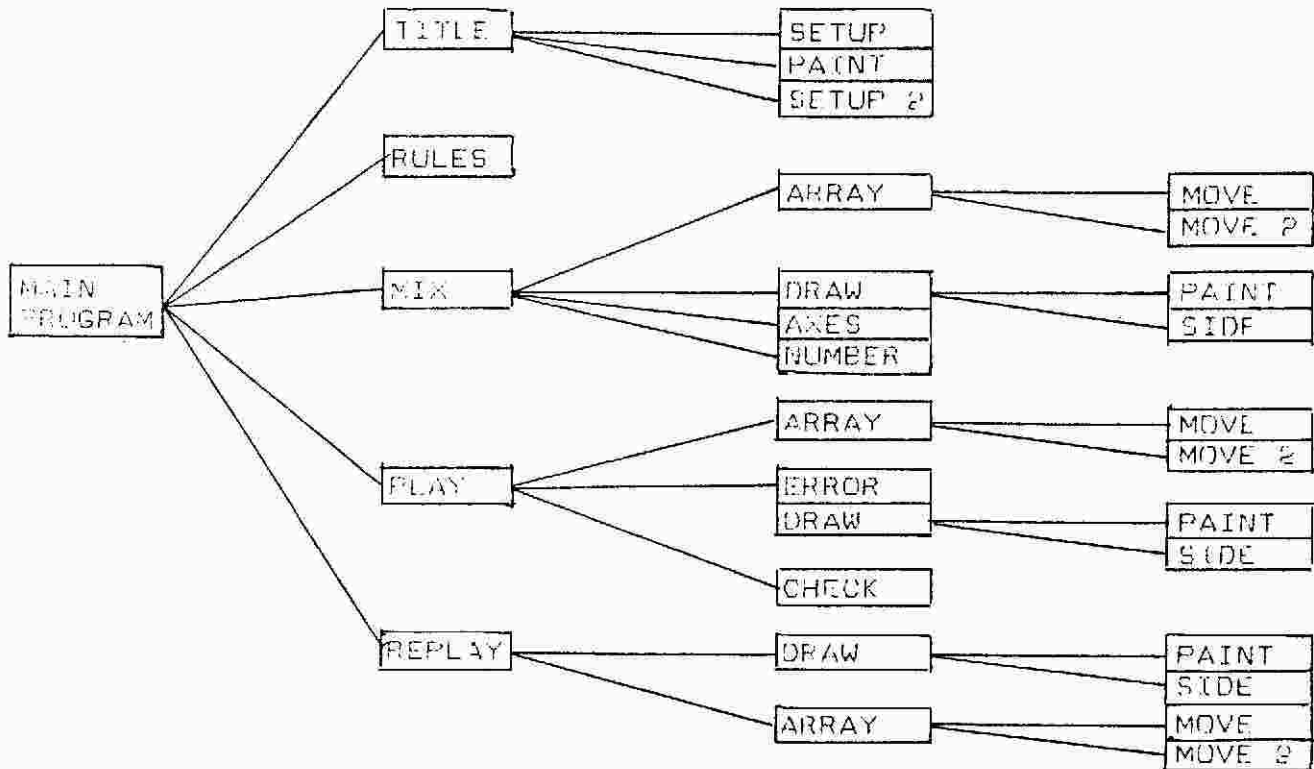
If the number of moves made by the player reaches 3 times those made by the computer the game is terminated. If a B/W display is being used, a number representing the colour square is added to the two views of the cube.

PROGRAM CUBE
`` `` `` `` `` `` `` ``

Flow diagram for the program:-

```
100 PROTECT 0
110 RANDOM
120 WINDOW 0,126,0,248
130 VDU 1,4,2,0,4,24
140 PRINT @ 36,50;"STOP THE TAPE";CHR$(
25)
150 GOSUB LABEL TITLE
160 GOSUB LABEL RULES
170 LABEL RESTART
180 GOSUB LABEL MIX
190 LET y=1,z=0
200 REPEAT
210   GOSUB LABEL PLAY
220   IF s=0 THEN  GOTO LABEL SKIP
230   IF y<3 THEN  LET a=GETN
240   LET y=y+1
250   IF a<>78 THEN  LET y=4
260   PRINT @ 0,160;" ";
270   VDU 30,31,30,31,30,31
280 UNTIL y=4
290 GOSUB LABEL REPLAY
300 LABEL SKIP
310 VDU 1,4,2,0,4
320 PRINT @ 2,20;"Another game?"
330 LET a=GETN
331 LET r=0
332 FOR T=1 TO 6
333   FOR p=1 TO 9
334     LET C(r)=T,r=r+1
335   NEXT p
336 NEXT T
340 IF a<>78 THEN  GOTO LABEL RESTART
350 END
360 LABEL TITLE
370 GOSUB LABEL SETUP
380 CLS
390 FOR a=0 TO 4 STEP 2
400   LET b=9#a
410   FOR c=1 TO 9
420     PROC PAINT(A(b)+60,B(b),C(b),D(a
),E(a),F(a),G(a),H(a)
430     LET b=b+1
440   NEXT c
450 NEXT a
460 VDU 1,4,24
470 PRINT @ 40,80;"THE CUBE";
480 GOSUB LABEL SETUP2
490 RETURN
500 LABEL SETUP
510 DIM A(125),B(125),C(53),D(5),E(5),F(
5),G(5),H(5),I(107),J(15),K(162),S(47)
520 DIM A$(3)/179
530 LET e=0,K(162)=0
540 FOR b=0 TO 5
550   LET D(b)=21,E(b)=1,F(b)=15,G(b)=0,
H(b)=9
560   FOR c=1 TO 9
570     LET C(e)=b+1,e=e+1
580   NEXT c
590 NEXT b
600 LET F(2)=-10,G(2)=-10,F(3)=-10,G(3)=
610 FOR a=0 TO 53
620   READ A(a),B(a)
630 NEXT a
```

```
1000 NEXT a
1010 GOSUB LABEL SIDE
1020 RETURN
1030 LABEL SIDE
1040 IF K(162)=0 THEN  RETURN
1050 FOR J=0 TO 53
1060   VDU 1,0,2,C(J)
1070   PRINT @ A(J+72),B(J+72);C(J),
1080 NEXT J
1090 VDU 1,7,2,0
1100 RETURN
1110 LABEL RULES
1120 VDU 1,4,4
1130 PRINT @ 4,10;"THE CUBE";CHR$(25);
1140 PRINT @ 0,40;"Two views of the cub
e will be displayed",CHR$(31),"The left
shows the front faces and the right the
rear faces"
1150 PRINT @ 0,80;"Moves are entered by
,for example,+1X"
1160 PRINT @ 0,100;"+VE moves are clock
wise when viewed in the direction of the
axes"
1170 PRINT @ 0,140;"For a B&W display e
ach face is numbered  to indicate it's c
olour"
1180 PRINT @ 4,180;"Is your display B&W
?"
1190 LET a=GETN
1200 IF a<>78 THEN  LET K(162)=1
1210 RETURN
1220 LABEL ARRAY
1230 LET Z=3#P,Y=Z,X=P+36,W=X
1240 IF P>2 AND P<=5 THEN  LET Z=3#(P-3)
+36,Y=Z+54,X=P+15,W=X+54
1250 IF P>5 THEN  LET Z=3#(P-6)+18,Y=Z+1
08,X=P-6,W=X+108
1260 GOSUB LABEL MOVE
1270 IF P=1 OR P=4 OR P=7 THEN  RETURN
1280 LET V=18,U=V
1290 IF P=2 THEN  LET V=27,U=V
1300 IF P=3 THEN  LET V=9,U=V+54
1310 IF P=5 THEN  LET V=0,U=V+54
1320 IF P=6 THEN  LET V=36,U=V+108
1330 IF P=8 THEN  LET V=45,U=V+108
1340 GOSUB LABEL MOVE2
1350 RETURN
1360 LABEL MIX
1370 VDU 4,24
1380 PRINT @ 10,50;"Level of difficulty
 - 1,2 or 3"
1390 LET N=GETN
1400 LET N=(N-49)#4+8
1410 IF N=8 OR N=12 OR N=16 THEN  GOTO L
ABEL JUMP
1420 PRINT @ 10,220;"Wrong input !"
1430 PAUSE 5000
1440 GOTO LABEL MIX
1450 LABEL JUMP
1460 PRINT @ 10,50;"Please wait I am de
stroying the cube"
1470 FOR J=0 TO N-1
1480   LET J(J)=RAND(18)
1490 NEXT J
```

```
2110 FOR Q=1 TO x
2120   IF P<>18 THEN  GOSUB LABEL ARRA
Y
2130   ELSE GOSUB LABEL ERROR
2140   FOR J=0 TO 53
2150     SWAP C(J),I(J)
2160   NEXT J
2170 NEXT Q
2180 IF P<>18 THEN  GOSUB LABEL DRAW
2190 IF P<>18 THEN  LET z=z+1
2200 GOSUB LABEL CHECK
2210 IF s=0 THEN  LET z=y#N
2220 WEND
2230 IF s=0 THEN  RETURN
2240 PRINT @ 2,180;"You have had ";y;"x
the number of goes that I had."
2250 IF y<3 THEN  PRINT @ 0,210;"Do you
give up?";CHR$(7)
2260 IF y=3 THEN  PRINT @ 0,200;"I must
insist that you give up";CHR$(7);CHR$(7
)
2270 RETURN
2280 LABEL ERROR
2290 PRINT @ 20,240;"Invalid input";
2300 FOR J=54 TO 107
2310   LET I(J)=0
2320 NEXT J
2330 FOR J=0 TO 53
2340   SWAP C(J),I(J)
2350 NEXT J
2360 PRINT @ 20,240;"
2370 RETURN
2380 LABEL CHECK
2390 LET s=0
2400 FOR a=1 TO 6
2410   LET q=(a-1)#9
2420   FOR b=1 TO 8
2430     IF C(q)<>C(q+b) THEN  LET s=1
2440   NEXT b
2450 NEXT a
2460 IF s=1 THEN  RETURN
2470 VDU 1,4,2,0,4,24
2480 PRINT @ 36,50;"WELL DONE";CHR$(25)
2490 FOR a=0 TO 38
2500   BEEP 31-a,80,63
2510   PAUSE 300
2520 NEXT a
2530 CLS
2540 RETURN
2550 LABEL REPLAY
2560 CLS
2570 PRINT @ 2,20;"Do you want a replay
of the game?"
2580 LET t=GETN
2590 IF t=78 THEN  RETURN
2600 CLS
2610 LET r=0
2620 FOR T=1 TO 6
2630   FOR p=1 TO 9
2640     LET I(r+54)=0,C(r)=T,r=r+1
2650   NEXT p
2660 NEXT T
2670 GOSUB LABEL DRAW
2680 PRINT @ 2,120;"Here are my ";N;"
```

```
640 DATA 50,51,50,74,50,97,67,51,57,74,6
7,97,84,51,84,74,84,97,155,15,155,35,155
,61,172,15,172,38,172,61,189,15,189,38,1
89,61,23,26,35,38,47,50,23,49,35,61,47,7
3,23,72,35,84,47,96

650 DATA 217,26,229,35,241,50,217,49,229
,61,241,73,217,72,229,84,241,96,15,15,32
,15,49,15,27,27,44,27,61,27,39,39,56,39,
73,39,156,84,173,84,190,84,168,96,185,96
,202,96,179,108,196,108,213,108

660 FOR J=0 TO 17

670   READ A$(J)

680 NEXT J

690 DATA +1X,+2X,+3X,+1Y,+2Y,+3Y,+1Z,+2Z
,+3Z,-1X,-2X,-3X,-1Y,-2Y,-3Y,-1Z,-2Z,-3Z

700 RETURN

710 LABEL SETUP2

720 FOR J=0 TO 161

730   READ K(J)

740 NEXT J

750 DATA 51,48,45,52,49,46,53,50,47,42,3
9,36,42,40,37,44,41,38,20,23,26,19,22,25
,18,21,24,29,32,35,28,31,34,27,30,33,0,3
,6,1,4,7,2,5,8,9,12,15,10,13,16,11,14,17

760 DATA 6,3,0,7,4,1,8,5,2,15,12,9,16,13
,10,17,14,11,38,41,44,37,40,43,36,39,42,
47,50,53,46,49,52,45,48,51,27,30,33,28,3
1,34,29,32,35,18,21,24,19,22,25,20,23,26

770 DATA 29,32,35,28,31,34,27,30,33,20,2
3,26,19,22,25,18,21,24,0,3,6,1,4,7,2,5,8
,9,12,15,10,13,16,11,14,17,42,39,36,43,4
0,37,44,41,38,51,48,45,52,49,46,53,50,47

780 FOR J=54 TO 125

790   READ A(J),B(J)

800 NEXT J

810 DATA 78,5,86,5,94,5,94,5,71,85,76,98
,82,111,70,22,73,45,70,68,5,5,17,5,24,5,
2,86,7,97,12,109,0,22,0,45,0,68

820 DATA 27,57,27,80,27,102,36,57,36,80,
36,102,45,57,45,80,45,102,80,21,80,44,80
,67,88,21,88,44,88,67,97,21,97,44,97,67,
5,28,14,39,20,51,8,50,14,62,20,74,8,74,1
4,14,86,20,97

830 DATA 105,27,111,48,117,53,105,50,111
,63,117,74,105,74,111,84,117,95,13,15,21
,15,30,15,18,27,26,27,35,27,24,39,33,39,
41,39,83,94,91,84,110,84,39,96,97,96,1)6
,96,95,108,103,108,111,108

840 RETURN

850 DEFPROC PAINT(A,B,C,D,E,F,G,H)

860 INK C

870 FOR d=1 TO D

880   MOVE A,B

890   PLOT 3,F,G

900   LET B=B+E,A=A+H

910 NEXT d

920 ENDPROC

930 LABEL DRAW

940 LET b=0

950 FOR a=0 TO 5

960   FOR c=1 TO 9

970     IF C(b)<>I(b+54) THEN PROC PAIN
T(A(b),B(b),C(b),D(a),E(a),F(a),G(a),H(a
))

980     LET b=b+1

990   NEXT c
```

```
1500 FOR J=54 TO 107

1510   LET I(J)=0

1520 NEXT J

1530 FOR M=0 TO N-1

1540   IF J(M)<=8 THEN LET P=J(M),x=1

1550   ELSE LET P=J(M)-9,x=2

1560   FOR Q=1 TO x

1570     GOSUB LABEL ARRAY

1580     FOR T=0 TO 53

1590       LET C(T)=I(T),I(T)=0

1600     NEXT T

1610   NEXT Q

1620 NEXT M

1630 VDU 4,25

1640 GOSUB LABEL DRAW

1650 GOSUB LABEL AXES

1660 GOSUB LABEL NUMBER

1670 RETURN

1680 LABEL MOVE

1690 FOR J=0 TO 53

1700   LET I(J)=C(J)

1710 NEXT J

1720 FOR a=1 TO 3

1730   LET I(K(Y))=C(Z),Z=Z+1,Y=Y+1

1740 NEXT a

1750 LET Z=Z+6,Y=Y+6

1760 FOR a=1 TO 3

1770   LET I(K(Y))=C(Z),Z=Z+1,Y=Y+1

1780 NEXT a

1790 FOR a=1 TO 6

1800   LET I(K(W))=C(X),W=W+3,X=X+3

1810 NEXT a

1820 RETURN

1830 LABEL MOVE2

1840 FOR a=1 TO 9

1850   LET I(K(U))=C(V),U=U+1,V=V+1

1860 NEXT a

1870 RETURN

1880 LABEL NUMBER

1890 LET Q=0

1900 FOR J=54 TO 71

1910   PRINT @ A(J),B(J);RIGHT$(A$(Q),2
);

1920   LET Q=Q+1

1930 NEXT J

1940 RETURN

1950 LABEL PLAY

1960 PRINT @ 16,140;"FRONT FACES",,"
REAR FACES"

1970 PRINT @ 40,160;"HERE IS YOUR CUBE"

1980 WHILE z<yIN

1990   FOR J=0 TO 53

2000     LET I(J+24)=C(J)

2010   NEXT J

2020   PRINT @ 20,180;"You have had ";z
;" moves so far"

2030   PRINT @ 20,200;"Enter your next
move"

2040   INPUT Z$

2050   LET P=18

2060   FOR J=0 TO 17

2070     IF Z$=A$(J) THEN LET P=J

2080   NEXT J

2090   LET S(z)=P,x=1

2100   IF P>8 AND P<18 THEN LET P=P-9,x
=3
```

```
. . oves"

2690 LET u=0,v=8,x=0,r=N,g=0

2700 REPEAT

2710   LET r=r-x,p=1,x=0

2720   IF r<8 THEN LET v=r

2730   FOR e=u TO u+v-1

2740     FOR J=0 TO 53

2750       I(J+54)=C(J)

2760     NEXT J

2770     PRINT @ (p-1)*10+1,135+g;A$(J(
e));

2780     LET P=J(e),q=1

2790     IF P>8 THEN LET P=P-9,q=3

2800     FOR Q=1 TO q

2810       GOSUB LABEL ARRAY

2820       FOR T=0 TO 53

2830         SWAP C(T),I(T)

2840       NEXT T

2850     NEXT Q

2860     GOSUB LABEL DRAW

2870     LET p=p+1,x=x+1

2880     PAUSE 5000

2890   NEXT e

2900   LET u=u+x,g=10

2910 UNTIL e=N

2920 PRINT @ 2,160;"Here are your ";z;"
moves"

2930 LET u=0,v=8,x=0,r=z

2940 REPEAT

2950   LET r=r-x,p=1,x=0

2960   IF r<8 THEN LET v=r

2970   FOR e=u TO u+v-1

2980     PRINT TAB (p-1)*4+1;A$(S(e));

2990     LET p=p+1,x=x+1

3000   NEXT e

3010   VDU 31

3020   LET u=u+x

3030 UNTIL e=z

3040 PRINT @ 2,0;"Press any key to cont
inue";

3050 LET e=GETN

3060 RETURN

3070 LABEL AXES

3080 INK 4

3090 MOVE 130,120

3100 PLOT 3,-12,-12

3110 MOVE 130,120

3120 PLOT 3,15,0

3130 MOVE 130,120

3140 PLOT 3,0,-25

3150 DOT 118,109

3160 DOT 119,108

3170 DOT 118,110

3180 DOT 120,108

3190 PRINT @ 55,102;"Y";

3200 DOT 131,96

3210 DOT 132,97

3220 DOT 129,96

3230 DOT 128,97

3240 PRINT @ 64,84;"Z";

3250 DOT 143,118

3260 DOT 144,119

3270 DOT 143,122

3280 DOT 144,121

3290 PRINT @ 73,115;"X";

3300 RETURN
```

```
DOMINOES
********

10 REM DOMINOES/b216
15 REM (C) Colin I Clayman 1984
20 CODE 00 00 00 00 00 00 00 00 00 00 00 0
[ 00 00 00 00 00 00 00 00 00 00 30 30 00
00 00 00 00 03 00 00 30 30 00 00 00 00 00
00 00 00 00 33 33 00 33 33 00 00 2C 2D 0
0 00 00 00 20 20 00
30 LET I=1,W=3,H=10,S=b,M=5+I,G=H+I,K=G+
G,N=27,Q=230,r=-1,H$=CHR$(23)+CHR$(M)
40 DIM D(N),d(I),h(I),H(N+N),X(W),Y(W),O
(W),E(W),G(S),V(S),S(I),T(I),Q(I)
50 LET T(I)=FALSE,T(FALSE)=FALSE
60 DPOKE GRAPHIC,LCTN(H+H)
70 RANDOM
80 REPEAT
90   PROC table
100  PROC shuffle
110  PROC start
120  REPEAT
130    LET p=NOTg,f=d
140    IF g AND r THEN PROC you
150    ELSE PROC me
160    UNTIL NOTd(g) OR Q(FALSE)=d AND Q(
I)=d
170    IF r THEN LET T(I)=T(I)+S(I),T(FA
LSE)=T(FALSE)+S(FALSE)
180    PAPER RED
190    INK YELLOW
200    IF NOTd(I) THEN PRINT H$"YOU WON!
".
210    ELSE IF NOTd(FALSE) THEN PRINT H
$"I WON!".
220    ELSE PRINT H$"STALEMATE!";
230    PRINT " Scores. "S(g);" v "S(NOTg)
." in Game "r
240    INK WHITE
250    PRINT ," TOTALS- You: "T(I)," v Me
: "T(FALSE);
260    LET X$=GET$,k$=UPC$(X$)
270 UNTIL k$="Q"
280 WINDOW W,123,5-I,245
290 DPOKE GRAPHIC,LETTER(32
300 PAPER BLACK
310 END
320 DEFPROC start
330 LET s=-I,g=RAND(I+I)
340 FOR x=FALSE TO I
350   FOR j=FALSE TO h(x)
360     PROC value(H(x%N+j))
370     IF p=q AND p>s THEN LET s=p,i=j
,g=x
380   NEXT j
390 NEXT x
400 WHILE s=-I
410   LET g=NOTg
420   PROC take
430   IF p=q THEN LET s=p
440 WEND
450 PROC place(i,I)
460 LET E(FALSE)=p,E(I+I)=p,E(W)=p
470 ENDPROC
480 DEFPROC you
490 PAPER RED
500 REPEAT

860 LET i=v+y
870 WHILE i>-I AND i<=a AND i<>v
880   IF NOTi THEN LET v=i
890   ELSE IF H(N+i-I)>-I THEN LET v=i
900   ELSE LET i=i+y
910 WEND
920 ENDPROC
930 DEFPROC me
940 FOR i=FALSE TO S
950   LET G(i)=FALSE
960 NEXT i
970 LET a=FALSE
980 FOR j=FALSE TO h(g)
990   PROC value(H(N%g+j))
1000  PROC legal
1010  LET a=a+v
1020  IF v THEN LET i=j,b=s
1030 NEXT j
1040 IF a THEN LET v=a,s=b
1050 IF v THEN GOTO 1090
1060 PROC take
1070 PROC legal
1080 IF NOTv THEN ENDPROC
1090 IF v=I THEN GOTO 1260
1100 LET s=-I
1110 FOR j=FALSE TO s
1120   IF NOTG(j) THEN GOTO 1260
1130   LET a=FALSE
1140   FOR i=FALSE TO h(g)
1150     PROC value(H(N%g+i))
1160     LET a=a+(j=p OR j=q)
1170   NEXT i
1180   IF a>s THEN LET s=a,b=j
1190 NEXT j
1200 LET i=-I
1210 REPEAT
1220   LET i=i+I,G(b)=FALSE
1230   PROC value(H(N%g+i))
1240   PROC legal
1250 UNTIL G(b)
1260 PROC place(i,s)
1270 ENDPROC
1280 DEFPROC value(n)
1290 IF n>-I THEN LET p=INT(7.5-SQR(225
-M%n-n)/2,q=n+p%p/2-6.5%p
1300 ELSE LET p=-I,q=-I
1310 ENDPROC
1320 DEFPROC domino(p,q)
1330 IF p>q THEN SWAP p,q
1340 LET n=b.5%p-p%p/2+q
1350 ENDPROC
1360 DEFPROC legal
1370 LET v=FALSE,s=v
1380 IF p=-I THEN ENDPROC
1390 FOR e=FALSE TO z
1400   PROC fit(p,q)
1410   PROC fit(q,p)
1420 NEXT e
1430 ENDPROC
1440 DEFPROC fit(x,y)
1450 IF NOTv OR E(e)<>E(s) THEN IF E(e)
=x AND (z=I OR V(x) AND (o(S-I OR o)=H-I
OR f<=I) OR x=y AND o)=S-I) THEN LET G
(y)=I,s=e,v=v+I

1770 IF ABS(O(s))=H THEN LET v=p-q,a=X(
s)+(S-I)%(O(s)(I)%v,b=Y(s)-S%v,X(s)=X(s)
+O(s)-(S-I)%SGN(O(s))%v
1780 IF ABS(O(s))=K THEN LET v=p<>q,a=X
(s)-(I+I)%NOTv,b=Y(s)+G%(O(s)(I)%NOTv,Y(
s)=Y(s)+O(s)-G%SGN(O(s))%NOTv
1790 IF E(s)=q AND O(s)>I OR E(s)=p AND
O(s)<I THEN SWAP p,q
1800 IF E(s)=p THEN LET E(s)=q
1810 ELSE LET E(s)=p
1820 IF p=q THEN LET V(p)=I
1830 PROC piece(a,b,v)
1840 PROC blank(5+M+i%G,v+2I5%g)
1850 ENDPROC
1860 DEFPROC equal
1870 LET v=FALSE
1880 FOR e=FALSE TO z
1890   LET v=v+(E(e)=E(s))
1900 NEXT e
1910 IF v=I THEN ENDPROC
1920 LET v=RAND(v)
1930 REPEAT
1940   LET e=e-I,v=v-(E(e)=E(s))
1950 UNTIL v=-1
1960 LET s=e
1970 ENDPROC
1980 DEFPROC take
1990 LET Q(g)=d,p=-I,q=p
2000 IF d=-I OR d=I AND z>I OR d(g)=H TH
EN ENDPROC
2010 IF d<M+M THEN PROC blank(I,N+d%I)
2020 LET i=FALSE,a=I
2030 WHILE i<d(g) AND a
2040   LET a=H(g%N+i)>-I,i=i+a
2050 WEND
2060 IF i>h(g) THEN LET h(g)=i
2070 PROC value(O(d%g-NOTg)
2080 PROC piece(5+M+i%G,W+2I5%g,FALSE)
2090 PROC value(D(d))
2100 LET H(g%N+i)=n,d(g)=d(g)+I,d=d-I,S(
g)=S(g)+p+q
2110 ENDPROC
2120 DEFPROC shuffle
2130 IF r=-1 THEN PROC intro
2140 ELSE PRINT H$
2150 PRINT
2160 LET r=r+I,d=N,o=FALSE,X(I)=b2,X(FAL
SE)=X(I)-H,E(I)=I
2170 FOR i=FALSE TO N
2180   LET D(i)=i
2190 NEXT i
2200 FOR i=I TO H%H
2210   SWAP D(RAND(N+I)),D(RAND(N+I))
2220 NEXT i
2230 FOR g=FALSE TO I
2240   LET S(g)=FALSE,d(g)=FALSE,h(g)=FA
LSE,Y(g)=G%H,X(W-g)=X(I),O(g)=H%(g+g-I)
O(W-g)=K%(g+g-I),Y(W-g)=Y(g)-S+O(W-g)
2250   FOR i=FALSE TO S
2260     PROC take
2270     LET V(i)=FALSE
2280   NEXT i
2290 NEXT g
2300 FOR i=FALSE TO d
```

```
510  PRINT H$CHR$(H),"Point at your dom     1460 ENDPROC                              2310   PROC value(-I)
ino! ";                                    1470 DEFPROC piece(x,y,v)                  2320   PROC piece(I,N+i*13,FALSE)
520   PROC point(h(I)+I,G)                  1480 REM Horiz. 19 x 10 or Vert. 9 x 31    2330 NEXT i
530   LET j=v,v=j                           pixels                                    2340 ENDPROC
540   IF j THEN  PROC value(H(N+j-I))       1490 PAPER BLACK                           2350 DEFPROC table
550   IF j THEN  PROC legal                 1500 PROC half(x,y,p)                      2360 PAPER GREEN
560 UNTIL NOT j OR w                        1510 PROC half(x+(S-I)*NOTv,y+G*v,q)       2370 CLS
570 IF NOT j THEN  PROC take                1520 INK YELLOW                            2380 INK BLUE
580 IF NOT j THEN  PROC legal               1530 MOVE x+x+(H-I)*NOTv,y+H*v             2390 PRINT @ FALSE,217;"PASS";
590 IF NOT j AND w THEN  LET j=i+I          1540 IF k>-I THEN  PLOT W,M*v+v,(H-I)*NO   2400 INK RED
600 PAPER RED                               Tv                                        2410 MOVE FALSE,15
610 WHILE j AND w<>I                        1550 INK WHITE                             2420 DRAW 245,15
620   PRINT H$CHR$(H), "Point at its joi    1560 ENDPROC                               2430 DRAW 245,215
ning side!";                               1570 DEFPROC half(a,b,k)                   2440 DRAW FALSE,215
630   PROC point(I,H)                       1580 INK WHITE                             2450 MOVE 23,-I
640   LET w=FALSE,s=v                       1590 PRINT @ a,b;" "; @ a+I,b;            2460 DRAW 23,FALSE
650   FOR e=FALSE TO z                      1600 IF k>-I THEN  PRINT CHR$(128+k+(k=S   2470 PAPER RED
660    IF v THEN  PROC fit(q,p)             AND NOTv));                                2480 INK WHITE
670    ELSE  PROC fit(p,q)                  1610 ELSE  PRINT " ";                      2490 ENDPROC
680   NEXT e                                1620 INK BLACK                             2500 DEFPROC intro
690 WEND                                    1630 MOVE a+a+M+I,b                        2510 WINDOW W,123,Q,Q+H+H
700 IF j THEN  PROC place(j-I,s             1640 PLOT W,FALSE,H-I                      2520 PRINT H$"Joystick or Keyboard?"
710 ENDPROC                                 1650 ENDPROC                               2530 REPEAT
720 DEFPROC point(a,b                       1660 DEFPROC blank(x,y)                    2540   LET J$=GET$,J$=UPC$(J$)
730 LET v=I,x=(S-I)*(b=G)+j*G*(b<>G)        1670 PAPER GREEN                           2550 UNTIL J$="J" OR J$="K"
740 REPEAT                                  1680 PRINT @ x,y;" " @ x+M,y;" ";        2560 PRINT "Star Dominoes (Y/N)?"
750   PRINT @ x+v*b,Q;CHR$(125);           1690 PAPER BLACK                           2570 PROC ask
760   REPEAT                                1700 ENDPROC                               2580 LET z=I+(I+I)*(D$="Y")
770    IF J$="J" THEN  LET k=INP(&007A)      1710 DEFPROC place(i,s)                    2590 PRINT "Would you like a demonstrati
,y=(NOT(k BAND M+I) AND v<a)-(NOT(k BNA     1720 IF o THEN  PROC equal                 on (Y/N)?"
ND W+I) AND v),k=k BNXOR &00FF              1730 PROC value(H(N*g+i))                  2600 PROC ask
780    ELSE  LET k=GETN,y=(k=G+I AND v<      1740 LET o=o+I,H(g*N+i)=-I,d(g)=d(g)-I,S   2610 IF D$="N" THEN  LET r=r+I
a)-(k=K AND v),k=I+(k=ASC(" "))*&001F       (g)=S(g)-p-q,Q(g)=N                       2620 ENDPROC
790    UNTIL k                              1750 IF ABS(O(s))=H AND (X(s)<15 OR X(s)   2630 DEFPROC ask
800    PRINT @ x+v*b,Q;" ";                 )>H*H) THEN  LET X(s)=X(s)-O(s)*(O(s)<I),  2640 REPEAT
810    IF b=G AND y THEN  PROC move          O(s)=SGN(O(s))*K,Y(s)=Y(s)+O(s)*(O(s)<I)   2650   LET D$=GET$,D$=UPC$(D$)
820    ELSE  LET v=v+y                       ,O(s)=SGN(O(s))*H,X(s)=X(s)-(S-I)*(O(     2660 UNTIL D$="Y" OR D$="N"
830 UNTIL k BAND &0020                       s)<I)                                     2670 ENDPROC
840 ENDPROC                                 1760 ELSE  IF ABS(O(s))=K AND (Y(s)<40 O
850 DEFPROC move                            R Y(s)>180) THEN  LET Y(s)=Y(s)+G*(O(s)<
                                            I),O(s)=-SGN(O(s))*H,X(s)=X(s)-(S-I)*(O(
                                            s)<I)
```

ANALOGUE CLOCK

```
100 REM -=*=-  E.J.EDWARDS.  -=*=-         250 DRAW 128,124                          460   NEXT I
      -=*=-   JAN. 1995   -=*=-            260 PROTECT MAGENTA                        470 NEXT J
110 PROTECT BLACK                          270 REM -=*=- DRAW CLOCK FACE -=*=-        480 INK BLACK
120 VDU 1,7,2,0,4                          280 INK GREEN                              490 PROTECT RED
130 PRINT @ 3,225;"";                      290 FOR C=0 TO 359 STEP 30                 500 MOVE 128+35*SIN(N),124+35*COS(N)
140 INPUT "ENTER TIME (HH.MM) ";H          300   LET A=RAD(C)                         510 DRAW 128,124
150 PRINT @ 61,60;"12"; @ 77,68;"1";       310    MOVE 128+50*SIN(A),124+50*COS(A)    520 INK BLUE
@ 85,90;"2"; @ 91,120;"3"; @ 88,150;"4     320    DRAW 128+45*SIN(A),124+45*COS(A)    530 LET E=E-b,N=RAD(E)
"; @ 77,17;"5"; @ 62,180;"6"; @ 46,17      330 NEXT C                                 540 MOVE 128+35*SIN(N),124+35*COS(N)
0;"7"; @ 37,150;"8"; @ 34,120;"9";         340 LET Q=GETN                             550 DRAW 128,124
160 PRINT @ 34,90;"10"; @ 45,68;"11";      350 PROTECT BLACK                          560 PROTECT MAGENTA
170 PRINT @ 3,225;"Press any *KEY* to S    360 PRINT @ 3,225;"                        570 INK BLACK
TART Clock."                                                      ";                  580 PROTECT CYAN
180 LET M=FRAC(H)*100,LET M=360-M*6,LET    370 PROTECT MAGENTA                        590 MOVE 128+30*SIN(U),124+30*COS(U)
E=M-180,LET H=360-H*30,LET R=H-180,LET N   380 REM -=*=- MAIN PROGRAM LOOP -=*=-      600 DRAW 128,124
=RAD(E),LET U=RAD(R)                       390 FOR J=174 TO -180 STEP -6             610 INK RED
190 INK RED                                400   LET A=RAD(J)                         620 LET R=R-0.5,U=RAD(R)
200 MOVE 128+30*SIN(U),124+30*COS(U)       410    FOR I=4 TO 0 STEP -4               630 MOVE 128+30*SIN(U),124+30*COS(U)
210 DRAW 128,124                           420     INK I                             640 DRAW 128,124
220 PROTECT RED                            430     MOVE 128+40*SIN(A),124+40*COS(A)   650 PROTECT MAGENTA
230 INK BLUE                               440     DRAW 128,124                       660 IF E<=-180 THEN  LET E=174
240 MOVE 128+35*SIN(N),124+35*COS(N)       450     PAUSE 1870*I+1                     670 GOTO 390
```

CODE LINE GENERATOR
....

This program will generate CODE lines directly from the Monitor. (Ed. This article was originally submitted to NILUG magazine for publication).

The program assumes that there is a machine code routine starting at &9000. When CALLed, the program (residing at &9200, although it can easily be relocated), assembles a CODE line, (line number no. 10000, this also can be changed) containing the machine code routine located at &9000. It also assumes that the m/c routine finishes with a C9 (RETURN). The CODE line must not be longer than 240 bytes.

The program can be divided into several sections:-

1) SET UP LINE NUMBER AND CODE TOKEN.
The program reads in the End Of Basic pointer into DE, sets HL=Top of data table (located at the end of program), sets BC=07 (number of bytes) and uses the LDIR instruction to transfer the data. The line number takes 5 bytes (C4 10 00 00 00), the next byte is the line length, which is set to zero initially and filled in later and the CODE token (33) is the 7th byte. The machine code follows directly on from this.
2) STORE MACHINE CODE ROUTINE.
Firstly the DE register is saved (at this point DE=Address of line length byte + 2) and the B register is zeroed. HL is set to the beginning of the machine code routine to be copied and DE points to the first location for the data to be copied to. Then follows a routine to transfer each byte, check for a C9 and continue until a C9 is found.
3) STORE END OF LINE TOKEN & END OF BASIC TOKEN.
The DE register is incremented past the last byte of machine code (which is C9), and the end of line token &0D is inserted. The End Of Basic token &80 is stored in the next location. The value held in DE is then transferred to HL for use later.
4) STORE LINE LENGTH
DE is "POPped" and is decremented twice to bring it back to the address of the line length byte.
B is recalled and has &09 added to it to bring it up to the line length. It is then stored in the correct byte.
5) RESTORE END OF BASIC POINTER
H=High byte of end of BASIC.
L=Low byte of end of BASIC.
These two are stored into the locations &61FD and &61FC and then a return is executed.

The program as it stands has several limitations:-

1) Machine code programs to be copied must have a "C9" at their end, but no where else (i.e. this program will not work on itself).
2) Maximum length of 240 bytes.
3) The program can be relocated provided that the third instruction has its argument changed. i.e.

LD HL,&9235. The argument =Start address + &35. Thus the program is not suitable for use in a CODE line since these lines will move around as a BASIC program is edited.
4) Before calling the routine it is important that the last BASIC line has a value <10000 but the value of 10000 can be changed by altering the data table at the end of the routine.

Finally, the program was written in m/c for several reasons:-

1) Speed
2) It was a challenge compared to BASIC.
3) I was not sure whether standard variables were stored at the end of a BASIC program, in which case they would be overwritten by the CODE line.

NOTES	CODE	MNEMONICS	COMMENTS
Set up	2A FC 61	LD HL,(&61FC)	;End Of Basic=Start
line	EB	EX DE,HL	;of created line
No. &	21 35 92	LD HL,&9235	;Start of data table
CODE	01 07 00	LD BC,07	;No. of bytes
token	ED B0	LDIR	;Load in data
	D5	PUSH DE	;Store address
	A8	XOR B	;Set counter
STORE	21 00 90	LD HL,&9000	;Beginning of m/c
M/C	7E	LD A,(HL)	;loop begins
	12	LD (DE),A	;Store m/c
	• FE C9	CP &C9	;Check for return
	28 05	JRZ,&05	;If RET then end
	13	INC DE	;Next byte
	23	INC HL	;Next byte
	04	INC B	;Increment counter
	18 F5	JR,&F5	;Back to loop start
St.End	13	INC DE	;Advance to end
of line	3E 0D	LD A,&0D	;of line byte
token	12	LD (DE),A	;Insert token
St.End	13	INC DE	;Advance to end of
Of	3E 80	LD A,&80	;program byte
Basic	12	LD (DE)	;Insert token
token	EB	EX DE,HL	;Save end of BASIC
St.Line	D1	POP DE	;Line length byte+2
length	1B	DEC DE	;Get to line length
	1B	DEC DE	;byte address
	78	LD A,B	;Recall bytes-1
	C6 09	ADD A,&09	;Make up line length
	12	LD (DE)	;Store it
Restore	7C	LD A,H	;Address high nibble
End Of	32 FD 61	LD &61FD,A	;Store it
Basic	7D	LD A,L	;Address low nibble
pointer	32 FC 61	LD &61FC,A	;Store it
	C9	RET	
Data	C4		;First 5 bytes
Table	10		;=line no.(10000)
	00		;6th byte=line lngth
	00		;fill in later
	00		;7th byte=Code token
	00		
	33		
		END	

P.Cottingridge.

TOWER OF HANOI
```` `` `` `` `` `` `` ``

```
50 REM TOWER of HANOI by E J EDWARDS
60 PROTECT BLACK
70 VDU 1,7,2,2,4
80 PRINT @ 6,35;"INSTRUCTIONS (Y/N) ";
90 LET A=GETN
100 IF A=ASC("N") OR A=ASC("n") THEN  GO
TO 240
110 IF A=ASC("Y") OR A=ASC("y") THEN  GO
TO 130
120 GOTO 80
130 PRINT @ 6,25;"The object of this ga
me is to move all"
140 PRINT "the Discs from one pile to an
other, mov-"
150 PRINT "ing one at a time. The only r
estriction"
160 PRINT "is that a larger Disc cannot
be placed"
170 PRINT "on a smaller. To make a move,
 type in"
180 PRINT "the pile from which the Disc
is to be"
190 PRINT "removed and the pile on which
it is to"
200 PRINT "be placed"
210 PRINT
220 PRINT "Hit ▓SPACEBAR▓ to Begin.";
230 LET A=GETN
240 DIM A(3▓8)
250 REM ▓ INITIALIZE SCREEN AND ARRAY ▓
260 VDU 1,3,2,0,4
270 PRINT @ 23,25;"T O W E R  O F  H A N
O I"
280 PRINT @ 23,30;CHR$(21);"============
=============";CHR$(20)
290 FOR X=28 TO 208
300   DOT X,160
310   DOT X,161
320 NEXT X
330 FOR Y=159 TO 100 STEP -1
340   FOR X=58 TO 178 STEP 60
350     DOT X,Y
360   NEXT X
370 NEXT Y
```

```
380 FOR Z=159 TO 100 STEP -9
390   IF Z=159 OR Z=141 OR Z=123 OR Z=10
5 THEN INK CYAN
400   ELSE INK MAGENTA
410   FOR Y=Z-2 TO Z-6 STEP -1
420     LET W=(Z-90)/3-2
430     FOR X=118-W TO 118+W
440       DOT X,Y
450       BEEP 1+X,2,63
460     NEXT X
470   NEXT Y
480 NEXT Z
490 FOR X=0 TO 2
500   PRINT @ 27+X▓30,175;X+1;
510 NEXT X
520 FOR Y=2 TO 8
530   LET A(2▓8-Y)=9-Y
540 NEXT Y
550 LET A(1▓8-1)=1,A(2▓8-1)=8,A(3▓8-1)=1
,N=0
560 PRINT @ 6,225;"ENTER NEXT MOVE : ";
570 REM : -=▓=-  ACCEPT MOVE  -=▓=-
580 GOSUB 1000
590 LET F=A
600 GOSUB 1000
610 LET T=A
620 IF A(F▓8-1)>1 AND A(F▓8-A(F▓8-1))<A(
T▓8-A(T▓8-1)) THEN GOTO 710
630 IF A(T▓8-1)=1 THEN GOTO 710
640 PRINT @ 6,225;" - INVALID, RE-ENTER
";
650 BEEP 90,200,63
660 BEEP 200,100,63
670 BEEP 90,200,63
680 PAUSE 1000
690 PRINT @ 6,225;"
";
700 GOTO 560
710 PRINT @ 6,225;"
"; @ 6,225;"";
720 REM : -=▓=-  MOVE DISK  -=▓=-
730 LET Z=168-(A(F▓8-1)-1)▓9,V=58+60▓(F-
1),W=3▓A(F▓8-A(F▓8-1))
740 FOR Y=Z-2 TO Z-6 STEP -1
```

```
750   FOR X=V-W TO V+W
760     INK PAPER
770     DOT X,Y
780     BEEP 200-X,1,63
790   NEXT X
800   INK GREEN
810   DOT V,Y
820 NEXT Y
830 LET A(T▓8-1)=A(T▓8-1)+1,A(T▓8-A(T▓8-
1))=A(F▓8-A(F▓8-1))
840 LET A(F▓8-A(F▓8-1))=1,A(F▓8-1)=A(F▓8
-1)-1,N=N+1
850 LET Z=168-(A(T▓8-1)-1)▓9,V=58+60▓(T
-1),W=3▓A(T▓8-A(T▓8-1))
860 IF Z=159 OR Z=141 OR Z=123 OR Z=105
THEN  INK CYAN
870 ELSE INK MAGENTA
880 FOR Y=Z-2 TO Z-6 STEP -1
890   FOR X=V-W TO V+W
900     DOT X,Y
910     BEEP 1+X,2,63
920   NEXT X
930 NEXT Y
940 INK GREEN
950 PRINT @ 23,195;"MOVES SO FAR :";N
";
960 IF A(1▓8-1)<>8 AND A(3▓8-1)<>8 THEN
GOTO 560
970 REM : -=▓=-  END OF GAME  -=▓=-
980 PRINT @ 6,225;"  YOU TOOK ";N;" MO
ES"
990 END
1000 REM ▓ ROUTINE ACCEPT MOVE ▓
1010 LET A$=KEY$
1020 IF A$="" THEN GOTO 1000
1030 ELSE LET A=VAL(A$)
1040 IF A<1 OR A>3 THEN GOTO 1010
1050 FOR J=50 TO 70 STEP 2
1060   BEEP J,8,63
1070 NEXT J
1080 PRINT A;" ";
1090 RETURN
```

E.J.EDWARDS

IMPORTANT NOTICE

As you may be aware, the government introduced new legislation last year (1985) regarding personal information held on computer filing systems. This was called the DATA PROTECTION ACT. It is still unclear at present as to whether user groups like ours have to register, I will therefore be contacting the committee of the ACC, to which we are affiliated, for clarification of this situation. For your information, the details you have supplied on the subscription form when you joined the group, are held on a card index system and also on disk. This is done for ease of operation, and also to produce the labels on the envelope! There are in fact only two copies of the disk in circulation, and this will REMAIN so. I have one and our membership secretary (Chris Mathews) has the other. Should the LYNX USER GROUP "fade away", then this information will be destroyed.

However in case we have to register and to update our records, I have to ask further questions:-

1) Do you have any objection to the fact that your details are held on disk, please answer YES or NO:- ....
Please make any comments on the back of this form.

2) If you are no longer interested in the LYNX for any reason, do you want your details erased from the files, YES or NO:- ....

3) If you have changed computers, do you still want to receive the outstanding magazines:- ....

4) Please insert here which LYNX and/or peripherals you now possess:-
....

If you don't return this form, I will assume that your answers to 1), 2), and 3) will be "NO", "NO", and "YES" respectively. Please respond as soon as possible. EDITOR, R.B.JONES.