

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## CONTENTS

---

BRICKOUT.....	Page 1
TENNIS.....	Page 4
SQUASH.....	Page 7
MASTERMIND.....	Page 10
SNAKE BITE.....	Page 13
DISASSEMBLER.....	Page 18
MUNCHER CRUNCH.....	Page 23
LYNX LOGO.....	Page 27
OTHELLO.....	Page 32
HANGMAN.....	Page 38

All programs will run on both the 48k and 96k Lynx  
unless otherwise stated.

### \*NOTE\*

Be careful that the following characters are distinguished :-

- 1 is the number one
- l is the little el
- I is the large letter ey

All material in this booklet remains the copyright of the Reading Lynx User Group and/or the authors. It must not be reproduced in whole or in part without written consent from the copyright holder.

If you live or work in the Reading area we would like to see you in the group. For further information contact Alan Bristow on 0734 - 412176.

(c) Reading Lynx User Group 1985

## B R I C K O U T

~~~~~

(c) Colin I Clayman 1985

## DESCRIPTION

This program plays the game of Brickout ( or Breakout ). Your aim is to knock out the most coloured bricks with 10 balls without the ball going out of play past your bat. The ball bounces on the walls of the court, and the bricks. The highest score to date is shown.

Use the left and right arrow keys to control the bat. At the end of the game, you can press the 'Q' key to quit or any other key for another game.

-----

```

10 REM **** BRICKOUT BY C.I.CLAYMAN,1985 *****
20 CODE 00 00 00 00 00 00 3F 3F 3F 3F 00 00 0E 1F 1F 1F 0E 00 00
     00 3F 20 20 20 20 20 20 20 20 3F 3F 01 01 01 01 01 01 01 01
     3F
30 LET I=1,W=3,H=10,L=63,N=5,M=H+H,T=N*H+I,X=123,Y=221,h=FALSE
40 DIM B(M*N)
50 RANDOM
60 CLS
70 DPOKE GRAPHIC,LCTN(M)
80 REPEAT
90   LET x=67,s=FALSE,n=H
100 REPEAT
110   LET a = W, b = Y, n = n + I, l = H * H
120   PROC bricks
130   REPEAT
140   PROC bat
150   PROC ball
160   UNTIL NOT l OR NOT n
170   UNTIL NOT n
180   IF s > h THEN LET h = s
190   PRINT @ 84, FALSE;"HiScore:"h; @ x,Y;" "
200   REPEAT
210   LET K$ = GET$,K$ = UPC$(K$)
220   UNTIL NOT " " > K$
230 UNTIL K$ ="Q"
240 DPOKE GRAPHIC,LETTER(32)
250 INK WHITE
260 END
270 DEFPROC bat
280 LET k = INP(&0980)
290 IF k < > &0OFF THEN PRINT @ x,Y;" "
300 LET x = x + W * W * ((NOT(k BNAND &0020) AND x < 112) -
     (NOT(k BNAND &0004) AND x > N))
310 INK YELLOW
320 PRINT @ x,Y;CHR$(128)CHR$(128)CHR$(128);
330 ENDPROC
340 DEFPROC ball
350 PRINT @ a,b;" ";
360 IF b = Y AND ABS(a - x - W) > N THEN PROC new
370 IF b = Y OR b + v < H THEN LET v = - v,u = SGN(u) * W *
     (RAND(I + I) + I)
380 IF b = Y THEN BEEP H * H,H,L

```

```

390 IF a + u < W OR a + u > X - W THEN LET u = - u
400 LET a = a + u,b = b + v
410 IF b < T + H * N AND b >= T THEN PROC check
420 INK CYAN
430 IF n THEN PRINT @ a,b;CHR$(129);
440 ENDPROC
450 DEFPROC check
460 LET j = (b - T) DIV H,i = (a - W) DIV (W + W),p = j * M + i
470 IF NOTB(p) THEN ENDPROC
480 LET B(p) = FALSE, s = s + 1, l = l + 1,v = -v
490 PRINT @ W + i * (W + W), T + j * H;" "I@ 24, FALSE;s;
500 BEEP 300,20,L
510 ENDPROC
520 DEFPROC new
530 BEEP H * H,1000,L
540 LET u = RAND(I + I) - 0.5, v = H,a = x + N,b = Y,n = n - I
550 PRINT @ 42, FALSE;"Balls Left:";n;" ";
560 ENDPROC
570 DEFPROC bricks
580 INK GREEN
590 MOVE N,Y + H
600 DRAW N,H
610 DRAW X + X,H
620 PLOT W, FALSE, Y
630 PRINT @ W + W, FALSE;"SCORE:";s;" "; @ W,T;
640 INK YELLOW
650 FOR j = FALSE TO N - I
660   PAPER j + I
670   FOR i = FALSE TO M - I
680     PRINT CHR$(130);CHR$(131);
690     LET B(M * j + i) = TRUE
700   NEXT i
710 NEXT j
720 PAPER BLACK
730 INK WHITE
740 ENDPROC

```

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## VARIABLES USED

~~~~~

### Constants

-----

I = 1  
W = 3 :Width of char.  
H = 10 :Height of char.  
N = 5 :No. of lines of bricks  
M = 20 :No. of bricks in line  
T = 51 :Top row of bricks  
X = 123:Right col. of wall  
Y = 221:Bat row position  
L = 63 :Volume of beep

### Variables

-----

h = high score  
s = score  
n = balls left  
a,b = position of ball  
u,v = direction of ball  
x = position of bat  
k = direction of bat  
i,j = loop indices  
K\$ = keyboard char. pressed

### Arrays

-----

B (M \* N) = brick in place or not ?

## PROGRAM STRUCTURE

~~~~~

| Lines     | Use                                                              |
|-----------|------------------------------------------------------------------|
| 20        | Graphic char. for bat , ball and brick<br>( each 6 x 10 pixels ) |
| 30 - 40   | Variables definitions                                            |
| 50 - 70   | Set up graphics                                                  |
| 80 - 200  | One game                                                         |
| 110 - 140 | One move                                                         |
| 210 - 230 | Clean up                                                         |
| 240 - 300 | Bat move                                                         |
| 310 - 410 | Ball - move ball and check for hit and out                       |
| 420 - 480 | Check if brick in place                                          |
| 580 - 610 | New ball when out of play                                        |
| 620 - 780 | Brick and wall drawn                                             |

TENNIS  
~~~~~

(c) Colin I Clayman 1985

## DESCRIPTION

This program allows two players to play a 3 - set tennis match with normal scoring.

The player on the left (1) uses the up and down arrows to control his bat. The player on the right (2) uses the left and right arrow keys to control his bat.

At the end of a match, you can press the 'N' key to quit, or the 'Y' key for another match.

```

10 REM **** TENNIS BY COLIN I CLAYMAN *****
20 CODE 1E 00 00 00 OC 1E 1E OC 00 00 00
   00
30 LET I = 1, P = I + I, W = P + I, S = P + W, H = 10, T = 30, OC
   B = 220, V = 63, D$ = CHR$(24), N$ = CHR$(25)
40 DIM q(I), g(I), s(I), p(I), W(I), Q$(P)$(S)
50 LET W(FALSE) = W + W, W(I) = 120
60 RANDOM
70 DPOKE GRAPHIC,LCTN(H + H)
80 PROTECT BLACK
90 PROC FITCH
100 REPEAT
110 PRINT @ W,W;D$CHR$(T)N$;
120 LET s(I) = FALSE, s(FALSE) = FALSE, g = RAND(P)
130 REPEAT
140 LET g(FALSE) = FALSE, g(I) = FALSE
150 REPEAT
160 LET q(FALSE) = FALSE, q(I) = FALSE, g = NOTg
170 REPEAT
180 LET p(FALSE) = 130, p(I) = p(FALSE), x = W(g), y = p(I) +
   S, u = (RAND(S) + S) * SGN(g + g - I), v = SGN(RAND(P) -
   0.5), z = TRUE
190 REPEAT
200 PROC BAT(NOTg)
210 PROC BAT(g)
220 PROC BALL
230 UNTIL x < W(FALSE) OR x > W(I) + W
240 LET w = x < W (FALSE)
250 PROC CLEAR
260 PROC POINT
270 UNTIL q(w) - q(NOTw) > = P AND q(w) > W
280 PROC GAME
290 UNTIL g(w) - g(NOTw) > = P AND g(w) > S
300 LET s(w) = s(w) + I
310 UNTIL s(w) = P
320 PRINT D$"Game, Set & Match to "w + I;N$;
330 REPEAT
340 LET K$ = GET$,K$ = UPC$(K$)
350 UNTIL K$ = "Y" OR K$ = "N"

```

```

360 UNTIL K$ = "N"
370 DPOKE GRAPHIC, LETTER(32)
380 PROTECT BLACK
390 END
400 DEFPROC BAT(p)
410 PROTECT RED
420 LET k = INP(&0080 + &0900 * p), k = (NOT(k BNAND &0020) AND
    p(p) <= B - H) - (NOT(k BNAND &0010 - &000C * p) AND
    p(p) >= T + H)
430 IF k THEN PRINT @ W(p), p(p) + H * (k = -I); " ";
440 LET p(p) = p(p) + H * k
450 IF k OR z THEN PRINT @ W(p), p(p)/P;D$CHR$(128)N$;
460 ENDPROC
470 DEFPROC BALL
480 PROTECT CYAN
490 PRINT @ x,y;" ";
500 LET x = x + u, y = y + v, z = FALSE
510 IF (x > W(FALSE) + P OR ABS(y - p(FALSE) - S) > H + I) AND
    (x < W(I) - P OR ABS(y - p(I) - S) > H + I) THEN GOTO 540
520 LET u = - u, v = (RAND(W + W) + W) * SGN(v), x = W(u < FALSE)
    + P * SGN(u)
530 BEEP 1000,H,V
540 PRINT @ x,y;CHR$(129);
550 IF y >= T - W AND y < B + 14 THEN ENDPROC
560 LET v = -v
570 BEEP 500,H,V
580 ENDPROC
590 DEFPROC CLEAR
600 FOR i = I TO H
610 BEEP H * H,H * H,V
620 NEXT i
630 PRINT @ x,y;" ";
640 PROTECT RED
650 FOR p = FALSE TO I
660 PRINT @ W(p), p(p)/P;D$" "N$;
670 NEXT p
680 ENDPROC
690 DEFPROC POINT
700 LET q(w) = q(w) + (q(NOTw) <= w), q(NOTw) = q(NOTw) -
    (q(NOTw) > w)
710 PRINT @ H * H, W;D$Q$(q(FALSE));"--"Q$(q(I));N$;
720 ENDPROC
730 DEFPROC GAME
740 LET g(w) = g(w) + I
750 PRINT @ W + P * H * (s(FALSE) + s(I)), W;D$g(FALSE);"--"
    g(I);:"CHR$(T)N$;
760 ENDPROC
770 DEFPROC PITCH
780 CLS
790 INK GREEN
800 FOR i = I TO W
810 MOVE W(FALSE) * P - i,T - i
820 DRAW W(I) * P + S + i,T - i
830 DRAW W(I) * P + S + i, B + H + H + i
840 DRAW W(FALSE) * P - i, B + H + H + i
850 DRAW W(FALSE) * P - i,T - i
860 MOVE W(I) + W(FALSE) + i,T - i
870 DRAW W(I) + W(FALSE) + i, B + H + H + i
880 NEXT i
890 INK WHITE
900 FOR i = FALSE TO S

```

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

910 READ Q$(i)
920 NEXT i
930 ENDPROC
940 DATA 0 ,15,30,40,A ,A

```

## VARIABLES USED

### Constants

```

I = 1
P = 2 :No. of players
W = 3 :No. of sets
S = 5 :No. of games points
H = 10 :Height of char.
T = 30 :Top row of pitch
B = 220 :Bottom of pitch
V = 63 :Volume of beep
D$ = CHR$(24) :Double height
N$ = CHR$(25) Normal height

```

### Arrays

```

q(I) = point scores
g(I) = game scores
s(I) = set scores
p(I) = bat vertical position (moveable)
w(I) = bat horizontal position (fixed)
Q$(3)(S) = point names (0,15,30,40,A)

```

## PROGRAM STRUCTURE

### LINES

```

20
30 - 50
60 - 90
100 - 360
130 - 310
150 - 290
170 - 270
190 - 230
370 - 390
400 - 460
470 - 580
590 - 680
690 - 720
730 - 760
770 - 940

```

### USE

```

Graphic char. for bat and ball
(each 6 x 10 pixels)
Variables definitions
Set up graphics and pitch
One match
One set
One game
One point
One move
Clean up
Bat (p) - move player p
Ball - move ball and check for hits and out
Clear bats and ball
Point score increase and print
Game score increase and print
Pitch drawn

```

### Variables

```

g = server
x,y = position of ball
u,v = direction of ball
z = service ?
p = player
w = winner
k = direction of bat
i = loop index
K$ = keyboard char. pressed

```

## S Q U A S H

~~~~~

(c) Colin I Clayman 1985

## DESCRIPTION

This program plays a squash game for 1 player. It is similar to the 2-player tennis game.

Your aim is to get most hits in the 21 balls without the ball going out of play past your bat. The ball bounces on the 3 walls of the court. The highest score to date is shown.

The player uses the up and down arrows to control his bat.

At the end of a match, you can press 'N' key to quit, or the 'Y' key for another game.

=====

```

10 REM *** SQUASH COLIN I CLAYMAN 1985 *****
20 CODE 1E 00 00 00 00 OC 1E 1E OC 00 00
   00
30 LET I = 1, P = I + I, W = P + I, H = 10, L = H, R = 120,
   T = 30, B = 220, V = 63, h = FALSE, D$ = CHR$(24),
   N$ = CHR$(25)
40 RANDOM
50 DPOKE GRAPHIC,LCTN(H + H)
60 PROTECT BLACK
70 PROC PITCH
80 REPEAT
90 LET s = FALSE, b = FALSE
100 PROC SCORE
110 REPEAT
120 LET p=130, x=L, y=p+W+P, u=-(RAND(W+W)+W), v=SGN(RAND(P)-
   0.5), b=b+I
130 REPEAT
140   PROC BAT
150   PROC BALL
160   UNTIL x < L
170   PROC CLEAR
180   PROC SCORE
190 UNTIL b > H + H
200 IF s > h THEN LET h = s
210 PROTECT MAGENTA
220 PRINT D$" Game Over!" @ W,W,;"Hi Score:"h;"."N$;
230 REPEAT
240 LET K$ = GET$,K$ = UPC$(K$)
250 UNTIL K$ = "Y" OR K$ = "N"
260 UNTIL K$ = "N"
270 DPOKE GRAPHIC,LETTER(32)
280 PROTECT BLACK
290 END
300 DEFPROC BAT
310 PROTECT RED
320 LET k = INP(&00B0),k = (NOT(k BNAND &0020) AND p < = B - H) -
   (NOT(k BNAND &0010) AND p > = T + H)

```

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

330 IF k THEN PRINT @ L,p + H * (k = -I); " ";
340 LET p = p + H * k
350 PRINT @ L, p/P; D$CHR$(128)N$;
360 ENDPROC
370 DEFFPROC BALL
380 PROTECT CYAN
390 PRINT @ x,y;" ";
400 LET x = x + u,y = y + v
410 IF x > L + P OR ABS(y - p - W - P) > H + I THEN GOTO 440
420 LET u = -u, v = (RAND(W + W) + W) * SGN(v), x = L + P
430 BEEP 1000,H,V
440 PRINT @ x,y; CHR$(129);
450 If y > T = W AND y < B + 14 AND x <= R - P THEN ENDPROC
460 IF x > R - P THEN LET u = - u, s = s + I
470 ELSE LET v = - v
480 BEEP 500,H,V
490 ENDPROC
500 DEFFPROC CLEAR
510 FOR i = I TO H
520 BEEP H * H,H * H,V
530 NEXT i
540 PRINT @ x,y;" ";
550 PROTECT RED
560 PRINT @ L, p/P;" "N$;
570 ENDPROC
580 DEFFPROC
590 PROTECT RED
600 PRINT @ 45, W; D$s;" Hits in "b;"."CHR$(T)N$;
610 ENDPROC
620 DEFFPROC PITCH
630 CLS
640 FOR i=I to W
650 INK GREEN
660 MOVE L * P - i, B + H + H + i
670 DRAW R * P + W + P + i, B + H + H + i
680 DRAW R * P + W + P + i, T - i
690 DRAW L * P - i, T - i
700 INK BLUE
710 PLOT I, FALSE, i + H
720 FOR k = I to H
730 PLOT W, FALSE, H
740 PLOT I, FALSE, H
750 NEXT k
760 NEXT i
770 INK WHITE
780 ENDPROC

```

=====

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## VARIABLES USED

~~~~~

### Constants

-----  
I = 1  
P = 2  
W = 3 :Width of char.  
H = 10 :Height of char  
L = 10 :Left col. of pitch  
R = 120 :Right of pitch  
T = 30 :Top row of pitch  
B = 220 :Bottom of pitch  
V = 63 :Volume of beep  
D\$ = CHR\$(24) :Double height  
N\$ = CHR\$(25) :Normal "

### Variables

-----  
h = high score  
s = score  
b = balls played  
x,y = position of ball  
u,v = direction of ball  
p = position of bat  
k = direction of bat  
i = loop index  
K\$ = keyboard char.pressed

## PROGRAM STRUCTURE

~~~~~

| Lines     | Use                                                 |
|-----------|-----------------------------------------------------|
| 20        | Graphic char. for bat and ball (each 6 x 10 pixels) |
| 30        | Variables definitions                               |
| 40 - 70   | Set up graphics and pitch                           |
| 80 - 260  | One game                                            |
| 110 - 190 | One ball                                            |
| 130 - 160 | One move                                            |
| 270 - 290 | Clean up                                            |
| 300 - 360 | Bat moved                                           |
| 370 - 490 | Ball - move ball and check for hits and out         |
| 500 - 570 | Clear bat and ball                                  |
| 580 - 610 | Score increased and printed                         |
| 620 - 780 | Pitch drawn                                         |

=====

MASTER MIND  
~~~~~

(c) Colin I Clayman 1985

## DESCRIPTION

This is a logic guessing game in which the computer thinks of a number and you have to guess it by giving it numbers of the same length. For each guess you are told:

- how many digits tally in their right place: given by WHITE pegs
- how many other digits tally, but in the wrong place: given by BLACK pegs

If you give a number of the wrong length, you are asked if you want to give up. If you do, the number is revealed and the game ends.

To start the game you are asked how many digits you want, between 1 and 9. 5 digits is often a good game to play.

```

10 REM **** MASTER MIND COLIN I CLAYMAN 1985 ****
20 LET i = 1, p = 3, d = 10, q = 235, L = p * d, N = i, C$ = CHR$(L)
30 CLS
40 RANDOM
50 PAPER BLUE
60 INK YELLOW
70 PRINT CHR$(24) " M A S T E R M I N D " C$ CHR$(25);
80 WINDOW p, 123, q, q + d
90 PRINT @ p, q; "How many digits(1:9)?";
100 REPEAT
110 LET D = ASC(GET$) - ASC("0")
120 UNTIL D > = i AND D < d
130 PRINT @ p, L; "GO GUESS ";
140 INK WHITE
150 PRINT " RIGHT ";
160 INK BLACK
170 PRINT " WRONG ";
180 INK CYAN
190 PRINT D; " Digits"; C$;
200 DIM V$(D), H(D)
210 FOR V = i TO D
220 LET V$=V$ + STR$(RAND(d))
230 NEXT V
240 REPEAT
250 LET R = FALSE, W = R, L = L + d
260 PAPER BLUE
270 INK YELLOW
280 REPEAT
290 PRINT @ p, q; C$ " Your guess "; N;
300 INPUT G$
310 IF LEN(G$) = D THEN GOTO 360
320 PRINT "Wrong Length! Do you give up (Y/N)?";
330 REPEAT

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

340 LET K$ = GET$, K$ = UPC$(K$)
350 UNTIL K$ = "Y" OR K$ = "N"
360 UNTIL LEN(G$) = D OR K$ = "Y"
370 PAPER RED
380 IF K$ = "Y" THEN PRINT @ p,L"No.=";V$;C$;
390 IF K$ = "Y" THEN GOTO 610
400 FOR V = i TO D
410   IF MID$(V$,V,i) = MID$(G$,V,i) THEN LET R = R + i,H(V) = i
420   ELSE LET H(V) = FALSE
430 NEXT V
440 FOR V = i TO D
450   IF H(V) = i THEN GOTO 510
460   LET G = FALSE
470   REPEAT
480     LET G = G + i,F = V < > G AND MID$(V$,V,i) =
        MID$(G$,V,i)
        AND NOT H(G)
490   UNTIL F OR G = D
500   IF F THEN LET W = W + i,H(G) = 2
510 NEXT V
520 PRINT @ p,L;N;C$; TAB 5;G$; TAB 19;
530 FOR G = i TO D
540   IF G < = R THEN INK WHITE
550   ELSE IF G < = R + W THEN INK BLACK
560   ELSE INK YELLOW
570   IF G < = R + W THEN PRINT CHR$(232);
580   ELSE PRINT ".";
590 NEXT G
600 LET N = N + i
610 UNTIL R = D OR K$ = "Y"
620 PAPER GREEN
630 INK RED
640 IF R = D THEN PRINT ,"WELL DONE!";
650 ELSE PRINT ,"YOU GAVE UP!";
660 PAPER BLACK
670 INK WHITE
680 WINDOW p,123,5,245

```

. +=====

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## VARIABLES USED

### Constants

```
-----  
i = 1  
p = 3 :width of char.  
d = 10 :depth of char.  
q = 235 : window row position  
C$ = CHR$(30) :clear to end of line
```

### Variables

```
-----  
D = no. of digits  
N = guess no.  
L = line position  
V = place within number  
G = place within guess  
R = no.of digits in right  
place  
W = no.of digits in wrong  
place  
V$ = number  
G$ = guess  
K$ = keyboard char. pressed
```

## ARRAYS

```
-----  
H (D) = 0 :digit wrong, 1 :in right place or 2 :in wrong place
```

## PROGRAM STRUCTURE

```
-----
```

Lines	Use
20	Variables definition
30 - 200	Set up screen
90 - 120	Get no. of digits
210 - 230	Think of a number
240 - 610	One guess
280 - 390	Have a guess and check for giving up
400 - 430	Check for digits in right place
440 - 510	Check for digits in wrong place
520 - 600	Show guess and WHITE and BLACK pegs
620 - 650	Give result
660 - 680	Clean up

```
-----
```

S N A K E   B I T E  
~~~~~  
(c) Colin I Clayman 1985

**DESCRIPTION**

-----  
Guide the snake, using the arrow keys, to eat the green Frogs. When you do a poisonous Blue Frog appears, or, more rarely, a Magic Mushroom. You may eat the Mushroom before it disappears again and you may also eat a Blue Frog but only immediately after eating a Mushroom.

You must not crash into yourself or a red wall, or a Blue Frog when you have not just eaten a Mushroom; otherwise the Snake dies and the game is over.

When you have eaten all the Green Frogs and remaining Mushrooms the game restarts with first one pair of interior walls and next with two pairs. If you get through that the game restarts at a faster speed, until the highest speed level 9.

The program keeps track of the highest score in all games played in one run.

To play just answer:

"What skill level ( 0 : 9 ) ?"

at the start. RETURN on its own means 0, the lowest speed.

-----

```

10 REM *** SNAKE BITE COLIN I CLAYMAN 1985 *****
20 LET I = 1, W = 3, H = 10, X = 42, Y = 25, D = 7, G = 12, S = 6, V =
   I + I, L = 63, z = FALSE, I$ = CHR$(18)
30 DIM A(X * Y), C(S), S$(I)(S)
40 REM ***** CHRS 128 - 131 *****
50 CODE 3F 00 2D 1E 1E 0C 1E 3F 3F 12
   33 00 00 0C 1E 3F 3F 0C 0C 0C 00 00 0C 12 21 2D 2D 21 12 0C 00
60 REM *** 0 = EMPTY, 1 = G.FROG, 2 = MUSHROOM, 3 = B.FROG, 4 & 5
   WALLS, 6 = SNAKE
70 DATA white, 128, green, 129, cyan, 130, blue, 129, red, 128,
   red, 128, magenta, 131
80 DPOKE GRAPHIC, LCTN(50)
90 RANDOM
100 PROC SETUP
110 PROC INTRO
120 REPEAT
130   LET s = FALSE
140   REPEAT
150     LET v = -I
160     REPEAT
170       LET l = I, f = H + H, g = 5, h = D * X + X + I + RAND(X
         - I - g - 1) + X * RAND(Y - W - D - D), t = h, d = I, m
         = FALSE, e = FALSE, v = v + I
180   PROC STAGE
190   REPEAT
200     FOR i = I TO H - u
210       LET k = KEYN
220       IF k = 12 THEN LET d = I
230       IF k = 22 THEN LET d = -I
240       IF k = 10 THEN LET d = X

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

250      IF k = 11 THEN LET d = -X
260      NEXT i
270      PROC CHECK(A(h + d))
280      IF k > - I THEN PROC SNAKE
290      IF m AND NOT RAND(H * H) THEN PROC NOMUSH
300      UNTIL k = - I OR NOT f + m + m
310      UNTIL k = - I OR v = V
320      LET u = u + (k > - I AND u < H - I)
330      UNTIL k = - I OR u = H - I
340      PROC DIE
350      PROC HISCORE
360      PRINT @ G,241;I$;
370      IF k = - I THEN PRINT "CRASHED !!";
380      ELSE PRINT "FULL UP !!";
390      PRINT " Another Game (Y/N) ?";I$;
400      REPEAT
410      LET K$ = GET$, K$ = UPC$(K$)
420      UNTIL K$ = "Y" OR K$ = "N"
430      UNTIL K$ = "N"
440      INK WHITE
450      PAPER BLACK
460      DPOKE GRAPHIC, LETTER (32)
470      END
480      DEFFPROC PLACE(p,n)
490      IF n < S THEN LET A(p) = n
500      INK C(n)
510      PRINT @ p MOD X * W + I, p DIV X * H + I;S$(n);
520      ENDPROC
530      DEFFPROC SNAKE
540      LET A(h) = h + d, h = h + d, A(h) = h
550      PROC PLACE(h,S)
560      IF NOT g THEN LET p = t, t = A(t)
570      IF g THEN LET g = g - I, I = I + I
580      ELSE PROC PLACE(p, FALSE)
590      ENDPROC
600      DEFFPROC CHECK(p)
610      LET k = 25 * (p = I) + H * H * (p = V) + 50 * (p = W AND e)
620      IF p = V THEN LET m = FALSE, e = TRUE
630      ELSE IF p THEN LET e = FALSE
640      IF p = I THEN PROC CREATE
650      IF k > I THEN LET s = s + k, g = g + W
660      IF k > I THEN PROC SCORE
670      IF k > I THEN BEEP H * k,W * H,L
680      ENDPROC
690      DEFFPROC CREATE
700      LET f = f - I, n = W
710      REPEAT
720      LET p = RAND(X * Y)
730      UNTIL NOT A(p)
740      IF NOT m THEN LET m = p * NOT RAND(H), n = n - (m = p)
750      PROC PLACE(p,n)
760      IF n < W THEN BEEP H * H,H * H,L
770      ENDPROC
780      DEFFPROC NOMUSH
790      PROC PLACE(m, FALSE)
800      BEEP H * H, H * H, L
810      LET m = FALSE
820      ENDPROC
830      DEFFPROC SCORE
840      INK RED

```

```

850 PRINT @ I,I;"Your Score:";s;i$;
860 ENDPROC
870 DEFPROC HISCORE
880 IF s > z THEN LET z = s
890 INK RED
900 PRINT @ 78, I;I$"High Score:"z;i$;
910 ENDPROC
920 DEFPROC DIE
930 REPEAT
940   LET p = t, t = A(t)
950   PROC PLACE(p, FALSE)
960   BEEP 1000,H * H, L
970 UNTIL t = h
980 ENDPROC
990 DEFPROC SETUP
1000 FOR i = FALSE TO X * Y
1010   LET A(i) = FALSE
1020 NEXT i
1030 FOR i = FALSE to S
1040   READ C(i),x
1050   LET S$(i) = CHR$(x)
1060 NEXT i
1070 ENDPROC
1080 DEFPROC STAGE
1090 FOR i = FALSE TO X * Y
1100   IF A(i) THEN IF A(i) < > S - I AND (A(i) < > S - V OR NOT
      v) THEN PROC PLACE(i, FALSE)
1110 NEXT i
1120 IF v THEN GOTO 1220
1130 FOR i = FALSE TO X - I
1140   IF K$ = "" THEN PROC PLACE(i,S-I)
1150   PROC PLACE(Y * X - X + i,S - I)
1160 NEXT i
1170 FOR i = I TO (Y - I) * (K$ = "")
1180   PROC PLACE(i * X,S - I)
1190   PROC PLACE(i * X + X - I,S - I)
1200 NEXT i
1210 GOTO 1260
1220 FOR i = I TO D
1230   PROC PLACE(i * X + G + (X - I - G - G) * (v = V),
      S - I - I)
1240   PROC PLACE(Y * X - X - i * X + G + (X - I - G - G) *
      (v = V), S - V)
1250 NEXT i
1260 PROC PLACE(h,S)
1270 FOR i = I to g
1280   PROC SNAKE
1290 NEXT i
1300 FOR i = I TO f
1310   REPEAT
1320     LET p = RAND(X * Y)
1330     UNTIL NOT A(p)
1340     PROC PLACE(p, I)
1350 NEXT i
1360 PROC SCORE
1370 IF NOT s THEN PRINT I$"      ";i$;
1380 PRINT @ 48,I;I$"STAGE "ui"."v;i$;
1390 BEEP H * H, H * H * H,L
1400 ENDPROC
1410 DEFPROC INTRO
1420 PAPER WHITE

```

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

1430 CLS
1440 INK GREEN
1450 PRINT CHR$(24),,"SNAKE BITE"
1460 PRINT,"(c) Colin Clayman, 1985."
1470 VDU 25,H,H,H
1480 PRINT "You have to guide the SNAKE";
1490 PROC SYM(S)
1500 PRINT ",using thearrow keys, to eat the GREEN FROGS";
1510 PROC SYM(I)
1520 PRINT ". When you eat one, a POISONOUS BLUE FROG";
1530 PROC SYM(W)
1540 PRINT "will appear, or occassionally a MAGIC MUSHROOM";
1550 PROC SYM(V)
1560 PRINT". You may eat the MUSHROOM", "before it disappears. You
    may also eat aBLUE FROG, but only immediately after eating
    a MUSHROOM. You must not crash into yourself or the walls";
1570 PROC SYM(S - I)
1580 PRINT ." CHR$(31)"The game gets more difficult in stages,
    and it keeps track of the highest score."
1590 PRINT @ 30,H * H;CHR$(24)"What skill level (0:9) ?"CHR$(25);
1600 REPEAT
1610 LET D$ = GET$, u = VAL(D$)
1620 UNTIL D$ > " / " AND NOT D$ > "9" OR D$ = """
1630 CLS
1640 ENDFROC
1650 DEFFPROC SYM(n)
1660 INK C(n)
1670 PRINT " " S$(n);"
1680 INK GREEN
1690 ENDFROC

```

=====

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## VARIABLES

~~~~~

### Constants

```
-----
I = 1
W = 3 :Width of char.
H = 10 :Height of char.
X = 42 :Width of screen
Y = 25 :Height of screen
D = 7 :Depth of interior walls
G = 12 :Gap between walls
S = 6 :No. of symbols
V = 2 :No. of stages -1
L = 63 :Beep volume
I$ = CHR$(18) :Inverse display
```

### Variables

```
-----
s = score
z = high score
v = stage no.
u = level no.
l = length of snake
g = growth fo snake
h = position of snake head
t = position of snake tail
d = direction of snake
f = no. of frogs
m = is there a mushroom?
e = just eaten mushroom?
k = key read/ eaten this go
i = loop index
```

## ARRAYS

~~~~~

C (S) Colour of symbols

S\$(1)(S) Symbols

A (X \* Y) Screen contents :

```
0 = Empty
1 = Green frog
2 = Mushroom
3 = Blue frog
4 = Interior wall
5 = Exterior wall
> 6 = Snake - pointer to next segment in
front.
```

## PROGRAM STRUCTURE

~~~~~

### LINES

```
-----
20 - 30
40 - 50
60 - 70
80 - 110
120 - 430
140 - 330
160 - 310
440 - 470
480 - 520
530 - 590
600 - 680
690 - 770
780 - 820
830 - 860
870 - 910
920 - 980
990 - 1070
1080 - 1400
1650 - 1690
```

### USE

```
-----
Variables and array definitions
Graphics characters
Symbol definition
Set up
1 game
1 level
1 move
Clean up & end
Place (p,n) - write symbol n at position p
SNAKE moves
CHECK (p) - what effect of hitting p ?
CREATE a blue frog or mushroom
NOMUSH - mushroom disappears
SCORE shown
HISCORE - show high score
DIE - snake dies
SET UP symbols and array A
INTRO - instructions
SYm (n) - write symbol n
```

-----

D I S A S S E M B L E R  
 ~~~~~~  
 (c) Colin I Clayman 1985

**DESCRIPTION**

This BASIC procedure will disassemble machine code programs into standard Z80 op-code mnemonics with hexadecimal operands.

Any unrecognised instruction will be output as hexadecimal.

The listing is printed to the screen, but by changing the PRINTs into LPRINTs it can be sent to a printer. The listing may be stopped, and restarted, at any time by pressing any key.

This has been written as a procedure so that it may be appended to any program and called to disassemble any addresses; for example to disassemble the LYNX ROM, use it with:-

```
10 REM ROM DISASSEMBLY
20 PROC DISAS( 0, &3FFF)
30 END
```

The procedure works, basically by looking at the various bits of the op-code byte and testing these in IF statements to construct the mnemonics. Accordingly, although written entirely in BASIC, it is quite efficient as it makes use of the patterns inherent in the Z80 instruction set.

**NOTE \*\*\***

- 1) For the £ sign type a hash sign (#)
- 2) Be careful to type the right no. of spaces in strings otherwise the instructions will be garbage.

```
100010 DEFPROC DISAS (A,B)
100020 LET I = 1, P = 3, E = 7, H = &0010, Q = H*H, X =&000A C7 = 0F
100030 DIM R$(I + I)(E), C$(P)(E), X$(P + P)(E), S$(P)(P), Q$ (P)(P),
Q$(P)(P), Y$(P)(P), W$(72)
100040 PROC SETUP
100050 CLS
100060 FOR a = A TO B
100070 PROC OP(a)
100080 NEXT a
100090 ENDPROC
100100 DEFPROC OP(a)
100110 PRINT fa;" : ";
100120 LET c = FALSE, x = c, A$ = ""
100130 REPEAT
100140 PROC TYPE(PEEK(a))
100150 UNTIL q
100160 IF NOT c THEN PROC NORMAL
100170 IF c = I THEN PROC CB-OP
100180 IF c = I + I THEN PROC ED-OP
100190 PROC CONV
100200 IF KEYN THEN LET q = GETN
100210 ENDPROC
100220 DEFPROC TYPE (b)
```

```

100230 REM h = HIGH b76, m = MID b543, l = LOW b210,
p = PLANE b54, r = RIGHT b3
100240 LET h = b DIV &0040, m = (b BNAND &0038) DIV (E + I), l =
b BNAND E, p = m DIV (I + I), r = m BNAND I, q = TRUE
100250 IF NOT c AND (b = &00CB OR b = &00ED) THEN LET c =
I + (b = &00ED), a = a + I + (x >= I), q = FALSE
100260 IF NOT c + x AND (b = &00DD OR b = &00FD) THEN LET x =
I + (b = &00FD), a = a + I, q = FALSE
100270 ENDPROC
100280 DEFPROC NORMAL
100290 IF b = &0076 THEN LET A$ = "HALT"
100300 ELSE IF NOT h THEN PROC NORMAL0
100310 ELSE IF h = I THEN LET A$ = "LD " + R$(m) + R$(l)
100320 ELSE IF h = I + I THEN LET A$ = X$(m) + R$(l)
100330 ELSE PROC NORMAL3
100340 ENDPROC
100350 DEFPROC NORMAL0
100360 IF NOT l AND m > P THEN LET A$ = "JR " +
C$(m - P - I) + "n"
100370 ELSE IF NOT l THEN LET W$ =
"NOP      EX AF AF'DJNZ n    JR n",
A$ = MID$(W$, X * m - m + I, X - I)
100380 ELSE IF l = I AND r THEN LET A$ = "ADD y " + S$(p)
100390 ELSE IF l = I THEN LET A$ + "LD " + S$(p) + "d"
100400 ELSE IF l = I + I THEN LET W$ =
"(BC) AA (BC) (DE) AA (DE) (d) y y
(d) (d) A A (d)", A$ = "LD " + MID$(W$, 6 * m + I, P + P)
100410 ELSE IF l = P THEN LET W$ = "INCDEC", A$ =
MID$(W$, P * r + I, P) + " " + S$(p)
100420 ELSE IF l = P + I THEN LET A$ = "INC " + R$(m)
100430 ELSE IF l = P + I + I THEN LET A$ = "DEC " + R$(m)
100440 ELSE IF l = P + P THEN LET A$ = "LD " + R$(m) + "n"
100450 ELSE LET W$ = "RLCARRCARLA RRA DAA CPL SCF CCF", A$ =
MID$(W$, P * m + m + I, P + I)
100460 ENDPROC
100470 DEFPROC NORMAL3
100480 IF NOT l THEN LET A$ = "RET " + C$(m)
100490 ELSE IF l = I AND r THEN LET W$ =
"RET    EXX    JP (y) LD SP y",
A$ = MID$(W$, E * p + I, E)
100500 ELSE IF l = I THEN LET A$ = "POP " + Q$(p)
100510 ELSE IF l = I + I THEN LET A$ = "JP " + C$(m) + "d"
100520 ELSE IF l = P THEN LET W$ =
"JP d          OUT (n) AIN A (n) EX (SP) yEX DE HL DI      EI"
A$ = MID$(W$, X * m - m + I, X - I)
100530 ELSE IF l = P + I THEN LET A$ = "CALL " + C$(m) + "d"
100540 ELSE IF l = P + I + I AND NOT x AND r THEN LET A$ =
"CALL d"
100550 ELSE IF l = P + I + I AND NOT r THEN LET A$ =
"PUSH " + Q$(p)
100560 ELSE IF l = P + P THEN LET A$ = X$(m) + "n"
100570 ELSE IF l = E THEN LET W$ = "0008101820283038", A$ =
"RST &" + MID$(W$, m + m + I, I + I)
100580 ENDPROC
100590 DEFPROC CB-OP
100600 IF h THEN LET W$ = " BITRESSET", A$ =
MID$(W$, P * h + I, P) + " " + STR$(m) + " " + R$(l)
100610 ELSE IF m < > P + P THEN LET W$ =
"RLCRRCRL RR SLASRA SRL", A$ =
MID$(W$, P * m + I, P) + " " + R$(l)
100620 ENDPROC

```

```

100630 DEFFPROC ED-OP
100640 IF NOT h OR h = P OR h = I + I AND (I > P OR m <= P)
    THEN ENDPROC
100650 IF h = I + I THEN LET W$ = "LDCPINOT", V$ =
    "I D IRDR", A$ =
    MID$(W$,1 + 1 + I,I + I) + MID$(V$, m + m - E,I + I)
100660 ELSE IF NOT I THEN LET A$ = "IN " + R$(m) + "(C)"
100670 ELSE IF I = I THEN LET A$ + "OUT (C)" + R$(m)
100680 ELSE IF I = I + I THEN LET W$ = "SBCADC", A$ =
    MID$(W$,P * r + I,P) + " HL " + S$(p)
100690 ELSE IF I = P AND r THEN LET A$ = "LD " + S$(p) + "(d)"
100700 ELSE IF I = P THEN LET A$ = "LD (d)" + S$(p)
100710 ELSE IF I = P + I AND NOT m THEN LET A$ = "NEG"
100720 ELSE IF I = P + I + I AND NOT p THEN LET W$ =
    "NI", A$ = "RET" + MID$(W$,r + I ,I)
100730 ELSE IF I = P + P AND m <= P AND m <> I THEN LET W$ =
    "IM 0 IM 1IM 2", A$ = MID$(W$,P * m + m + I,P + I)
100740 ELSE IF I = E THEN LET W$ = "LD I ALD R ALD A ILD A RRRD RLD",
    A$ = MID$(W$,6 * m + I,P + P)
100750 ENDPROC
100760 DEFFPROC CONV
100770 IF A$ = "" THEN LET a = a - NOT x, W$ = "&CB&ED&DD&FD"
100780 IF A$ = "" THEN PRINT MID$(W$,P * (x + I +(c - I - I) * NOT x)
    + I,P)
100790 IF A$ = "" THEN ENDPROC
100800 FOR i = I TO LEN(A$)
100810 LET B$ =MID$(A$,i,I)
100820 IF B$ = "x" THEN PROC INDEX
100830 ELSE IF B$ = "y" THEN PRINT Y$(x);
100840 ELSE IF B$ = "d" THEN PROC ADDR
100850 ELSE IF B$ = "n" THEN PROC BYTE
100860 ELSE PRINT B$;
100870 NEXT i
100880 PRINT
100890 ENDPROC
100900 DEFFPROC INDEX
100910 PRINT "("
100920 PRINT Y$(x);
100930 IF x THEN PRINT "+";
100940 IF x THEN PROC PBYTE(PEEK(a + I -2 * (c >= I)))
100950 IF x THEN LET a = a + (NOTc)
100960 PRINT ")";
100970 ENDPROC
100980 DEFFPROC ADDR
100990 PRINT EDPEEK(a + I);
101000 LET a = a +I +I
101010 ENDPROC
101020 DEFFPROC BYTE
101030 LET a = a + I
101040 PROC PBYTE(PEEK(a))
101050 ENDPROC
101060 DEFFPROC HEX(h)
101070 IF h < X THEN PRINT h;
101080 ELSE PRINT CHR$(ASC("A") + h - X);
101090 ENDPROC
101100 DEFFPROC PBYTE(b)
101110 PRINT "&";
101120 PROC HEX(b DIV H)
101130 PROC HEX(b MOD H)
101140 ENDPROC

```

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```
101150 DEFPROC SETUP
101160 DATA B ,NZ ,ADD A ,C ,Z ,ADC A ,D ,NC ,SUB ,E ,C ,SBC
      A ,H ,PO ,AND ,L ,PE ,XOR ,x ,P ,OR ,A ,M ,CP space
101170 DATA BC ,BC ,HL ,DE ,DE ,IX ,y ,y ,IY ,SP ,AF ,space
101180 RESTORE
101190 FOR i = FALSE TO E
101200   READ R$(i), C$(i), X$(i)
101210 NEXT i
101220 FOR i = FALSE TO P
101230   READ S$(i), Q$(i), Y$(i)
101240 NEXT i
101250 ENDPROC
```

=====

## VARIABLES USED

~~~~~

### Constants

-----  
I = 1  
P = 3 : No. of double registers  
E = 7 : No. of registers  
H = 16  
Q = 256  
X = 10

### Variables

-----  
A = start address  
B = end address  
a = current address  
b = current byte  
c = class of op-code  
 ( 0 :normal, 1:CB, 2:ED )  
x = index register  
 ( 0:none, 1:IX, 2:IY )  
h = high 2 bits (76) of op-code  
m = middle 3 bits (543)  
l = low 3 bits (210)  
p = bits 5 & 4  
r = bit 3  
q = prefix found?  
i = loop index  
W\$ = string of instructions  
A\$ = disassembled instruction

### Arrays

-----  
R\$(2)(E) = mnemonics for registers  
C\$(2)(E) = mnemonics for conditions  
X\$(2)(E) = mnemonics for arithmetic instructions  
S\$(3)(P) = }  
Q\$(3)(P) = } various mnemonics for double registers  
Y\$(3)(P) = }

# TEN CHIP-TEASING PROGRAMS FOR THE LYNX

## PROGRAM STRUCTURE

---

| LINES       | USE   |
|-------------|---|
| 100010-90   | DIAS (A,B) - Disassemble from address A to B  |
| 100020-40   | Variables definition & set up   |
| 100060-80   | Loop through addresses  |
| 100100-210  | OP (a) - Disassembles instruction at address a  |
| 100220-270  | TYPE (b) - Determines type of op-code<br>b and increases address accordingly  |
| 100280-340  | NORMAL - Interprets normal instructions;<br>in particular 8-bit Loads & Arithmetic  |
| 100359- 460 | NORMAL0 - Interprets instructions with h = 0  |
| 100470-580  | NORMAL3 - Interprets instructions with h = 3  |
| 100590-620  | CB-OP - Interprets instructions prefixed by CB  |
| 100630-750  | ED-op - Interprets instructions prefixed by ED  |
| 100760-890  | CONV - Prints disassembled instruction,<br>replacing: x by (HE), (IX+n) or (IY+n)<br>y by HL, IX or IY<br>d by 2-byte address<br>n by 1-byte data |
| 100900-970  | INDEX - Prints index & displacement   |
| 100980-1010 | ADDR - Prints 2-byte address  |
| 101020-1050 | BYTE - Print 1-byte data  |
| 101060-1090 | HEX (h) - Prints h as hex. char.  |
| 101100-1140 | PBYTE(b) - Prints byte b in hex.  |
| 101150-1250 | SETUP string arrays   |

---

## MUNCHER CRUNCH

~~~~~

(c) S J Sawyer 1985

## DESCRIPTION

You are a man in the middle of a maze. You have got to eat as many 'crunches' as possible before the monster catches you, using the arrow keys to move. The problem is that you cannot go back where the 'crunches' have been eaten.

```

100 REM **** MUNCHER CRUNCH S J SAWYER *****
110 RESERVE 38719
120 PAPER 0
130 CLS
140 PRINT CHR$(24); "LOADED"; CHR$(25);
150 DPOKE GRAPHIC, 39680
160 FOR J = 39680 TO 39680 + 1280
170 POKE J, PEEK(&01D4 + (J - 39680))
180 NEXT J
190 GOSUB LABEL SETUP
200 PROC CHR
210 GOSUB 600
220 RANDOM
230 LET S = 0, A = 460, X = 42 + (RAND(3) * 18), K = 878 -
(RAND(3) * 18), L = 0
240 VDU 1, 6, 2, 2
250 PRINT @ 97, 5; S;
260 VDU 1, 7, 2, 0
270 IF S MOD 100 = 0 THEN PROC BEEP
280 PRINT @ 3 * (A MOD 40), 10 * (A DIV 40) + 5; CHR$(128);
290 INK 4
300 PRINT @ 3 * (X MOD 40), 10 * (X DIV 40) + 5; CHR$(142); @ 3 *
(K MOD 40), 10 * (K DIV 40) + 5; CHR$(142);
310 POKE M + A, 128
320 POKE M + X, 142
330 POKE M + K, 142
340 LET Y = X, J = K, B = A
350 PROC KEY
360 IF PEEK(M + A) = 239 OR PEEK(M + A) = 128 THEN LET A = B
370 IF PEEK(M + A) = 142 THEN LET S = S + 1
380 PRINT @ 3 * (A MOD 40), 10 * (A DIV 40) + 5; CHR$(143);
390 POKE (M + A), 143
400 IF RAND(2) = 1 THEN GOTO 420
410 LET X = X + (40 * SGN((A DIV 40) - (X DIV 40)))
420 LET X = X + SGN(A MOD 40) - (X MOD 40)
430 IF X = A THEN GOTO LABEL END
440 IF PEEK(M + X) = 239 THEN LET X = Y
450 IF PEEK(M + X) = 40 THEN GOTO LABEL END
460 PRINT @ 3 * (X MOD 40), 10 * (X DIV 40) + 5; CHR$(192);
470 POKE (M + X), 192
480 IF RAND(2) = 1 THEN GOTO 500
490 LET K = K + (40 * SGN((A DIV 40) - (K DIV 40)))
500 LET K = K + SGN((A MOD 40) - (K MOD 40))
510 IF K = A THEN GOTO LABEL END
520 LET k = K
530 IF PEEK(M + K) = 239 THEN LET K = J
540 IF PEEK(M + k) = 239 AND L = J THEN LET K = K + 1

```

## TEN CHIP-TEASING GAMES FOR THE LYNX

```

550 IF PEEK(M + K) = 239 AND L = J THEN LET K = K - 2
560 IF PEEK(M + K) = 239 AND L = J THEN LET K = J
570 LET L = J
580 PRINT @ 3 * (K MOD 40), 10 * (K DIV 40) + 5;CHR$(192);
590 GOTO 240
600 REM BOARD
610 LET M = MHIMEM
620 FOR B = 0 TO 960
630 POKE M + B, 239
640 NEXT B
650 FOR J = 42 TO 882 STEP 80
660 FOR S = 0 TO 36
670 POKE M + J + S, 142
680 NEXT S
690 FOR S = 0 TO 37 STEP 2
700 POKE M + S + J + 40, 142
710 NEXT S
720 NEXT J
730 FOR J = 921 TO 960
740 POKE M + J, 128
750 POKE M + J - 40, 239
760 NEXT J
770 VDU 4
780 FOR B = 1 TO 960
790 IF PEEK(M + B) = 142 THEN INK RED
800 ELSE INK 1
810 PRINT CHR$(PEEK(M + B));
820 NEXT B
830 VDU 1, 7, 2, 4
840 PRINT @ 20,5;"MUNCHER CRUNCH";
850 VDU 1, 6, 2, 2
860 PRINT @ 70,5;" SCORE : 0";
870 VDU 1, 7, 2, 0
880 RETURN
890 DEFFPROC KEY
900 LET I = INP(&0080)
910 IF I = 255 THEN GOTO 950
920 IF I = 239 THEN LET A = A - 40
930 IF I = 223 THEN LET A = A + 40
940 ENDPROC
950 LET I = INP(&0980)
960 IF I = 251 THEN LET A = A - 1
970 IF I = 223 THEN LET A = A + 1
980 ENDPROC
990 DEFFPROC CHR
1000 FOR J = 0 TO 9
1010 READ A
1020 POKE LETTER(192) + J, A
1030 NEXT J
1040 FOR J = 0 TO 9
1050 READ A
1060 POKE LETTER(143) + J, A
1070 NEXT J
1080 FOR J = 0 TO 9
1090 READ A
1100 POKE LETTER(175) + J, A
1110 NEXT J
1120 FOR J = 0 TO 9
1130 READ A
1140 POKE LETTER(200) + J, A
1150 NEXT J

```

## TEN CHIP-TEASING GAMES FOR THE LYNX

```

1160 FOR J = 0 TO 9
1170   READ A
1180   POKE LETTER(210) + J,A
1190 NEXT J
1200 DATA 0, 12, 30, 45, 63, 30, 12, 18, 33, 0
1210 DATA 0, 4, 14, 4, 31, 4, 4, 10, 17, 0
1220 DATA 30, 63, 45, 45, 45, 45, 45, 45, 63, 30
1230 DATA 0, 28, 62, 63, 63, 63, 31, 31, 24
1240 DATA 0, 0, 16, 48, 48, 56, 56, 56, 48, 0
1250 ENDPROC
1260 DEFFPROC TABLE
1270 PRINT @ 30,80;"YOU SCORED : ";S;
1280 LET T = 3
1290 FOR J = 0 TO 2
1300   IF T(J) > S THEN GOTO 1390
1310   LET T = J, J = 2
1320   IF T = 2 THEN GOTO 1390
1330   FOR j = 2 TO T + 1 STEP -1
1340     LET T(j) = T(j -1)
1350     FOR N = 0 TO 9
1360       LET X((j * 10 + N)) = X((j -1) * 10 + N)
1370     NEXT N
1380   NEXT j
1390 NEXT J
1400 IF T = 3 THEN GOTO 1490
1410 LET T(T) = S
1420 VDU 1, 6, 2, 1
1430 PRINT @ 30,90;"Enter your name ";
1440 PRINT @ 30,100;
1450 INPUT S$
1460 FOR J = 0 TO 9
1470   LET X(T * 10 + J) = ASC(MID$, (S$,J + 1,1))
1480 NEXT J
1490 FOR J = 0 TO 2
1500   LET T$ = ""
1510   FOR j = 0 to 9
1520     LET T$ = T$ + CHR$(X(J * 10 + j))
1530   NEXT j
1540   VDU 1, 4, 2, 2
1550   PRINT @ 30, 110 + (J * 10);T(J);"";T$;
1560 NEXT J
1570 VDU 1, 7, 2, 0
1580 ENDPROC
1590 LABEL SETUP
1600 DIM T(2), X(29)
1610 LET Z$ = "PINK FLOYD"
1620 FOR J = 0 TO 2
1630   FOR j = 1 TO 10
1640     LET X(J * 10 + j - 1) = ASC(MID$(Z$,j,1))
1650   NEXT j
1660   LET T(J) = 0
1670 NEXT J
1680 RETURN
1690 LABEL END
1700 FOR d = 1 to 3
1710   FOR D = 10 TO 30
1720     BEEP D * 2 * d, D * 2, 63
1730   NEXT D
1740 NEXT d
1750 PROC TABLE
1760 GOTO 210

```

# TEN CHIP-TEASING GAMES FOR THE LYNX

```
1770 DEFPROC BEEP
1780 FOR J = 100 TO 1 STEP -2
1790 BEEP J, 100,63
1800 NEXT J
1810 ENDPROC
```

=====

## VARIABLES USED

~~~~~  
A = position of man  
B = position of man(temp.store)  
I = keyboard input  
J = position of monster 1 (Temp.store)  
K = position of monster 1  
L = 0  
M = pointer to start - screen map  
S = score  
X = position monster 2  
Y = position monster 2 (Temp.store)  
j = for / next loop  
Z\$ = "Pink Floyd"

## ARRAYS

-----  
T(2)  
T\$ = Table names

X(29) = score board names  
S\$ = name entered for table

## PROGRAM STRUCTRE

-----  
LINES USE  
-----  
150 - 180 sets up character set in memory  
190 - 200 set up position of monsters  
240 - 590 main program loop  
600 - 880 board set up & draw  
890 - 980 keyboard input  
990 - 1250 define char. and set up  
1260 - 1580 score table  
1590 - 1680 score table set up  
1690 - 1810 ending sounds

=====

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

LYNX LOGO

Lynx Logo is a simple graphics program. It is simplicity itself to draw any shape you require using easy to use instructions.

NOTE - ALWAYS PUT A SPACE BETWEEN COMMAND AND NAME OR NUMBER.

## - COMMANDS -

| COMMAND | ACTION   |
|---------|--|
| LI      | List a Procedure   |
| DLIST   | List of Procedures                                       |
| CL      | Clears Screen and Sets to Start Position                 |
| SA      | Save Program   |
| FO n    | Draws a Line Forwards ( n = Distance to Move )           |
| BW n    | Draws a line Backwards( n = " " " " )                    |
| RT n    | Turns Right ( n = number of Degrees )                    |
| LT n    | Turns Left ( n = " " " " )                               |
| RESET   | Run Program i.e. Clear Everything                        |
| RP      | Repeat Action ( note DO NOT REPEAT WITHIN A REPEAT )     |
| DF nnnn | Define Procedure must be a name of Four Characters.      |
| EN      | End a Procedure  |
| UN n    | Until n Finish a Repeat ( n = number of times repeated ) |
| INK n   | Change Ink Colour to n                                   |

## How to Define a Procedure

To define a procedure called Fred  
Procedure names must only be 4 characters long.

First type DF FRED The cursor will go green.

Type in the following instructions:-  
FD 100 (Forward 100 units)  
RT 90 (Right Turn 90 degrees)  
FD 100 (Forward 100 units)

End the procedure with EN

Now to run the procedure enter the name Fred.

### How to draw a square .

|        |                         |
|--------|-------------------------|
| DF SQR | - define procedure      |
| RP     | - repeat                |
| FD 50  | - Forward 50            |
| RT 90  | - Right Turn 90 degrees |
| UN 4   | - Until 4 times         |
| EN     | - End procedure         |

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

90 REM **** LYNX LOGO BY S J SAWYER 1985 ****
100 DIM D$(6)(200), E$(10)(20), A(10)
120 LET W = 1, X = 1, V = 10, U = 0, I = 4, S = 0
130 PROC CLS
140 GOTO 240
160 DEFPROC CLS
170 CLS
180 LET A = 100, B = 100, D = 0
190 MOVE A, B
200 PROC MOVE(D, O, A, B)
210 LET A = C, B = G
220 PROC ARROW(D)
230 ENDPROC
240 WINDOW 3,123,230,250
250 PRINT @ 3,230;CHR$(30);
260 INK 7
270 INPUT A$
280 PROC LIST
285 IF X > 200 THEN GOTO LABEL END
290 GOTO 250
300 DEFPROC LIST
310 IF LEFT$(A$,2) = "LI" THEN PROC LI
320 IF LEFT$(A$,5) = "DLIST" THEN PROC DLIST
330 IF LEFT$(A$,2) = "CL" THEN PROC CLS
340 IF LEFT$(A$,2) = "SA" THEN PROC SAVE
350 IF LEFT$(A$,2) = "FD" THEN PROC FORWARD
360 IF LEFT$(A$,2) = "BW" THEN PROC BACKWARD
370 IF LEFT$(A$,2) = "RT" THEN PROC RIGHT
380 IF LEFT$(A$,2) = "LT" THEN PROC LEFT
390 IF LEFT$(A$,2) = "DF" THEN PROC DF
400 IF LEFT$(A$,5) = "RESET" THEN RUN
410 IF LEFT$(A$,3) = "INK" THEN PROC INK
420 IF LEFT$(A$,2) = "RP" THEN PROC REPEAT
430 IF W = 1 THEN GOTO 470
440 FOR w = 1 TO W - 1
450 IF LEFT$(A$,4) = LEFT$(E$(w),4) THEN PROC EXECUTE
460 NEXT w
470 ENDPROC
480 DEFPROC DF
490 IF LEN(A$) < > 7 THEN ENDPROC
500 LET E$(W) = RIGHT$(A$,LEN(A$) - 3)
510 LET E$(W) = E$(W) + STR$(X)
520 IF LEN(E$(W)) < 7 THEN LET E$(W) = E$(W) + " "
530 ELSE GOTO 550
540 GOTO 520
550 IF W < 20 THEN LET W = W + 1
560 ELSE GOTO LABEL END
570 REPEAT
580 PRINT @ 3,230;CHR$(30);
590 INK 4
600 INPUT A$
610 LET D$(X) = A$
620 IF A$ = "RP" THEN PROC DF REPEAT
630 IF LEFT$(D$(U), 2) = "UN" THEN GOTO 650
640 PROC LIST
650 LET X = X + 1
660 UNTIL D$(X - 1) = "EN" OR X > 200
670 LET E$(W - 1) = E$(W - 1) + STR$(X - 1)
680 ENDPROC

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

690 DEFFPROC EXECUTE
700 LET B$ = E$(w)
710 PRINT @ 50, 0;"      "; @ 50, 0;LEFT$(B$,4);
720 LET U = VAL(MID$(B$,5,3))
730 REPEAT
740 LET A$ = D$(U)
750 LET U = U + 1
760 PRINT @ 50,240;CHR$(30);A$;
770 IF LEN(A$) = 4 THEN GOTO 860
780 VDU 7
790 IF A$ = "RP" THEN PROC EX REPEAT
800 IF LEFT$(D$(U),2) = "UN" THEN GOTO 1970
810 PROC LIST
820 UNTIL D$(U - 1) = "EN" OR D$(U) = "EN"
830 IF v < 10 THEN GOTO 920
840 LET A$ = " "
850 ENDPROC
860 LET A(v) = U, v = v - 1
870 FOR x = 1 TO w
880   IF LEFT$(E$(x),4) = A$ THEN LET B$ = E$(x)
890 NEXT x
900 UNTIL B$ = B$
910 GOTO 710
920 LET v = v + 1, U = A(v)
930 GOTO 730
940 END
950 DEFPROC ARROW(d)
960 INK 2
970 PRINT @ 0;0;"      "; @ 0,10;"      ";
980 FOR e = d - 180 TO d STEP 180
990 MOVE 10,10
1000 PROC MOVE(e,7,10,10)
1010 NEXT e
1020 INK 5
1030 LET a = c, b = g
1040 PROC MOVE((d + 147) MOD 360,5,a,b)
1050 MOVE a, b
1060 PROC MOVE(d - 147, 5, a, b)
1070 ENDPROC
1080 END
1090 DEFPROC MOVE(R, f, c, g)
1100 LET E = (ABS(360 - R) + 180) MOD 360
1110 LET c = c + f * SIN(RAD(E))
1120 LET g = g + f * COS(RAD (E))
1130 DRAW c, g
1140 ENDPROC
1150 DEFPROC BACKWARD
1160 INK I
1170 MOVE A, B
1180 LET B$ = RIGHT$(A$,LEN(A$) - 3)
1190 PROC MOVE(D - 180, VAL(B$),A, B)
1200 LET A = c, B = g
1210 ENDPROC
1220 DEFPROC FORWARD
1230 MOVE A, B
1240 INK I
1250 LET B$ = RIGHT$(A$,LEN(A$) - 3)
1260 PROC MOVE(D, VAL(B$), A, B)
1270 LET A = c, B = g
1280 ENDPROC

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

1290 DEFFPROC RIGHT
1300 LET B$ = RIGHT$(A$, LEN(A$) - 3)
1310 LET R = VAL(B$)
1320 LET D = (D + R) MOD 360
1330 PROC ARROW(D)
1340 ENDPROC
1350 DEFFPROC LEFT
1360 LET B$ = RIGHT$(A$, LEN(A$) - 3)
1370 LET R = VAL(B$)
1380 LET D = (D - R) MOD 360
1390 PROC ARROW(D)
1400 ENDPROC
1410 DEFFPROC INK
1420 LET I = VAL(RIGHT$(A$, 1))
1430 INK I
1440 ENDPROC
1450 DEFFPROC SPEED
1460 IF LEN(A$) = 5 THEN GOTO 1490
1470 LET z = LEN(A$)
1480 LET s = VAL(RIGHT$(A$, z - 6))
1490 SPEED s
1500 PRINT @ 3, 240; "SPEED : "s;
1510 ENDPROC
1520 DEFFPROC REPEAT
1530 DIM C$(8)(10)
1540 LET x = 1
1550 IF x > 10 THEN GOTO LABEL END
1560 PRINT @ 2, 230; CHR$(30);
1570 INK 2
1580 INPUT C$(x)
1590 IF LEFT$(C$(x), 2) = "UN" AND LEN(C$(x)) > 3 THEN GOTO LABEL DO
1600 LET A$ = C$(x)
1610 PROC LIST
1620 LET x = x + 1
1630 GOTO 1550
1640 LABEL DO
1650 LET z = LEN(C$(x))
1660 LET T = VAL(RIGHT$(C$(x), z - 3))
1670 FOR Z = 1 TO T - 1
1680 FOR y = 1 TO x
1690 LET A$ = C$(y)
1700 PROC LIST
1710 NEXT y
1720 NEXT z
1730 ENDPROC
1740 DEFFPROC DF REPEAT
1750 LET X = X + 1
1760 DIM C$(8)(10)
1770 LET x = 1
1780 PRINT @ 3, 230; CHR$(30);
1790 IF x > 10 THEN GOTO LABEL END
1800 INK 2
1810 INPUT C$(x)
1820 LET D$(X) = C$(x)
1830 IF LEFT$(C$(x), 2) = "UN" AND LEN(C$(x)) > 3 THEN GOTO LABEL DO
1840 LET A$ = D$(X)
1850 PROC LIST
1860 LET X = X + 1, x = x + 1
1870 GOTO 1780

```

```

1880 DEFFPROC EX REPEAT
1890 DIM C$(8)(10)
1900 LET x = 1
1910 LET C$(x) = D$(U)
1920 IF LEFT$(C$(x), 2) = "UN" THEN GOTO LABEL DO
1930 LET A$ = C$(x)
1940 PROC LIST
1950 LET U = U + 1, x = x + 1
1960 GOTO 1910
1970 LET U = U + 1
1980 GOTO 820
1990 DEFFPROC DLIST
2000 PRINT @ 0,0;
2010 FOR Q = 1 TO W - 1
2020 PRINT TAB 34;LEFT$(E$(Q), 4)
2030 NEXT Q
2040 ENDPROC
2050 DEFFPROC LI
2060 IF LEN(A$) < > 7 THEN ENDPROC
2070 LET B$ = " "
2080 FOR Q = 1 TO W - 1
2090 IF RIGHT$(A$, 4) = LEFT$(E$(Q), 4) THEN LET B$ = E$(Q)
2100 NEXT Q
2110 IF B$ = " " THEN ENDPROC
2120 LET u = VAL(MID$(B$, 5, 3))
2130 LET r = 1
2140 PRINT @ 50,0;
2150 REPEAT
2160 PRINT TAB 30;r"; ";D$(u)
2170 LET r = r + 1, u = u + 1
2180 UNTIL D$(u - 1) = "EN"
2190 ENDPROC
2200 DEFFPROC SAVE
2210 PRINT @ 0,230;
2220 INPUT "Enter program name ($ CHARs) ";Q$
2230 IF LEN(Q$) < > 6 THEN GOTO 2220
2240 FOR q = 1 to 6
2250 POKE LCTN((2280) + q, ASC(MID$(Q$, q, 1)))
2260 NEXT q
2270 CODE EB 21 F6 61 E5 2A 1F 62 E5 21 00 00 C3 FF 3E
2280 REM "LOGO.."
2290 CALL LCTN(2270), LCTN(2280)
2300 PROC CLS
2310 GOTO 250
2320 LABEL END
2330 PRINT "PROC TABLE FULL";
2340 GOTO 250

```

-----

O T H E L L O  
~~~~~[ For the 96k Lynx ]  
(c) S J Sawyer and M R Sawyer 1985

## DESCRIPTION

The object of the game is to finish the game with as many of your pieces as possible on the board. Every move you make must lead to your opponents pieces being changed to your pieces.

```

90 REM ***** OTHELLO BY S J SAWYER 1985 *****
100 EXT VRESET
110 PROC GRAPHIC
120 CCHAR &202A
130 LET O$ = CHR$(1) + CHR$(2) + CHR$(128) + CHR$(1) + CHR$(7), X$ =
    CHR$(1) + CHR$(4) + CHR$(128) + CHR$(1) + CHR$(7)
140 LET D$ = ":" 13;< =", c = 0, h = 0, Z = 0, g = 0, b = 1, a = -1
150 GOTO LABEL INTRO
160 WINDOW 3,123,240,250
170 FOR A = 1 TO 10
180 BEEP A * 10, 100, s
190 NEXT A
200 DIM A(100), E(64), C(100)
210 LET A(0) = 0
220 FOR A = 1 TO 100
230 READ B$
240 LET A(A) = ASC(B$), C(A) =ASC(B$)
250 NEXT A
260 DATA *,1,2,3,4,5,6,7,8,*,,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,B,
    C,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,B,
    F,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,E,
    *,1,2,3,4,5,6,7,8,*
270 IF RAND(2) = 1 THEN RESTORE 450
275 ELSE RESTORE 455
280 FOR A = 1 TO 60
290 READ J
300 LET E(A) =J
310 NEXT A
320 IF g = 1 THEN PROC g
330 IF g = 1 THEN GOTO 930
340 VDU 1, 6
350 PRINT @ 40,150;"AGAINST COMPUTER"; @40,150;
    CHR$(21)"-----";CHR$(20);
360 PRINT @ 20,200;"Do you want to go first, press 'Y'; @ 20,
    210;"Otherwise press another key"
370 LET T$ = GET$
380 CLS
390 PROC B
400 PRINT @ 40,0;
410 VDU 24,1,1,79,1,2,84,1,3,72,1,4,69,1,5,76,1,6,76,1,7,79,25
420 LET B = 79
430 IF UPC$(T$) ="Y" THEN GOTO 930
440 LET B = 79, C = 88, Z = 0
450 DATA 19,89,82,12,62,17,32,87,69,14,39,84,64,37,34,67,49,15,59,
```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

85, 16, 52, 42, 86, 65, 54, 66, 44, 35, 57, 36, 47, 63, 48, 58, 76, 24, 27, 38,
25, 74, 77, 43, 26, 68, 33, 53, 75, 72, 13, 29, 18, 88, 22, 83, 79, 73, 28, 78, 23
455 DATA 12, 19, 82, 89, 67, 34, 37, 64, 84, 39, 14, 69, 87, 32, 17, 62, 86, 42, 52,
16, 85, 59, 15, 49, 35, 57, 65, 44, 36, 47, 66, 54, 32, 68, 74, 38, 24, 77, 62, 27,
43, 25, 48, 75, 53, 26, 58, 76, 72, 88, 29, 13, 83, 79, 18, 22, 23, 28, 78, 73
460 GOTO 590
470 DEFPROC BOARD
480 FOR J = 1 TO 8
490   FOR K = 2 TO 9
500     PRINT @ 30 + K * 3, 40 + J * 10;
510     IF A(10 * J + K) = 79 THEN PROC O
520     IF A(10 * J + K) = 88 THEN PROC X
530     IF A(10 * J + K) = 79 THEN LET h = h + 1
540     IF A(10 * J + K) = 88 THEN LET c = c + 1
550     BEEP 20, 2, s
560   NEXT K
570 NEXT J
580 ENDPROC
590 FOR k = 1 TO 60
600   IF Z = 1 THEN GOTO 630
610   LET M = E(k)
620   IF A(M) < > 46 THEN GOTO 780
630   LET j = 0
640   FOR X = 1 TO 8
650     LET N = ASC(MID$(D$, X, 1))
660     LET N = N - 50, E = 0, F = M
670     IF A(F + N) < > B THEN GOTO 700
680     LET E = 1, F = F + N
690   GOTO 670,
700   IF A(F + N) < > C OR E = 0 THEN GOTO 750
710   FOR Q = M TO F STEP N
720     LET A(Q) = C
730     LET j = 1
740   NEXT Q
750   NEXT X
760   IF Z = 1 THEN LET k = 60
770   IF B = 88 OR j = 1 THEN LET k = 60
780 NEXT k
790 IF Z = 1 AND j = 0 THEN GOTO 1050
800 IF G$ = "PASS" AND g = 1 THEN GOTO 1740
810 IF G$ = "PASS" AND Z = 0 AND j = 0 AND c > h THEN GOTO LABEL
I WIN
815 IF G$ = "PASS" AND c + h = 64 AND c > h THEN GOTO LABEL I WIN
820 IF G$ = "PASS" AND Z = 0 AND j = 0 AND h > c THEN GOTO LABEL
YOU WIN
825 IF G$ = "PASS" AND c + h = 64 AND h > c THEN GOTO LABEL YOU
WIN
830 IF G$ = "PASS" AND h < 1 THEN GOTO LABEL I WIN
840 IF G$ = "PASS" AND c < 1 THEN GOTO LABEL YOU WIN
850 IF G$ = "PASS" AND c = h THEN GOTO 1230
890 IF B = 79 AND g = 1 THEN GOTO 920
900 IF B = 88 AND g = 1 THEN LET B = 79, C = 88
910 GOTO 930
920 IF B = 79 AND g = 1 THEN LET B = 88, C = 79
930 LET c = 0, h = 0
940 PROC BOARD
950 VDU 1, WHITE, 25
960 IF g = 1 THEN GOTO 1660
970 PRINT @ 20, 220; "YOU HAVE (";0$;"") : "; "h;" "; @ 20, 230;
" I HAVE (";X$;"") : "; "c;" ";
980 IF B = 88 THEN GOTO 440

```

```

990 LET B = 88, C = 79
1000 REM
1020 PROC END CHECK
1030 IF c + h = 64 THEN LET A = 1
1040 IF A = 1 THEN GOTO 1090
1050 PRINT @ 10, 240;A$;
1060 INPUT "Your move (ie D4)";G$
1070 PRINT @ 10, 240;""
1080 PROC END CHECK
1090 IF A = 1 THEN LET G$ = "PASS"
1100 IF G$ = "PASS" THEN GOTO 440
1110 IF LEN(G$) < > 2 OR ASC(LEFT$(G$,1)) < 65 OR
     ASC(LEFT$(G$,1)) > 72 OR ASC(LEFT$(G$,1)) > 56 OR
     ASC(RIGHT$(G$,1)) < 49 THEN GOTO 1000
1120 LET M = 10 * (ASC(G$) - 64) + (ASC(RIGHT$(G$,1))) - 47
1130 IF A(M) < > 46 THEN GOTO 1000
1140 BEEP 200, 100, s
1150 LET Z = 1
1160 GOTO 590
1170 INK 3
1180 PRINT @ 10, 240;"Press Y to play again";
1190 LET Y$ = GET$
1200 IF UPC$(Y$) = "Y" THEN RUN
1210 EXT VRESET
1220 END
1230 INK 5
1240 PROC CLEAR
1250 PRINT @ 40, 220;"CHEAT"; @ 40, 230;"I WIN";
1260 BEEP 500, 200, s
1270 GOTO 1170
1280 PROC CLEAR
1290 LABEL INTRO
1300 VDU 24, 1, 5, 4
1304 FOR Y = 1 TO 4
1305   FOR X = 1 TO 20 STEP 2
1306     BEEP X * 10, Y * 5, 63
1307   NEXT X
1308 NEXT Y
1310 PRINT @ 50, 0;"OTHELLO"
1320 VDU 25, 1, 6
1330 INK RED
1340 EXT CIRCLE 1, 15, 15, 5
1350 INK GREEN
1360 EXT CIRCLE 1, 235, 15, 5
1370 PRINT @ 50,8;
1380 VDU 21, 6, 95, 95, 95, 95, 95, 95, 95, 20, 1, 2
1390 PRINT @ 3,25;"The object of the game is to have more of your
     pieces on the board at the end of the game , than your
     opponent"
1400 PRINT @ 3, 60;"Press PASS to leave game and when you and
     your opponent can't move."; @ 10, 230;CHR$(1) + CHR$(5) t"
     (c) By S.J.Sawyer and M.R. Sawyer"
1410 INK GREEN
1420 PRINT @ 10,90;"Do you want sound [ Y OR N ]?";
1430 LET Z$ = GET$
1440 BEEP 50, 50, 63
1450 IF UPC$(Z$) = "Y" THEN LET s = 63
1460 IF UPC$(Z$) = "N" THEN LET s = 0
1470 IF NOT UPC$(Z$) = "N" AND NOT UPC$(Z$) = "Y" THEN GOTO 1430
1480 BEEP 100, 50, s
1490 PRINT @ 0, 110;;

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

1500 PRINT @ 5, 90;"Input the type of game you want to play : "
1510 INK YELLOW
1520 PRINT @ 10, 110;"1. Against the computer"; @ 10, 120;"2. For
two players"; @ 40, 140;"Enter choice [ 1 OR 2]?";
1530 PRINT @ 80, 130;
1540 LET z = GETn - 48
1550 IF z < 1 OR z > 2 THEN GOTO 1540
1560 BEEP 100, 200, s
1570 PRINT @ 40, 140;""
1580 IF z = 1 THEN GOTO 160
1590 PRINT @ 3, 150;
1600 INPUT "First player enter your name"; I$
1605 LET I$ = LEFT$(I$,8)
1610 BEEP 100, 200, s
1620 INPUT "Second player enter your name"; H$
1625 LET H$ = LEFT$(H$,8)
1630 BEEP 100, 200, s
1640 LET g = 1, B = 79, C = 88
1650 GOTO 160
1660 PRINT @ 10, 220;I$;" (";0$;") HAS ";h
1670 PRINT @ 10, 230;H$;" (";X$;") HAS ";c
1680 IF B = 88 THEN LET A$ = I$
1690 IF B = 79 THEN LET A$ = H$
1700 FOR x = 30 TO 5 STEP -1
1710 BEEP x * 2, B, s
1720 NEXT x
1730 GOTO 1000
1740 PROC CLEAR
1750 IF c + h < 60 AND c > 0 AND h > 0 OR c = h THEN PRINT @ 40,
220;"VOID"; @ 40, 230;"GAME";
1760 IF c + h < 60 AND c > 0 AND h > 0 OR c = h THEN GOTO 1790
1770 IF c < h THEN GOTO LABEL YOU WIN
1780 ELSE GOTO LABEL I WIN
1790 FOR A = 50 TO 0 STEP -1
1800 BEEP A * 2, 100, s
1810 NEXT A
1820 GOTO 1170
1830 LABEL I WIN
1840 PROC CLEAR
1850 FOR A = 1 TO 28
1860 INK A
1870 PRINT @ 40, 220;"I WIN!";
1880 BEEP A * 50, 1, s
1890 BEEP 1400 - (A * 50), 1,s
1900 NEXT A
1910 GOTO 1170
1920 DEFFPROC O
1930 INK RED
1940 EXCT CIRCLE 1,10 +(K * 20), ((J + 2) * 20) -10,5
1950 BEEP 50, 10, S
1960 ENDPROC
1970 DEFFPROC X
1980 INK GREEN
1990 EXT CIRCLE 1, 10 + (K * 20), ((J + 2) * 20) -10, 5
2000 BEEP 100, 2, s
2010 ENDPROC
2020 DEFFPROC CLEAR
2030 PRINT @ 0, 220;""
2040 ENDPROC
2050 DEFFPROC END CHECK
2060 LET A = 0, a = 0, b = 0

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

2070 FOR J = 1 TO 8
2080   FOR K = 2 TO 9
2090     IF A(10 * J + K) = 79 THEN LET a = 1
2100     IF A(10 * J + K) = 88 THEN LET b = 1
2110   NEXT K
2120 NEXT J
2130 IF a < > 1 OR b < > 1 THEN LET A = 1
2140 ENDPROC
2150 LABEL YOU WIN
2160 FOR A = 500 TO 1 STEP -1
2170   BEEP A, 1, s
2180 NEXT A
2190 PROC CLEAR
2200 LET G$ = ""
2210 IF c > h AND g = 1 THEN LET G$ = H$
2220 IF h > c AND g = 1 THEN LET G$ = I$
2230 PAPER BLUE
2240 FOR A = 1 TO 28
2250   BEEP A * 25, 2, s
2260   BEEP 1400 - (A * 25), 2, s
2270   INK A
2280   PRINT 2 40, 220;G$" * YOU WIN * ";
2290 NEXT A
2300 PAPER BLACK
2310 GOTO 1170
2320 DEFFPROC B
2330 INK WHITE
2340 FOR A = 1 TO 9
2350   MOVE 20 + (A * 20), 40
2360   DRAW 20 + (A * 20), 200
2370 NEXT A
2380 FOR A = 1 TO 9
2390   MOVE 40, (A * 20) + 20
2400   DRAW 200, (A * 20) + 20
2410 NEXT A
2420 VDU 24, 1, 6
2430 FOR A + 1 TO 8
2440   RINT @ 13, (A * 10) + 10;CHR$(A + 64); @ 102,(A * 10) +
      10;CHR$(A + 64); @ (A * 10) + 13, 10;CHR$(48 + A); @ (A *
      10) + 13, 101;CHR$(48 + A);
2450 NEXT A
2460 INK RED
2470 EXT CIRCLE 1, 34, 30, 7
2480 EXT CIRCLE 1, 205, 30, 7
2490 INK GREEN
2500 EXT CIRCLE 1, 34, 210, 6
2510 EXT CIRCLE 1, 205, 210, 6
2520 VDU 25
2530 ENDPROC
2540 DEFFPROC g
2550 CLS
2560 PRINT @ 40, 0;
2570 VDU 24, 1, 1, 79, 1, 2, 84, 1, 3, 72, 1, 4, 69, 1, 5, 76, 1,
      6, 76, 1, 7, 79, 25,
2580 PROC B
2590 ENDPROC
2600 DEFPROC GRAPHIC
2610 CODE 00 00 1E 3F 3F 3F 3F 3F 1E 00
2620 DPOKE GRAPHIC, LCTN(2610)
2630 ENDPROC

```

## VARIABLES USED

~~~~~

|                           |                          |
|---------------------------|--------------------------|
| B = player piece pointer  | j = move indicator       |
| C = player piece pointer  | k = main loop            |
| Z = player move marker    | s = beep volume          |
| c = computer piece count  | z = 1 : against computer |
| g = 1 = 2 player game     | 2 : 2 player game        |
| 2 = play against computer |                          |
| h = human piece count     |                          |

## STRING VARIABLES

-----

|                         |                     |
|-------------------------|---------------------|
| O\$ = players O piece   | I\$ = player 1 name |
| X\$ = players X piece   | H\$ = player 2 name |
| D\$ = direction pointer |                     |

## ARRAYS

-----

|                        |
|------------------------|
| A(100) pieces on board |
| E(60) computer moves   |
| C(100) pieces          |

## PROGRAM STRUCTURE

-----

|             |   |
|-------------|---|
| 130 - 140   | Set up start variables                        |
| 160 - 310   | Initialise array for board                    |
| 340 - 460   | Against computer ?                            |
| 470 - 580   | Print pieces on board                         |
| 540 - 780   | Set up pieces on board                        |
| 800 - 900   | Check for pass                                |
| 1000 - 1150 | Input entry (1 player) & check if legal move. |
| 1170 - 1220 | End of game                                   |
| 1230 - 1260 | Check   |
| 1290 - 1570 | Instructions                                  |
| 1590 - 1650 | Input both players names                      |
| 1610 - 1730 | Print both players names                      |
| 1740 - 1820 | Check if you or computer wins                 |
| 1830 - 1410 | I win   |
| 1920 - 1960 | First players piece                           |
| 1970 - 2010 | Second players piece (or computers)           |
| 2020 - 2040 | Clear   |
| 2050 - 2140 | Check for end of game                         |
| 2130 - 2310 | You win                                       |
| 2310 - 2530 | Print board outline                           |
| 2600 - 2630 | Graphics set up                               |

=====

## H A N G M A N

~~~~~

(c) M R Sawyer 1985

For the 96k Lynx

## DESCRIPTION

-----  
 This version of hangman is for one or two players. The object of the game is to guess the word as quickly as possible. If you are too slow or your opponent beats you, you will be hanged. Both players have the same length of word.

```

1 REM ***** HANGMAN BY M R SAWYER *****
2 EXT VRESET
5 RANDEOM
10 WINDOW 3,123,5,245
20 CODE IF 20 29 20 29 2F 20 1F 00 00 20 10 10 10 10 10 10 10 20 00
   00 1F 20 29 20 2F 29 20 1F 00 00 20 10 10 10 10 10 10 10 20 00:00
30 DPOKE GRAPHIC,LCTN(20)
40 CCHAR &2023
100 CLS
105 PROTECT 0
110 LET X = 0, Z = 0, O = 0, z = 0, Q = 0, v = 0, W = 0, G = 0
120 VDU 7
130 PROC START
135 WINDOW 0,123,230,240
140 PROC WORD
150 VDU 7
155 IF H = 2 THEN GOTO LABEL ONE PLAYER
160 PROC SETUP
170 VDU 7,1,7
180 PROC PLAYER ONE
190 PROC PLAYER TWO
200 GOTO 180
210 DEFFPROC WORD
220 RESTORE
225 RANDOM
230 LET B = INT(RND *351) + 1
250 FOR A = 1 TO B
260   READ A$
270 NEXT A
280 LET C = LEN(A$)
290 LET B + INT(RND * 351) + 1
310 RESTORE
320 FOR A = 0 TO B
330   READ B$
340 NEXT A
350 IF LEN(B$) < > C THEN GOTO 290
360 ENDPROC
370 DATA ADDRESS,BASIC,COMMAND,JOYSTICK,KEYBOARD,MACHINE,MEMORY,
  PROCESSOR,STRING,STATEMENT,LANGUAGE,COMPUTER,HOUSE,WORDS,
  ENVELOPES,ASSORTED,CHARTS
375 DATA YEAST,YARBOROUGH,YACHT,XYLOPHONE,XEROGRAPHY,PHOTOCOPY,
  YEARN,YELLOW,YOGHURT,YORKSHIRE,PUDGING,ZOOGRAPHY,ZOOLOGY,
  INORGANIC,OPPORTUNE,INOPERABLE,OFFENSIVE,INHIBIT,INHERIT,
  INGREDIENT,FAMOUS,FAHRENHEIT,DILUTE

```

377 DATA CAPILLARY, CAPITAL, CANNON, CAMPUS, CANDIDATE, CAPACITY,  
 GLUCOSE, GLITTER, GOGGLE, JITTERBUG, LUXURY, LUKEWARM, LUBRICATE,  
 MACHEREL, MEASURE, PATHOLOGY, PATROL, PAVILION, PNEUMONIA, POLICE,  
 POWDER, PRESCRIBE, PROPULSION, QUARTZ, QUARTER, QUARANTINE, QUEEN  
 380 DATA NETWORK, ANIMALS, PROGRAM, BOARDS, SWITCH, PRICE, SPEED,  
 DISPLAY, MAGIC, FLIGHT, MONDAY, TUESDAY, WEDNESDAY, FRIDAY,  
 SATURDAY, SOUND, LIGHT, BRIGHTER, NUMBER, FLAVOURS, PAPER, CONTENT,  
 MATHS, TAPES, CLOCK, MOUSE, SCHOOL, WORKER, VOLUME, PRINT, WINDOW,  
 ROYAL  
 385 DATA DIFFUSE, DINNER, CAT, TROLLY, DIOXIDE, DIVIDE, FLATTER, FLIRT,  
 AUTOMATIC, TURNTABLE, COMMUNICATE, COMPASS, EXCHANGE, GATHER,  
 GUIDE, PRODUCE, PROFOUND, PROLONG, RAZOR, SHOWER, SHOULDER, SHOVEL,  
 SHUFFLE, SHUTTLE, SKEWER, TENT, SKIPPER, SLAUGHTER  
 390 DATA CONTROL, SCRREN, PAPER, SUMMARY, MUSIC, BIRDS, HOUSEHOLD, RACER,  
 FIRST, BOATS, DOORS, COURSE, TRAINS, STEERING, HELPER, HARDWARE,  
 FILLING, LOADING, VIDEO, ROUTINE, DATES, MONTHS, BREAKFAST, CREAM,  
 BRICKS, DEADLY, FOOTBALL, EXPRESS, MORTAR, FORKLIFT, PILGRIM  
 392 DATA AFTERWARDS, AFRICA, AFFILATE, CLASSIC, CLEMENT, DODGE, EXPLODE,  
 EXTEND, EXTERIOR, FILIBUSTER, FINGER, HITCH, HOLOGRAPH, HOLSTER,  
 HOLLAND, HOGMANAY, HOLIDAY, IBERIAN, IDENTITY, IGNORE, ILLEGAL,  
 IMPLODE, WARDEN, WARNING, WATCH, WATERGLASS, SKELETON, SKETCH  
 395 DATA PARAMOUNT, PARADOX, PARAFFIN, PARALLEL, PARALLAX, PARISH,  
 PARITY, PARTNER, PARTRIDGE, PARTY, PASSAGE, PASSENGER, PASSION,  
 PASSPORT, PROPOSAL, ACCESSORY, ACCIDENT, ACCOMPANY, ABSLUTE,  
 ABNORMAL, AERIAL, ADVOCATE, ADVERTISE, AESTHETIC  
 396 DATA ENGLISH, FRENCH, SPANNISH, GERMAN, BALANCE, RECEIVER, VOLUME,  
 EQUALISER, STERO, ISOBAR, PAUSE, ELECTRONIC, FUNCTION, IVORY,  
 JACKET, INTERCOM, INTEREST, DIESEL, INTERLEAVE, MONITOR, MOMENT,  
 MONSOON, MONKEY, MONEY, MONUMENT, MOLECULAR, MANOEUVRE  
 397 DATA MANUSCRIPT, BLOUSE, BLANKET, BLACKMAIL, BLIZZARD, BLOCKADE,  
 BLOOD, BLIND, BEAUTY, BATTLEMENT, BARRISTER, BUTTERFLY, BULKHEAD,  
 BULLDOZER, COURTESY, COURAGE, CRANBERRY, CREMATION, CROCODILE,  
 CROQUETTE, CRITERION, CUMULATIVE, DELIGHT, DEMOCRACY, DIELECTRIC  
 398 DATA DYNAMIC, DYNAMITE, HAMBURGER, NICKEL, HEMISPHERE, NUCLEUS,  
 TELESCOPE, TELEPHONE, NUMERAL, VESSEL, VETERINARY, VERMOUTH,  
 UNTOWARD, UNION, UNITY, UNIVERSITY, UNIVERSE, UNIQUE, PUNCTUAL,  
 PUBLIC, PROMISE, PRECISE, PLANET, EARTH, PERPLEX, OLIVE,  
 OINTMENT, OBSOLETE  
 399 DATA LOTTERY, LONGITUDE, LOOPHOLE, LITMUS, LIQUORICE, LINOTYPE,  
 LINGERIE, LAUGH, LABORATORY, KNUCKLE, KANGAROO, JUKEBOX, JUNCTION,  
 JUGGERNAUT, JOURNAL, JANUARY, JACKPOT, ENLARGE, ENDEAVOUR, ENDORSE,  
 ENERGY, EMERALD, ELIMINATE, ELECTRON, ELEPHANT  
 400 DAT COLOUR, PLANT, TREES, TRANSPORT, SYSTEM, NORTH, POWER, PLUG,  
 BATTLES, LIVING, REGIONS, WORLD, FARMING, FOODS, ALARM, HEATER,  
 ELECTRICITY, LANDING, MONEYBOX, PLASTICS, RADIO, TELEVISION,  
 PEOPLE, SHIPS, SOUTH, WEATHER, CLIMATE, BODYGUARD, METALS, STARS,  
 STUNTMAN  
 410 DEFPROC SETUP  
 420 CLS  
 430 MOVE 123,0  
 440 INK 7  
 450 DRAW 123,130  
 460 MOVE 0,130  
 470 DRAW 255,130  
 480 VDU 1,4,2,2  
 490 PRINT @ 50,0; "HANGMAN";  
 495 VDU 1,5,2,0  
 500 INK 5  
 505 PRINT @80,0; CHR\$(94);CHR\$(127); "By M. Sawyer";  
 510 LET Y = 0  
 520 FOR Q = 1 TO 2

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

530 FOR A = 1 TO C
540 LET V = Y
550 MOVE A * 10 + V,200
560 DRAW V + A * 10 + B,200
570 NEXT A
580 LET Y = 140
590 NEXT B
592 PRINT @ 3,230;
600 PRINT "Player one enter your name ";
610 INPUT Z$
615 LET Z$ = LEFT$(Z$,8)
620 PRINT @ 10,220;""
630 INK 2
635 PRINT @ 3,230;
640 PRINT "Player two enter your name ";
650 INPUT X$
655 LET X$ = LEFT$(X$,8)
660 INK MAGENTA
670 PRINT @ 10,230;""
680 PRINT @ 10,140;Z$;
690 PRINT @ 90,140;X$;
700 LET Q = 0
710 ENDPROC
720 DEFFPROC PLAYER ONE
730 INK 5
740 PRINT @ 13,230;CHR$(30);Z$;"Enter letter ";
750 INPUT D$
755 LET D$ = UPC$(D$)
760 LET F = 1
765 PRINT @ 0,230;CHR$(30);
770 IF LEN(D$) > 1 THEN IF D$ = A$ AND H = 2 THEN GOTO LABEL YOU
    WIN
770.22 IF LEN(D$) > 1 THEN IF D$ = A$ THEN GOTO LABEL A$ WINS
780 LET A = 0
790 REPEAT
800 LET A = A + 1
810 IF D$ = MID$(M$,A,1) THEN GOTO LABEL AGAIN 2
820 UNTIL A = C
830 FOR A = 1 TO C
840 IF D$ = MID$(A$,A,1) THEN PROC PRINT 1
850 NEXT A
850.2 IF Q = C AND H = 2 THEN GOTO LABEL YOU WIN
850.5 IF Q = C THEN GOTO LABEL A$ WINS
851 IF W = 14 AND H = 2 THEN GOTO LABEL LOSE
855 IF W = 14 THEN GOTO LABEL B$ WINS
856 IF H = 2 THEN LET V = 70
857 ELSE LET V = 0
860 IF X = 0 THEN PROC HANGMANM(W,V)
870 ELSE GOTO 890
880 LET W = W + 1
890 LET X = 0
895 IF Q = C AND H = 2 THEN GOTO LABEL YOU WIN
900 IF Q = C THEN GOTO LABEL A$ WINS
902 IF W = 14 AND H = 2 THEN GOTO LABEL LOSE
905 ELSE IF W = 14 THEN GOTO LABEL B$ WINS
910 ENDPROC
920 LABEL W
930 LABEL A$ WINS
932 FOR A = 1 TO C
933 LET D$ = MID$(A$,A,1)
934 PROC PRINT 1

```

```

935 NEXT A
936 IF G = 14 THEN GOTO 947.2
940 FOR K = G TO 13
945 PROC HANGMAN(K,123)
947 NEXT K
947.2 INK 6
947.5 PRINT @ 70,202;B$;
947.6 INK RED
948 PRINT @ 30,210;" ";Z$;" WINS ";
948.2 FOR A = 100 TO 200
948.21 BEEP A,1,63
948.22 NEXT A
948.3 GOTO 4275
950 LABEL B$ WINS
951 FOR A = 1 TO C
952 LET E$ = MID$(B$,A,1)
953 PROC PRINT 2
954 NEXT A
954.5 IF W = 14 THEN GOTO 965.2
955 FOR K = W TO 13
960 PROC HANGMAN(K,0)
965 NEXT K
965.2 INK 6
965.5 PRINT @ 5,202;A$;
965.6 INK RED
965.7 FOR A = 100 TO 200
965.8 BEEP A,1,63
965.9 NEXT A
966 PRINT @ 30,210;" ";X$;" WINS ";
969 GOTO 4275
970 END
980 NEXT K
990 DEFPROC PRINT 1
991 IF H = 2 THEN LET V = 10, h = 15
992 ELSE LET V = 0, h = 0
995 BEEP 100,100,63
997 FOR x = 1 to 7
998 INK x
1000 PRINT @ A * 5 + h,190 + V;D$
1005 NEXT x
1010 LET X = 1
1020 LET Q = Q + 1
1030 LET M$ = M$ + D$
1040 ENDPROC
1050 DEFPROC PLAYER TWO
1060 INK 2
1070 PRINT @ 0,230;;
1080 PRINT @ 10,230;X$;" Enter letter ";
1090 INPUT E$
1095 LET E$ = UPC$(E$)
1100 LET P = 2
1110 LET A = 0
1120 REPEAT
1130 LET A = A + 1
1140 IF E$ = MID$(N$,A,1) THEN GOTO LABEL AGAIN 1
1150 UNTIL A = C
1160 IF LEN (E$) > 1 THEN IF E$ = B$ THEN GOTO LABEL B$ WINS
1170 FOR A = 1 TO C
1180 IF E$ = MID$(B$,A,1) THEN PROC PRINT 2
1190 NEXT A
1192 IF O = C THEN GOTO LABEL B$ WINS

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

1195 IF G = 14 THEN GOTO LABEL A$ WINS
1200 IF X = 0 THEN PROC HANGMAN(G,123)
1210 ELSE GOTO 1230
1220 LET G = G + 1
1230 LET X = 0
1240 IF O = C THEN GOTO LABEL B$ WINS
1245 IF G = 14 THEN GOTO LABEL A$ WINS
1250 ENDPROC
1260 DEFFPROC PRINT 2
1265 BEEP 100,100,63
1270 LET X = 1
1275 FOR x = 1 to 7
1277 INK x
1280 PRINT @ 70 + A * 5, 190;E$;
1285 NEXT x
1290 LET O = O + 1
1300 LET N$ = N$ + E$
1310 ENDPROC
1360 DEFROPC START
1370 VDU 1,5,24,4
1380 PRINT @ 40,0;"COMPUTER HANGMAN"
1390 INK 2
1400 MOVE 80,17
1410 DRAW 175,17
1420 VDU 25,1,4
1430 PRTECT 3
1440 PRINT @ 10,50;"This is the game of computer hangman"
1450 PRINT @ 10,60;"The game is for one or two players."
1460 PRINT @ 10,80;"You can enter a word or a letter"
1470 PRINT @ 10,90;"Both players will get the same length"
1480 PRINT @ 10,100;"of word."
1500 PROTECT 0
1510 INK 5
1515 PROTEXT 0
1520 PRINT @ 20,230;CHR$(94);CHR$(127);;"By Malcolm Sawyer";
1532 INK WHITE
1533 PRINT @ 30,125;"Number of players[1 or 2]?";
1533.2 RESTORE 9400
1533.3 LET A = 1
1533.5 REPEAT
1533.6 LET A = A + 1
1533.7 READ B,C
1533.8 BEEP B,C,63
1533.85 LET H = KEYN -48
1533.9 IF A = 13 THEN RESTORE 9400
1533.95 IF A = 13 THEN LET A = 1
1534 UNTIL H = 1 OR H = 2
1535 IF H < 1 OR H > 2 THEN GOTO 1533.2
1536 IF H = 2 THEN LET H = 1
1537 ELSE IF H = 1 THEN LET H = 2
1550 ENDPROC
1560 LABEL AGAIN 1
1570 INK 3
1580 PRINT @ 0,220;;
1590 PRINT @ 220;"ALREADY HAD THAT LETTER";CHR$(30)
1600 BEEP 500,500,63
1610 PRINT @ 0,220;;
1620 GOTO 190
1630 LABEL AGAIN 2
1640 INK 2
1650 PRINT @ 0,220;;

```

## TEN CHIP-TEASING PROGRAMS FOR THE LYNX

```

1660 PRINT @ 0,220;"ALREADY ENTERED THAT LETTER";CHR$(30)
1670 BEEP 500,500,63
1680 PRINT @ 0,220;"
1685 IF H = 2 THEN GOTO 3076
1690 GOTO 180
1700 DEFFPROC HANGMAN(a,b)
1701 IF H = 2 THEN LET V = V - 20
1702 BEEP 500,100,63
1705 IF a = 7 THEN GOTO LABEL HEAD
1706 IF a = 2 OR a = 13 THEN GOTO LABEL BOX
1710 RESTORE a + 1790
1720 READ c,f,g
1730 INK c
1740 MOVE f + b, g + v
1750 FOR J = 1 TO 3
1760   READ f,g
1770   DRAW f + b, g + v
1780 NEXT J
1790 DATA 6,0,130,10,120,40,120,40,130
1791 DATA 6,40,120,50,120,70,120,80,130
1793 DATA 5,25,120,25,40,26,120,26,40
1794 DATA 5,25,40,60,40,25,41,60,41
1795 DATA 2,26,60,40,41,26,61,41,41
1796 DATA 7.55,41,55,50,55,50,55,50
1797 DATA 5,55,58,55,79,54,79,54,58,54,58
1799 DATA 3,55,65,45,60,45,50,45,50
1800 DATA 3,56,65,65,60,67,50,67,50
1801 DATA 4,54,79,49,85,49,97,47,97
1802 DATA 4,56,80,61,85,61,97,63,97
1840 ENDPROC
1930 LABEL HEAD
1931 INK 6
1932 IF b = 123 THEN PRINT @ 87,50;
1933 ELSE PRINT @ 25,50;
1933.5 IF H = 2 THEN PRINT @ 25 + 35,100;
1934 IF a = 7 THEN VDU 128,129,31
1936 IF a < > 7 THEN VDU 130,131
1950 GOTO 1840
3000 LABEL BOX
3005 IF a = 2 then ink 2
3006 ELSE INK 0
3020 MOVE 44 + b,119
3030 FOR J = 44 TO 64 STEP 4
3032   MOVE J + b,119 + v
3035   DRAW J + b,100 + v
3040 NEXT J
3050 FOR J = 98 TO 100
3055   MOVE 44 + b, J + v
3056   DRAW 64 + b,J + v
3057 NEXT J
3058 IF a < > 2 THEN GOTO LABEL HEAD
3060 GOTO 1840
3070 LABEL ONE PLAYER
3071PROC WORD
3072 PROC SETUP1
3076 PROC PLAYER ONE
3080 GOTO 3076
3999 LET V = GETN
3999.5 PRINT @ 0,0;
4000 DEFFPROC SETUP1
4005 WINDOW 0,123,240,250

```