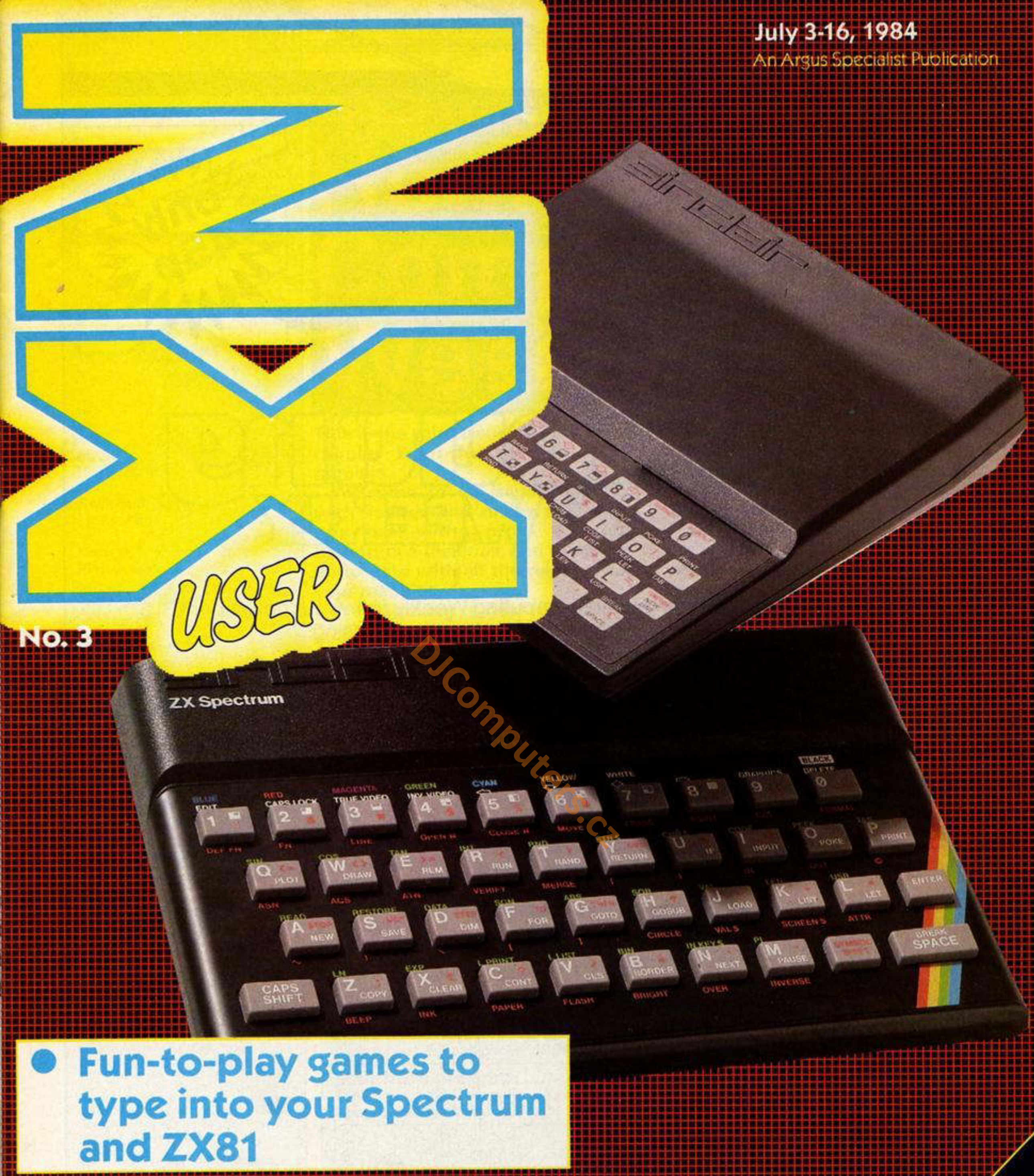
ZX USER

July 3-16, 1984
An Argus Specialist Publication

No. 3



- **Fun-to-play games to type into your Spectrum and ZX81**
- Software reviews: read what our experts have to say
- Test your brain in our logic game for the Spectrum

Make sure of your

only 45p

Home Computing WEEKLY

Unless you order Home Computing Weekly from your newsagent, this is what you're in danger of missing:

- ★ Latest software reviews — our experts tell you what to watch for . . . and what to avoid
- ★ Quality-tested listings for all popular home computers — games utilities and educational
- ★ Up-to-date news of the micro scene
- ★ Software charts — see what's selling best for your computer
- ★ Free supplements on things you need to know
- ★ U.S. Scene — our man in California brings you regular reports
- ★ Free-to-enter competitions. Our readers have won prizes worth tens of thousands of pounds
- ★ The liveliest letters page around
- ★ Spot the bargains on our classified pages

You can't afford to be without Home Computing Weekly. And all you have to do to make sure you get your copy is to fill in the form below and hand it to your newsagent.

Dear newsagent

Please deliver/reserve for me a copy of Home Computing Weekly every week.

Name .....

Address .....

.....

.....

Newsagent: trade distribution of Home Computing Weekly is by Argus Press Sales and Distribution, 12-14 Paul Street, London EC2A 4JS. Phone: 01-247 8233

July 3-July 16, 1984 No. 3

ZX User

Free with Home Computing Weekly

CONTENTS

This is the third issue of ZX User, a regular treat for all you HCW regulars.

Welcome to an issue crammed with listings and reviews, to help you make the most of your Spectrum's facilities, and keep you up to date with all the releases for your machine.

We also want to hear from you — tell us what you like and what you want to see more of. We welcome your contributions — this is your magazine — and we're interested in your programs.

So don't keep us in the dark — tell us exactly what you think. We always try our best to give you just what you want and we want to hear whether you think we're right.

You can get into print

WE WELCOME programs and articles from our readers. If you feel that your work meets our standards, please submit it to us for consideration for publication.

Programs must always be sent on cassette. Listings are helpful, but not necessary. Check carefully that they are bug-free. Include full details of what your program does, how it works, variables you have used and hints on conversion. See the programs in this issue for guidance on what your paperwork should include.

Articles on using the Spectrum and the ZX81 should be no longer than 2,000 words. Those most likely to be published will help our readers make better use of their computers by giving useful advice, possibly with programming examples, tables and so on. Short hints are also welcome.

All submissions will be acknowledged and the copyright in such works which will pass to Argus Specialist Publications Ltd will be paid for at competitive rates.

Keep a copy of your work and include and SAE. Label everything clearly and give a daytime and home phone number if you can. All work for consideration should be sent to:

Paul Liptrot, ZX User, No. 1 Golden Square, London W1R 3AB

Editor
Paul Liptrot

Assistant Editor
Liz Graham

Design
MM Design

Managing Editor
Ron Harris

Group Editor
Elspeth Joiner

Advertisement Manager
Paul Stanyer

Divisional Advertisement Manager
Coleen Pimm

Chief Executive
Jim Connell

Argus Specialist Publications Ltd,
No.1 Golden Square,
London W1R 3AB
01-437 0626

ZX User is published fortnightly with Home Computing Weekly. Subscriptions and back issues: Infonet Ltd, 10-13 Times House, 179 Marlowes, Hemel Hempstead, Herts HP1 1BB. 0442 48432. Trade distribution: Argus Press Sales and Distribution Ltd, 12-14 Paul Street, London EC2A 4JS. 01-247 8233. Printed by Alabaster Passmore & Sons Ltd, of London and Maidstone, Kent. Design and origination: MM Design, Circus House, 26 Little Portland Street, London W1N 5AF

CHECKMATE

Use this handy program by Mike Biddell to store and play chess games on your ZX81. Your move next

As a chess enthusiast, I decided that my scattered games listings needed organising, so that I could access them readily. I decided to use my trusty ZX81 to carry out the task (a computer for which my enthusiasm remains undiminished).

The initial design considerations were:

- Double board, with pieces moved on both boards simultaneously — so each player sees the board from his point of view.
- Auto-Replay, so that stored games can be replayed automatically (hands off), with a pause function to stop the action at any point.
- Listing of move notation to screen or printer.
- Games recorded automatically as play proceeds.
- Manual mode to allow games to be replayed one move at a time.
- Games up to 80 moves in length.
- A storage capacity of 30 games per C60 tape (120 games if you use QSAVE or a simular device).
- Standard international notation for moves.

The final program proved to be an extremely convenient way of both storing games and playing them. An additional feature which enthusiasts can add is a chess clock.

Enter the program listing. Note that the contents of the REM statements in lines 1 and 2 are irrelevant, as long as each contains at least 161 characters. Please also note that I have sinned and used O as a variable in lines 2330, 2335 and 2340, for which I apologise to the purists.

Now RUN the program and as a test piece, enter the game shown in Listing 2. Moves are entered as two digits followed by Return. For example, to move the king pawn key in

52 Return 54 Return.

When you have completed entry of the game use 0 Return to signify end of recording. The computer will now execute STOP and you can SAVE the game together with the program on to tape in the normal way.

A summary of the operating features is presented below:

0 End of recording, used as described above.

1 Replay recording, one move at a time.

2 Reset board, ready to replay, or make a fresh recording.

3 Correct a move. If you make a mistake enter 3 Return, move the piece back where it came from, then enter 3 Return, followed by the correct move.

4 Castling or en-passent. These are considered to be two moves. For en-passent move your pawn on to the square occupied by your opponent's pawn, then enter 4 Return 4 Return (yes, twice), then move the pawn forward to its correct location. For castling move the king first, then enter 4 ENTER 4 ENTER followed by the rook move.

5 Copy position to printer. At interesting points during the game you can copy the positions on both boards to the printer, using 5 ENTER. The game can then continue as normal.

During Auto-Replay, holding down "P" will stop the game until you release.

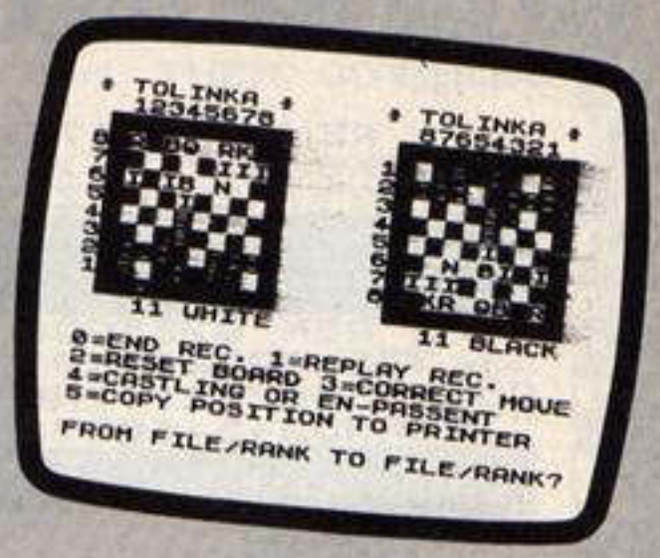
You can save an adjourned game by **not** entering 0 ENTER when you finish. When you reload, put the program into the Auto-Replay mode and it will stop at the adjourned position, ready to continue.

Variables

A$ select listing
B$ select auto-replay
S select board flag (S = 0 left board, S = 1 right board)
E input move numbers (from?)
B move numbers (dummy)
F input move numbers (to?)
C address in first REM for move number storage
D address in second REM for move number storage
Z move number count
X flag for position of move indicator
G,H,P,R compute screen address for POKEs
Q screen address (from?)
O screen address (to?)
V code for piece to be moved
X$ select copy

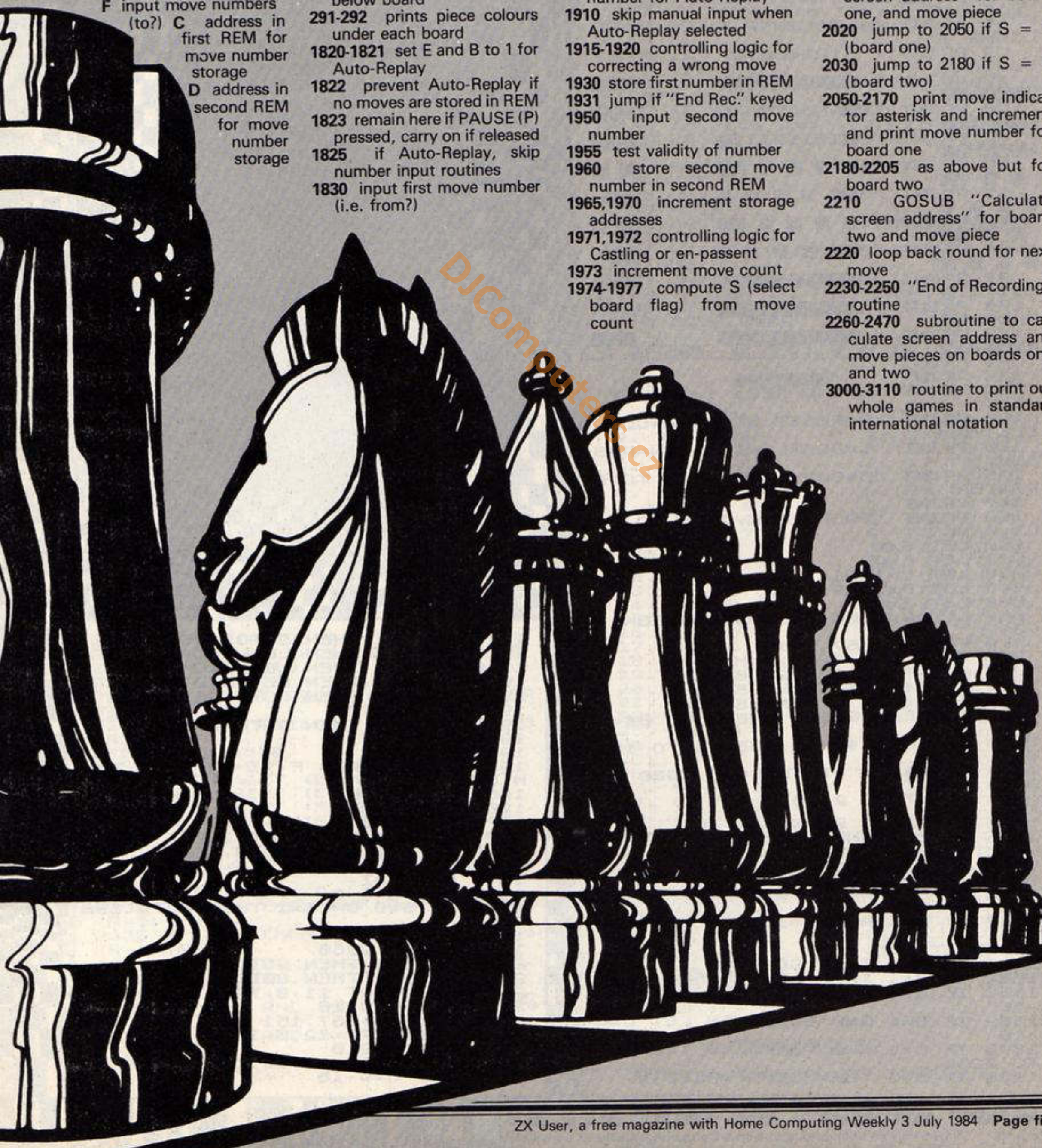
How it works

1,2 REMs to store all moves
3-10 select LISTING or AUTO-REPLAY
18-190 print board and pieces
210-260 initialise variables (260 sets W to first address in screen display)
270 POKEs move indicator below board
291-292 prints piece colours under each board
1820-1821 set E and B to 1 for Auto-Replay
1822 prevent Auto-Replay if no moves are stored in REM
1823 remain here if PAUSE (P) pressed, carry on if released
1825 if Auto-Replay, skip number input routines
1830 input first move number (i.e. from?)
1831 blank out old move numbers
1832 print move number to ?
1835 check valid input, loop back if not
1836-1837 copy screen if 5 pressed, then loop back
1840-1850 controlling logic for correcting a mistake in a move
1860 set up dummy variable B
1870 reset board if B = 2
1880 replay recording, peek REM for move number
1881-1882 controlling logic en-passent or Castling
1890 stop Auto-Replay if zero encountered in REM
1900 let F = second move number for Auto-Replay
1910 skip manual input when Auto-Replay selected
1915-1920 controlling logic for correcting a wrong move
1930 store first number in REM
1931 jump if "End Rec." keyed
1950 input second move number
1955 test validity of number
1960 store second move number in second REM
1965,1970 increment storage addresses
1971,1972 controlling logic for Castling or en-passent
1973 increment move count
1974-1977 compute S (select board flag) from move count



Screen dump

1980 GOSUB "Calculate screen address" for board one, and move piece
2020 jump to 2050 if S = 0 (board one)
2030 jump to 2180 if S = 1 (board two)
2050-2170 print move indicator asterisk and increment and print move number for board one
2180-2205 as above but for board two
2210 GOSUB "Calculate screen address" for board two and move piece
2220 loop back round for next move
2230-2250 "End of Recording" routine
2260-2470 subroutine to calculate screen address and move pieces on boards one and two
3000-3110 routine to print out whole games in standard international notation



Listing 1

```
    1 REM 09?TE7Z) +03?XJQSN■ ??G.P
U■ ?DAX8GKZ?"?F)-75SDI?KOH6EN4(0£
KOE?????PI??6?ZT?£ZI?RS"J+S-6?5
[redacted]
    2 REM Q7ZSGGG(58PPIFHHH?■ ¥I68
G?■ ?ZSN7PIPIPIPIDKO0*6HHH??H6E(4
OOOK:E6£??ZT?INKEY$INKEY$PI?"?"IZ
"?RSRNDJ+S-6£5?J[redacted]
    3 CLS
    4 PRINT "DO YOU WANT A LISTIN
G   Y OR N?"
    5 INPUT A$
    6 IF A$="Y" THEN GOTO 3000
    7 CLS
    8 PRINT "AUTO REPLAY Y OR N?"
    9 INPUT B$
   10 CLS
   18 PRINT "* TOLINKA *       * TO
LINKA *"
   19 PRINT "   12345678         87
654321"
   20 PRINT " [redacted]
[redacted]"
   30 PRINT "8■RNBQKBNR■       1■N
BKQBNR■"
   40 PRINT "7■IIIIIIII■       2■II
IIIIII■"
   50 PRINT "6■ ■ ■ ■ ■■       3■ ■
 ■ ■ ■■"
   60 PRINT "5■■ ■ ■ ■ ■       4■■
■ ■ ■ ■"
   70 PRINT "4■ ■ ■ ■ ■■       5■ ■
 ■ ■ ■■"
   80 PRINT "3■■ ■ ■ ■ ■       6■■
■ ■ ■ ■"
   90 PRINT "2■[redacted]       7■II
IIIIII■"
  100 PRINT "1■[redacted]       8■RN
BKQBNR■"
  110 PRINT " [redacted]
[redacted]"
  120 PRINT
  130 PRINT
  150 PRINT "0=END REC. 1=REPLAY
REC."
  160 PRINT "2=RESET BOARD 3=CORR
ECT MOVE"
  170 PRINT "4=CASTLING OR EN-PAS
SENT"
  175 PRINT "5=COPY POSITION TO P
RINTER"
  180 PRINT
  190 PRINT "FROM FILE/RANK TO FI
LE/RANK?"
  210 LET S=0
  220 LET E=0
  230 LET C=16514
  240 LET D=16694
  250 LET Z=1
  260 LET W=PEEK 16396+256*PEEK 1
6397
  270 POKE W+367,151
  291 PRINT AT 12,6;"WHITE"
  292 PRINT AT 12,22;"BLACK"
 1820 IF B$="Y" THEN LET E=1
 1821 IF B$="Y" THEN LET B=1
 1822 IF PEEK C=128 THEN LET B$="
N"
 1823 IF INKEY$="P" THEN GOTO 182
3
 1825 IF B$="Y" THEN GOTO 1880
 1830 INPUT E
 1831 PRINT AT 21,0;"
"
 1832 PRINT AT 21,0;E;" TO ";
 1835 IF E>5 AND E<11 OR E>88 THE
N GOTO 1830
 1836 IF E=5 THEN COPY
 1837 IF E=5 THEN GOTO 1830
 1840 IF E=3 AND S=0 THEN LET W=W
-16
 1850 IF E=3 AND S=1 THEN LET W=W
+16
 1860■LET B=E
 1870 IF B=2 THEN GOTO 1
 1880 IF B=1 THEN LET E=PEEK C
 1881 IF E=4 AND S=0 THEN LET W=W
-16
 1882 IF E=4 AND S=1 THEN LET W=W
+16
 1890 IF B=1 AND PEEK C=0 THEN GO
TO 2230
 1900 IF B=1 THEN LET F=PEEK D
 1910 IF B=1 THEN GOTO 1965
 1915 IF B=3 THEN LET Z=Z-1
 1916 IF B=3 THEN LET C=C-1
 1917 IF B=3 THEN LET D=D-1
 1920 IF B=3 THEN GOTO 1830
 1930 POKE C,E
 1931 IF E=0 THEN GOTO 2480
 1950 INPUT F
 1951 PRINT F
 1955 IF F>4 AND F<11 OR F>88 THE
N GOTO 1950
 1960 POKE D,F
 1965 LET C=C+1
 1970 LET D=D+1
 1971 IF E=4 THEN LET Z=Z-1
 1972 IF E=4 THEN GOTO 1820
 1973 LET Z=Z+1
 1974 LET S=Z
 1975 LET S=S-2
 1976 IF S=0 OR S=1 THEN GOTO 198
0
 1977 GOTO 1975
 1980 GOSUB 2260
 2020 IF S=0 THEN GOTO 2050
 2030 IF S=1 THEN GOTO 2180
 2050 PRINT AT 11,3;"■■"
 2060 LET W=W+16
 2070 POKE W+367,151
 2087 PRINT AT 12,3;INT (Z/2)
 2170 GOTO 2210
 2180 PRINT AT 11,19;"■■"
 2190 LET W=W-16
```

```
2385 IF X=0 AND S=1 THEN GOTO 24
20
2390 GOTO 2360
2400 POKE Q,0
2410 GOTO 2430
2420 POKE Q,8
2430 LET G=9-G
2440 LET H=9-H
2450 LET P=9-P
2460 LET R=9-R
2470 RETURN
2480 STOP
3000 CLS
3005 LET C=16514
3006 LET D=16694
3010 PRINT "          TOLINK"
3020 PRINT
3030 PRINT "     WHITE                B
LACK"
3040 FOR J=1 TO 100
3045 IF PEEK C=0 OR PEEK D=0 THE
N GOTO 3095
3046 IF PEEK C=4 THEN LET C=C+2
3047 IF PEEK D=4 THEN LET D=D+2
3050 PRINT J;". ";PEEK C;" TO ";
PEEK D;
3060 LET C=C+1
3070 LET D=D+1
3071 IF PEEK C=0 OR PEEK D=0 THE
N GOTO 3095
3072 IF PEEK C=4 THEN LET C=C+2
3073 IF PEEK D=4 THEN LET D=D+2
3080 PRINT TAB 18;J;". ";PEEK C;
" TO ";PEEK D
3081 LET C=C+1
3082 LET D=D+1
3090 NEXT J
3095 PRINT
3096 PRINT "          END OF GAME"
3097 PRINT
3098 PRINT "PRESS C NEWLINE TO C
ONTINUE"
3099 INPUT X$
3100 IF X$="C" THEN COPY
3101 IF X$="C" THEN GOTO 7
3110 GOTO 3099
```

```
2200 POKE W+367,151
2205 PRINT AT 12,19;INT (Z-1)/2
2210 GOSUB 2310
2220 GOTO 1820
2230 PRINT AT 20,0;"END OF RECOR
DING"
2240 PRINT "PRESS 2  TO RESET BO
ARD"
2250 GOTO 1830
2260 LET H=0
2270 LET G=INT (E/10)
2280 LET H=INT (E-10*G)
2290 LET P=INT (F/10)
2300 LET R=INT (F-10*P)
2310 IF S=1 THEN LET 0=W+109+33*
(H-1)-(G-1)
2315 IF S=0 THEN LET 0=W+332-33*
(H-1)+G
2320 LET V=PEEK 0
2330 IF S=1 THEN LET 0=W+109+33*
(R-1)-(P-1)
2335 IF S=0 THEN LET 0=W+332-33*
(R-1)+P
2340 POKE 0,V
2350 LET X=G+H
2360 LET X=X-2
2370 IF X=1 AND S=0 THEN GOTO 24
00
2375 IF X=0 AND S=0 THEN GOTO 24
20
2380 IF X=1 AND S=1 THEN GOTO 24
00
```

Listing 2

TOLINK

```
     WHITE                BLACK
1.  52 TO 54         1.  37 TO 35
2.  71 TO 63         2.  57 TO 56
3.  42 TO 44         3.  35 TO 44
4.  63 TO 44         4.  17 TO 16
5.  21 TO 33         5.  28 TO 36
6.  31 TO 53         6.  78 TO 66
7.  61 TO 43         7.  47 TO 45
8.  54 TO 45         8.  56 TO 45
9.  51 TO 71         9.  68 TO 46
10.  44 TO 36        10.  27 TO 36
11.  53 TO 44        11.  58 TO 78
12.  41 TO 63        12.  38 TO 56
13.  61 TO 51        13.  36 TO 35
14.  44 TO 66        14.  48 TO 66
15.  63 TO 66        15.  77 TO 66
16.  11 TO 41        16.  68 TO 48
17.  43 TO 52        17.  18 TO 28
18.  22 TO 23        18.  35 TO 34
19.  33 TO 45        19.  56 TO 45
20.  41 TO 45        20.  46 TO 82
21.  71 TO 82        21.  48 TO 45
22.  52 TO 34        22.  45 TO 42
23.  34 TO 16        23.  42 TO 32
24.  51 TO 52        24.  32 TO 52
25.  16 TO 52        25.  28 TO 48
26.  12 TO 14        26.  48 TO 42
27.  52 TO 34        27.  42 TO 12
28.  82 TO 73        28.  78 TO 68
29.  73 TO 63        29.  68 TO 57
30.  72 TO 74        30.  66 TO 65
31.  74 TO 65        31.  67 TO 66
32.  34 TO 78        32.  87 TO 86
33.  63 TO 73        33.  57 TO 46
34.  73 TO 63        34.  12 TO 11
35.  63 TO 72        35.  46 TO 55
36.  78 TO 56        36.  55 TO 64
37.  56 TO 47        37.  11 TO 21
38.  47 TO 56        38.  21 TO 22
39.  56 TO 34        39.  22 TO 12
40.  34 TO 56        40.  86 TO 85
41.  56 TO 47
```

END OF GAME

PRESS C NEWLINE TO CONTINUE

Keep tabs on REST of the WORLD 0

This program provides a handy scoreboard for any game where a pencil and paper would normally be used to record scores.

When RUN the participants' names are requested. It may be most convenient to enter names in the order of play or how the players are seated. You don't have to key in the initial player first. When all names are entered, press 0 to escape the identification routine. If seven names are listed, escape is automatic.

The computer now requests entry of player 1's score. Enter the score which can be positive, negative or a calculation. For example, enter 5 or -10 or 3×9-22/2. All entries are acceptable in the range -9999 to 9999. If an input score becomes too long it is rejected and another score is requested. This would happen if pi were to be entered as 22/7.

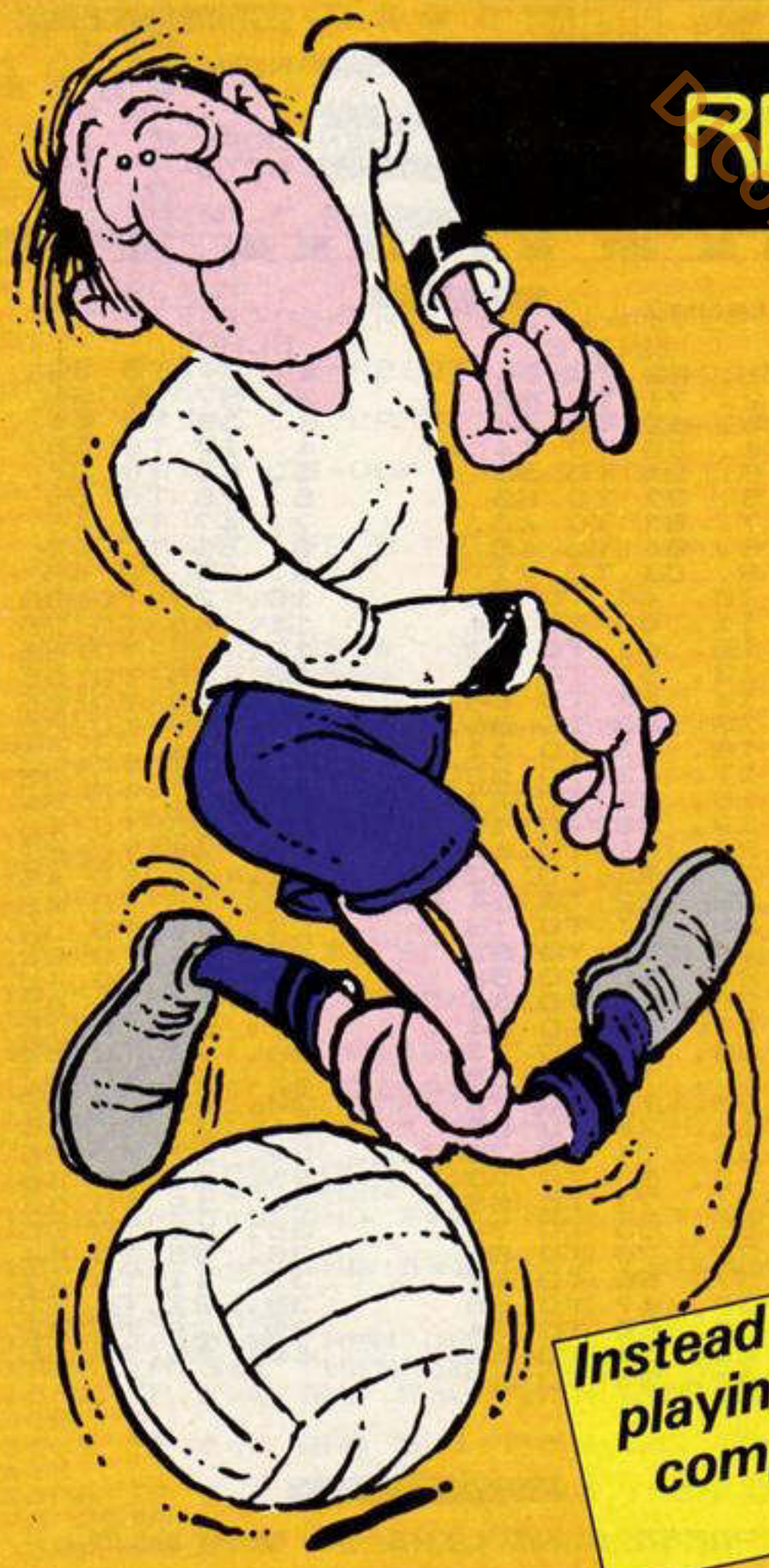
The computer now asks if the entered number is correct. If the number shown in the score box is correct then press Y,y or ENTER. If the number is wrong press N or n and enter the correct number. This facility guards against "finger trouble" when entering the scores.

When the current score is approved the player's total is updated and the next player's score is requested.

Now you're all set to have a computerised scoreboard. Don't cheat though — even if you're the only person in charge of scoring.

Listing for Scoreboard

```
 10 REM ** SCOREBOARD **
 20 REM ** D.G.HUGHES **
 30 REM ** MAY10~1984 **
100 REM set variables
110 LET a=0: LET b=0: LET c=0: LET d=0: LET e=0: LET f=0: LET g=0
120 DIM n$(8,10)
150 GO SUB 9000: REM title
160 GO SUB 8000: REM start
190 REM player identification
200 FOR n=1 TO 7
210 PRINT AT 20,2;"PRESS '0' WHEN LIST COMPLETE"
220 INPUT "PLAYER ";n$(n)
230 IF n$(n,1)="0" THEN  LET q=n-1: PRINT AT 20,2; FLASH 1;"                O.K.
       ": PAUSE 50: FLASH 0: GO TO 300
240 PRINT AT 4+n*2,1;n;":";n$(n)
250 BEEP .1,n*5
260 IF n>=7 THEN  GO TO 280
270 NEXT n
280 LET q=n
290 REM score entry loop
300 FOR n=1 TO q
310 PRINT  FLASH 1;AT 4+n*2,1;n;AT 4+n*2,3;n$(n)
320 PRINT AT 4+n*2,16;"      "
```



Variable

- **a to g** players' total scores
- **s** input score
- **1** number of players programmed
- **p** vertical position of current player's score
- **n$** player's name
- **g$** input score verification

How it works

100-160 set variables, dimension (DIM) name strings, call subroutines to format screen
200-280 request and enter players' names (n$)
290-460 request and verify players scores in sequence
500-570 update each player's total score
8000-3200 end of round subroutine
9000-9200 title and frame subroutine

Conversion hints

DIM n$ (x,y) sets up x arrays to a maximum length of y characters. *STR$(x)* converts a number, x, into string form, e.g. if x = 50 then STR$(x) = "50". **LEN** gives the length of a string, e.g. LEN "50" = 2. **BRIGHT** and **FLASH** attributes are set to on with 1

Instead of laboriously writing out scores when playing games, why not input them on your computer? This program by David Hughes shows you how

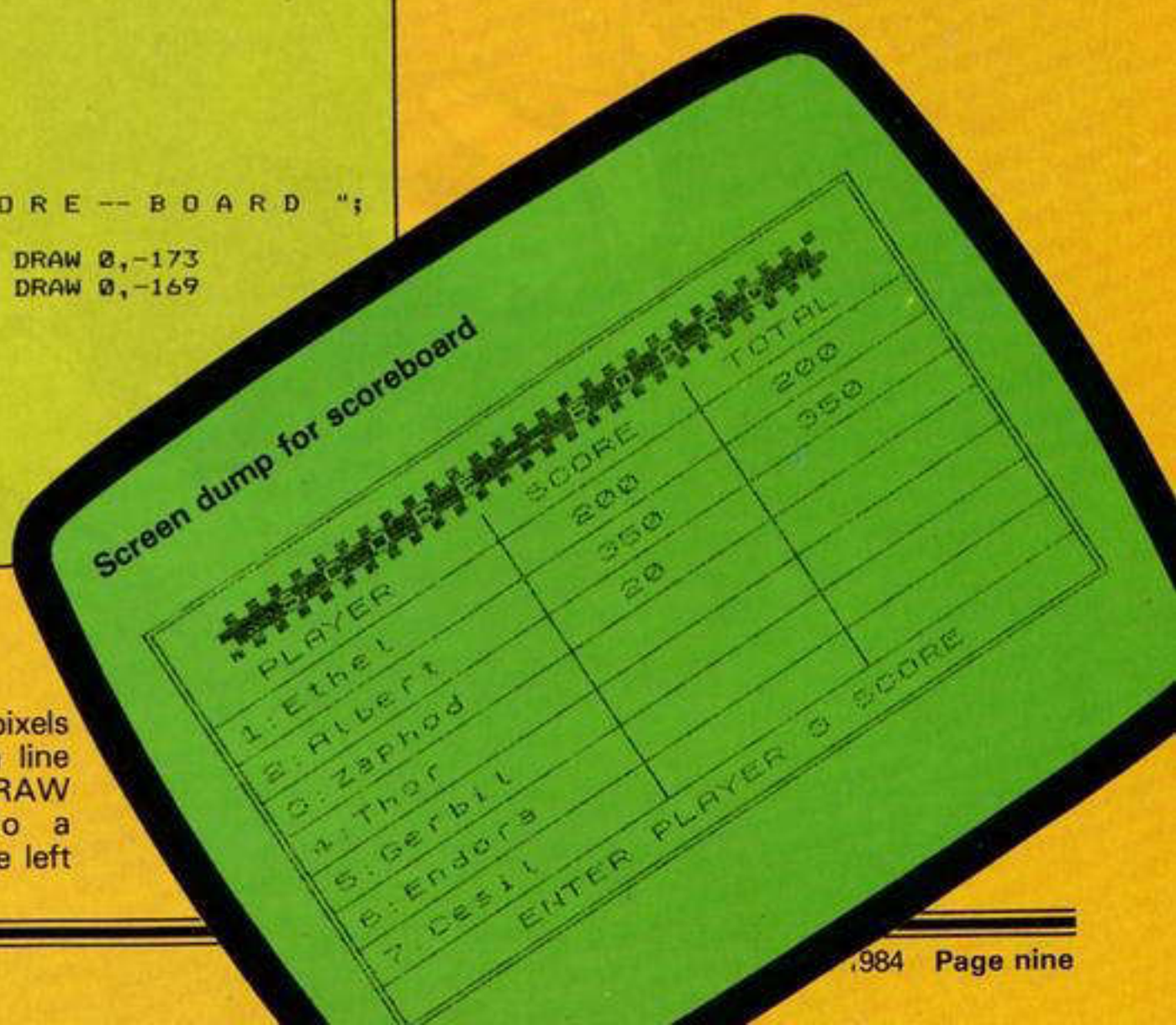
high scores

```
330 PRINT AT 20,1;"     ENTER PLAYER ";n;" SCORE     "
340 INPUT s: IF s>=9999 OR s<=-9999 THEN  GO TO 340
345 IF LEN STR$ (s)>6 THEN  PRINT AT 20,1;"SORRY, THAT NUMBER IS
TOO LONG": FOR z=10 TO -30 STEP -5: BEEP .1,z: NEXT z: GO TO 330
350 PRINT AT 4+n*2,16;s
360 INPUT "IS ENTRY CORRECT ? (Y/N) ";q$
370 IF q$="n" OR q$="N" THEN  PRINT AT 4+n*2,16;"     ": GO TO 340
380 IF s<65 AND s>-40 THEN  BEEP .1,s
390 IF s>65 AND s<650 THEN  BEEP .1,s/10
400 IF s>650 THEN  BEEP .1,65
410 IF s<-40 THEN  BEEP .1,-40
420 PRINT  FLASH 0;AT 4+n*2,1;n;AT 4+n*2,3;n$(n)
430 LET p=4+n*2
440 GO SUB 500+n*10: REM total
450 NEXT n
460 GO SUB 8000: GO TO 300
500 REM update all players'                    total scores
510 LET a=a+s: PRINT AT p,25;"     ";: PRINT AT p,25;a: RETURN
520 LET b=b+s: PRINT AT p,25;"     ";: PRINT AT p,25;b: RETURN
530 LET c=c+s: PRINT AT p,25;"     ";: PRINT AT p,25;c: RETURN
540 LET d=d+s: PRINT AT p,25;"     ";: PRINT AT p,25;d: RETURN
550 LET e=e+s: PRINT AT p,25;"     ";: PRINT AT p,25;e: RETURN
```

ENGLAND 26

Half Time


```
 560 LET f=f+s: PRINT AT p,25;"     ";: PRINT AT p,25;f: RETURN
 570 LET g=g+s: PRINT AT p,25;"     ";: PRINT AT p,25;g: RETURN
8000 REM end of round routine
8005 INK 1: PAPER 7: BRIGHT 1
8010 FOR n=3 TO 28
8020 PRINT AT 1,n;" ";: NEXT n
8030 PRINT AT 1,29;" ";AT 2,29;" ";AT 3,29;" "
8040 FOR n=28 TO 3 STEP -1
8050 PRINT AT 3,n;" ";: NEXT n
8060 PRINT AT 3,2;" ";AT 2,2;" ";AT 1,2;" "
8070 FOR n=3 TO 28
8080 PRINT AT 1,n;" ";: NEXT n
8090 PRINT AT 1,29;" ";AT 2,29;" ";AT 3,29;" "
8100 FOR n=28 TO 3 STEP -1
8110 PRINT AT 3,n;" ";: NEXT n
8120 PRINT AT 3,2;" ";AT 2,2;" ";AT 1,2;" "
8150 INK 0: PAPER 7: BRIGHT 0
8200 RETURN
9000 REM  title & frame
9010 PRINT  INK 1; PAPER 7; BRIGHT 1;AT 2,3;"  S C O R E -- B O A R D  ";
9100 PRINT AT 4,2;" PLAYER       SCORE     TOTAL ";
9102 PLOT 1,1: DRAW 253,0: DRAW 0,173: DRAW -253,0: DRAW 0,-173
9104 PLOT 3,3: DRAW 249,0: DRAW 0,169: DRAW -249,0: DRAW 0,-169
9110 PLOT 106,144
9115 DRAW 0,-124
9120 PLOT 178,144
9125 DRAW 0,-124
9130 FOR n=1 TO 8
9140 PLOT 4,4+n*16
9150 DRAW 247,0
9160 NEXT n
9200 RETURN
```



and to off with 0. **PRINT AT x,y;** "z" prints a character z at a position x characters down and y characters across from the left of screen.
PLOT x,y locates a pixel x pixels from the left and pixels up. **DRAW x,y** draws a line from the last PLOT or DRAW statement position to a position x pixels from the left and y pixels up.

This game of logic can be so frustrating, but satisfying when you finally get it right.

What you have to do is guess the four coloured pegs which are in sequence out of your sight. You are told if you have guessed the right peg in the wrong hole and if you get the right peg in the right hole. However, it's not as simple as it sounds and takes a lot of thought and due consideration.

You have 10 attempts and there are six colours from which the pegs may be selected. And don't despair — if you still haven't cracked it after 10 guesses, you'll be told the formula at the end.

Instructions are included in the program and you can choose whether to read them or not. If you want to convert this game, re-arrange the screen displays. Also check the many variables used for little loops.

So, get ready for some brainteasing fun. It's all in the process of elimination, so marshall your thought, and off you go!

How it works

1-10 sort out whether instructions are wanted, go to correct area if yes

11-50 set up and check that no colours appear twice in sequence

54-89 input guess and check for correct length, ask if you want to change sequence

90-120 print up your guess in the columns

200-240 check if your guess is correct and if you ran out of attempts. If not, have another go

500-1160 check for correct colour in wrong place or correct colour in correct place

1200-1310 print out the series of circles and dots in a random order

4000-4050 GOSUBs to check your solution against the correct one

5000-5510 set up series of circles and dots to represent your guess

5600-5830 print up of instructions

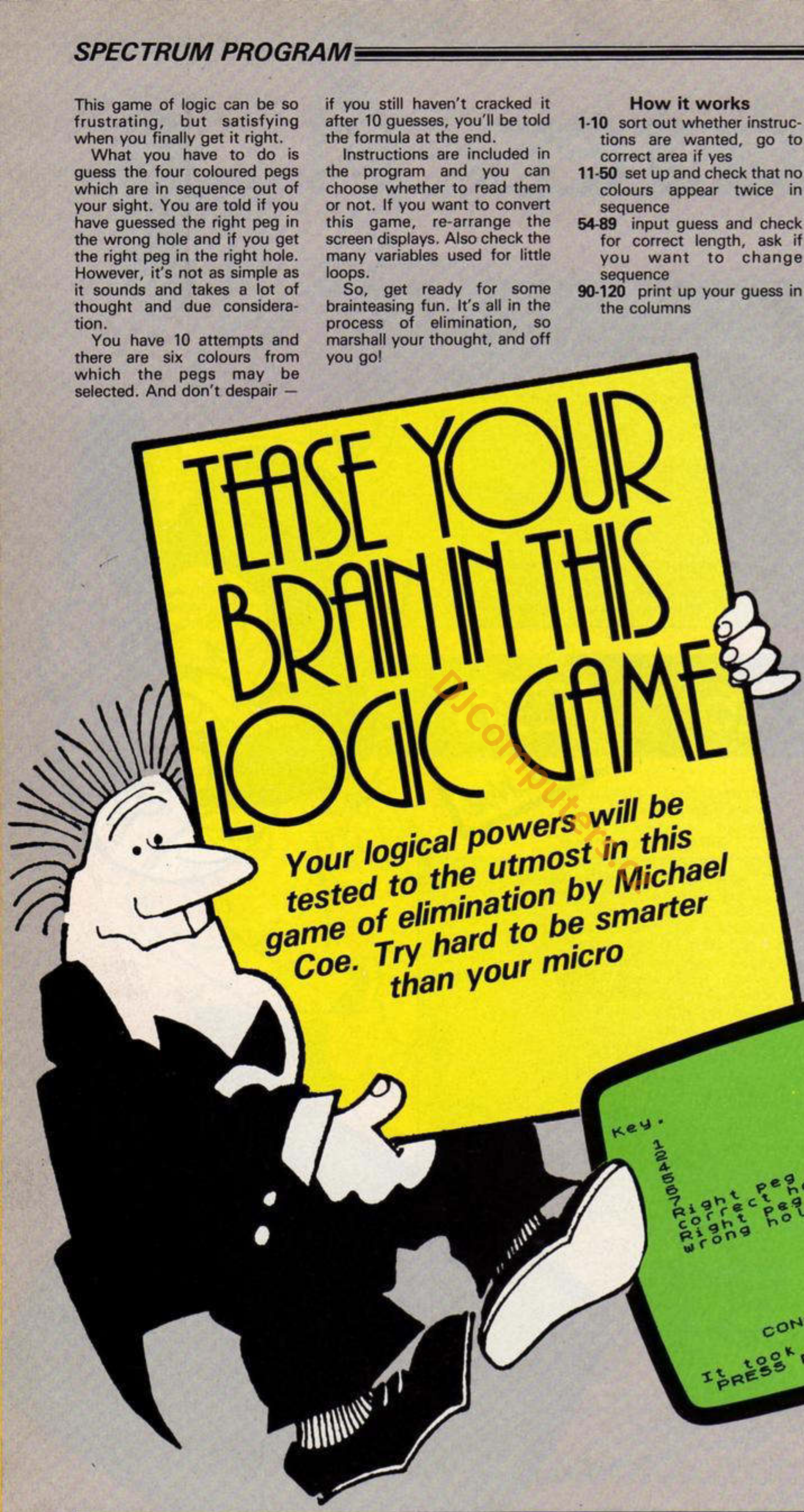
7000-7060 print up answer if you ran out of attempts

8000-8999 print up if you guess correctly

9000-9320 set up initial screen

Variables

m number of attempts
c answer colour generator
a$ hidden answer
b$ your guess
i print your guess in sequence
n part of b$ to be checked against a$
o part of a$ to be used to check b$
h sort out order to print dots and circles
j,d,g, used to print out dots and circles
p,q circle and dot co-ordinates
e$ sequence of dots and circles



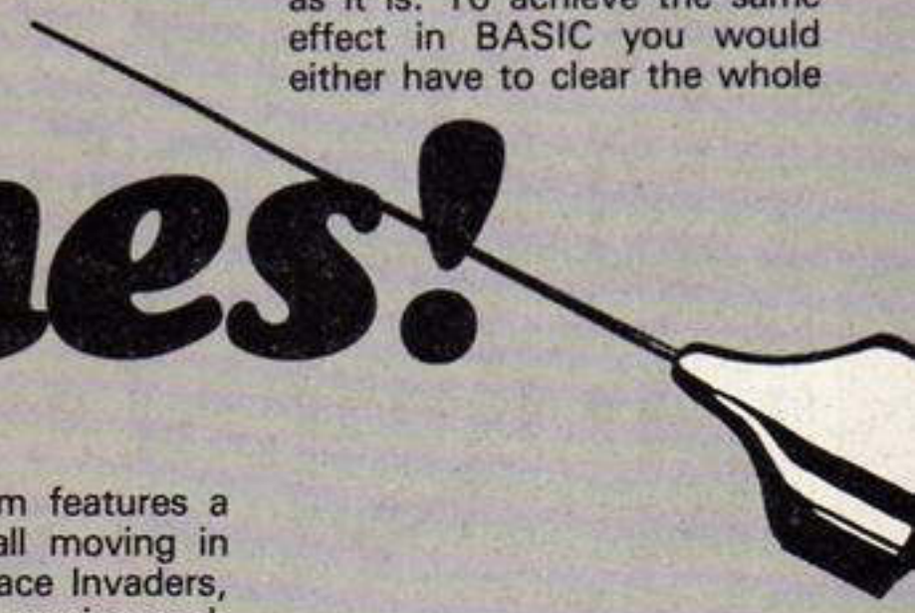
Screen dump

```
 2 INK 0
 3 CLS
 4 PRINT AT 10,0;"Do you want instructions y or n"
 5 INPUT a$: IF a$="y" THEN  GO SUB 5600
 9 CLS
10 GO SUB 9000
11 LET m=0
14 LET a$=""
15 FOR f=1 TO 4
20 LET c=INT (RND*7)+1
30 IF c=3 THEN  GO TO 20
35 FOR v=1 TO LEN a$: IF c=VAL a$(v TO v) THEN  GO TO 20
36 NEXT v
40 LET a$=a$+STR$ c
50 NEXT f
54 LET x=4
60 LET b$=""
70 PRINT AT 10,0;"Select sequence to be inputed   "
75 PRINT AT 19,10;"              "
76 INPUT b$
77 IF LEN b$<4 THEN  GO TO 76
78 FOR f=1 TO 4
79 LET i=VAL b$(f TO f)
80 PRINT  PAPER i; INK 9;AT 15,19+f;i
81 NEXT f
84 PRINT AT 18,0;"Are you happy with this sequence";AT 19,14;"y or n"
85 IF INKEY$="y" THEN  GO TO 90
86 IF INKEY$="n" THEN  GO TO 88
87 GO TO 85
88 PRINT  INK 0;AT 15,19;"■■■■"
89 GO TO 60
90 FOR f=1 TO 4
91 PRINT AT 15,19+f;"■"
100 LET i=VAL b$(f TO f)
105 FOR z=0 TO 10: NEXT z
110 PRINT  PAPER i; INK 9;AT x,19+f;i
120 NEXT f
200 IF a$=b$ THEN  GO TO 8000
205 LET m=m+1
207 PRINT AT 18,0;"                                    ";AT 19,10;"
210 PRINT AT 18,10;"wrong ";
215 GO SUB 4000
220 IF m=10 THEN  GO TO 7000
225 LET x=x+1
230 PRINT AT 18,10;"          "
240 GO TO 60
500 FOR r=1 TO 4
510 LET n=VAL b$(r TO r)
520 LET o=VAL a$(f TO f)
530 IF n=o AND f<>r THEN  GO TO 5000
540 IF n=o THEN  GO TO 5500
550 NEXT r
560 RETURN
700 FOR r=2 TO 4
710 LET n=VAL b$(r TO r)
720 LET o=VAL a$(f TO f)
730 IF n=o AND f<>r THEN  GO TO 5000
740 IF n=o THEN  GO TO 5500
750 NEXT r
760 LET n=VAL b$(1)
770 IF n=o THEN  GO TO 5500
780 RETURN
900 FOR r=3 TO 1 STEP -1
910 LET n=VAL b$(r TO r)
920 LET o=VAL a$(f TO f)
930 IF n=o AND r<>f THEN  GO TO 5000
940 IF n=o THEN  GO TO 5500
950 NEXT r
960 LET n=VAL b$(4)
970 IF n=o THEN  GO TO 5500
980 RETURN
1100 FOR r=4 TO 1 STEP -1
1110 LET n=VAL b$(r TO r)
1120 LET o=VAL a$(f TO f)
1130 IF n=o AND f<>r THEN  GO TO 5000
1140 IF n=o THEN  GO TO 5500
1150 NEXT r
1160 RETURN
1200 LET h=INT (RND*2)+1
1210 IF h=2 THEN  LET j=LEN e$
1215 IF h<>2 THEN  LET w=1
1220 IF h<>2 THEN  LET j=1
1225 IF h<>2 THEN  LET w=LEN e$
1230 IF j=LEN e$ THEN  LET b=-1
1240 IF j=1 THEN  LET b=1
1250 FOR g=j TO w STEP b
1260 LET d=VAL e$(g TO g)
1270 IF d=1 THEN  CIRCLE p,q,1.5
1280 IF d=0 THEN  PLOT p,q: PLOT p,q-1: PLOT p-1,q: PLOT p-1,q-1
1290 LET p=p+5
1300 NEXT g
1310 RETURN
4000 LET p=202
4001 LET e$=""
4002 LET q=((21-y)*8)+4
4003 LET t=0
4010 FOR f=1 TO 4
4020 LET t=(f*200)+300
4030 GO SUB t
4040 NEXT f
4045 BEEP .2,0: GO SUB 1200
4050 RETURN
5000 LET e$=e$+"1"
5020 RETURN
5500 LET e$=e$+"0"
5510 RETURN
5600 CLS
5610 PRINT
5615 PRINT
5620 PRINT " The object of the game is to";AT 3,0;"discover the sequence of four
"
5630 PRINT "hidden colours,these are hidden by the computer at the top of "
5640 PRINT "the panel in the game display."
5650 PRINT " To discover the colours you    must enter a selection of four"
5660 PRINT "colours,the computer will then  give you clues as to whether "
5670 PRINT "any of your entered colours are correct.This is done by the"
5680 PRINT "following symbols:-"
5690 CIRCLE 4,60,2
5700 PRINT AT 14,3;"indicates one of your colours"
5710 PRINT AT 15,3;"is correct and in the right     place."
5720 PLOT 4,36: PLOT 5,36: PLOT 4,35: PLOT 5,35
5730 PRINT AT 17,3;"indicates one of your colours   is correct but in
     the wrong place."
5740 PRINT AT 21,6;"PRESS P TO CONTINUE"
5750 IF INKEY$="p" THEN  GO TO 5770
5760 GO TO 5750
5770 CLS
5780 PRINT " To make your selection of four colours enter all four using"
5790 PRINT "the key shown on the game        display and then press enter."
5800 PRINT " You have up to ten atteepts to discover the correct solution."
5810 PRINT AT 10,6;"PRESS P TO PLAY"
5820 IF INKEY$="p" THEN  RETURN
5830 GO TO 5820
6999 STOP
7000 PRINT AT 18,0;"You ran out of atteepts.The     "
7010 PRINT "correct solution was:            "
7020 FOR f=1 TO 4
7030 INK VAL a$(f TO f)
7040 PRINT AT 2,f+19;"■"
7050 NEXT f
7060 GO TO 8500
8000 FOR f=1 TO 4: INK VAL a$(f TO f): PRINT AT 2,f+19;"■": NEXT f
8010 INK 3
8020 PRINT  FLASH 1;AT 18,0;"          CONGRATULATIONS
8025 PRINT AT 19,0;"
8030 PRINT AT 20,2;"  It took you ";m+1;" goes to solve"
8040 FOR n=0 TO 5: FOR f=20 TO 30 STEP 2: BEEP .02,f: NEXT f: NEXT n
8500 PRINT  INK 0;AT 21,5;"PRESS P FOR ANOTHER GO"
8510 LET a$=INKEY$
8520 IF a$="p" THEN  GO TO 9
8999 GO TO 8510
9000 INK 0
9005 FOR f=1 TO 3
9010 PRINT AT f,16;"■■■■■■■■■■■■■■"
9020 NEXT f
9030 FOR f=4 TO 13
9040 PRINT AT f,16;"■■■■■■■■■■■■■■"
9050 NEXT f
9060 PRINT AT f,16;"■■■■■■■■■■■■■■"
9065 PRINT AT f+1,16;"■■■■■■■■■■■■■■"
9066 PRINT AT f+2,16;"■■■■■■■■■■■■■■"
9070 FOR f=10 TO 1 STEP -1
9072 LET l=0
9075 IF f>1 THEN  LET l=1
9080 PRINT  PAPER 0; INK 7;AT 14-f,17+l;11-f
9090 NEXT f
9100 LET f=159
9110 PLOT 160,159
9120 DRAW  OVER 1;31,0
9130 DRAW  OVER 1;0,-8
9140 DRAW  OVER 1;-31,0
9150 DRAW  OVER 1;0,8
9160 FOR f=144 TO 62 STEP -8
9170 PLOT 159,f
9180 DRAW  OVER 1;32,0
9190 NEXT f
9200 PLOT 159,145
9210 DRAW  OVER 1;0,-81
9220 PLOT 192,145
9230 DRAW  OVER 1;0,-81
9240 PRINT AT 0,0;"Key."
9250 FOR f=1 TO 7
9260 IF f=3 THEN  NEXT f
9264 LET j=f
9265 IF f>3 THEN  LET j=j-1
9270 PRINT  INK f;AT j+1,0;"■■ ": INK 0;f
9280 NEXT f
9290 CIRCLE 7,106,2: PRINT AT 8,3;"Right peg in";AT 9,3;"correct hole"
9300 PLOT 7,92: PLOT 7,91: PLOT 8,92: PLOT 8,91: PRINT AT 10,3;"Right peg in";AT
 11,3;"wrong hole"
9310 PLOT 0,160: DRAW 16,0: DRAW 0,-49: DRAW -16,0
9320 RETURN
```

Listing for logic game

Are you fed up with the slow reactions of games you write in BASIC? This utility program by Philip Carre helps you speed up those slow-moving missiles

Inject new ZAP in your games!



So there you are sitting in front of your Spectrum. You've just typed in your latest masterpiece (completely debugged, of course). You press RUN and ENTER then sit back to watch the action.

The alien crawls across the top of the screen — not a good start. You press the key to move the laser base and sit there wondering how it can flicker so fast yet move so slowly. In desperation you jab a finger on the fire button and watch as the missile inches its way up the screen.

You've now got two choices: either you pull the plug and instead load in your latest games tape or you read the rest of this article.

Spectrum BASIC isn't very fast compared to some other computers, but if used efficiently you can still produce some exciting and reasonably fast games.

The first trap most people seem to fall into is using missiles. A missile is extremely slow because it has to advance one square at a time. It's far better to use a laser. Do this by using PLOT and DRAW, as Listing 1 shows. Press key 0 to fire.

If lasers don't fit into the context of your game and you really must use missiles then you can speed them up by using a FOR...NEXT loop around the printing and detection routines. Study this example:

```
MAIN LOOP MOVE ALIEN
FOR a=1 to 3 PRINT MISSILE
   SEE IF MISSILE HAS HIT
   ALIEN
NEXT a MOVE BASE GOTO
   MAINLOOP
```

Most people use

```
LET b = b+ (INKEYS = "x"
   and b  28)-(INKEYS = "z"
   AND b  0)
```

to make the laser base move right and left. Another way of doing it is to use the IN function; enter it as line 70 in the above program.

```
70 LET b=b+ (IN 65278 =
   251 and b  28)-(IN 65278 =
   253 AND b  0)
```

This is only slightly faster but with Spectrum BASIC every little bit helps. This is for key "z" to go left and key "x" to go right If you want to alter the keys, refer to page 160 in the Spectrum manual; this gives you the addresses for each half row of the keyboard. As an exercise you could also change the fire routine in line 80 to use this method.

If your program features a lot of graphics all moving in unison, as in Space Invaders, then printing and erasing each one separately is very slow and tedious. The best way to do this is to put them in a string. Then you can use the Spectrum's string slicing facility to decrease the length of the string as each graphic is hit. This has the added bonus that as the string gets smaller there is less to print so the graphics appear to move faster.

Another problem is not being able to move and fire at the same time. Using the laser routine helps eliminate this, but it is still noticeable. The way to get round this is to intentionally stop the graphics, so that instead of moving smoothly across the screen and then stopping when you press the fire button, they are programmed to move and stop, move and stop, so that when you fire, although the action stops, you don't notice it.

If you synchronise the length of pause to the amount of time it takes to fire, it will give the illusion of being able to move and fire simultaneously.

Incorporating the above routines as well as the more obvious ones, such as avoiding GOTOs and putting DEF FN statements at the beginning of programs, you should have a reasonably fast game.

If all this still doesn't satisfy you then you must look beyond BASIC. You could use a compiler to speed up your games a lot, but be warned: with most compilers, when you save your game on tape it also saves part of the compiler, which it needs to run the program. Although this doesn't matter for home use it's not very good if you have any dreams of selling your programs.

The final resort — and the best one — is to learn machine code. The second listing is a demonstration of the speed of machine code, and it can be incorported into your own BASIC programs. The first part is a routine to clear a specified number of lines on screen. this is especially useful if you want to clear part of the screen display leaving the rest as it is. To achieve the same effect in BASIC you would either have to clear the whole screen and then reprint the parts you need, or use a FOR...NEXT loop to clear each line one character at a time, which isn't very fast. This listing uses the Spectrum ROM and the effect is instantaneous.

The second DATA number in line 60 controls the number of lines to be cleared from the bottom up. The second part of the routine scrolls a specified number of lines up the screen. This can give a very good effect in BASIC program by moving one line or several, lines of characters swiftly up the screen.

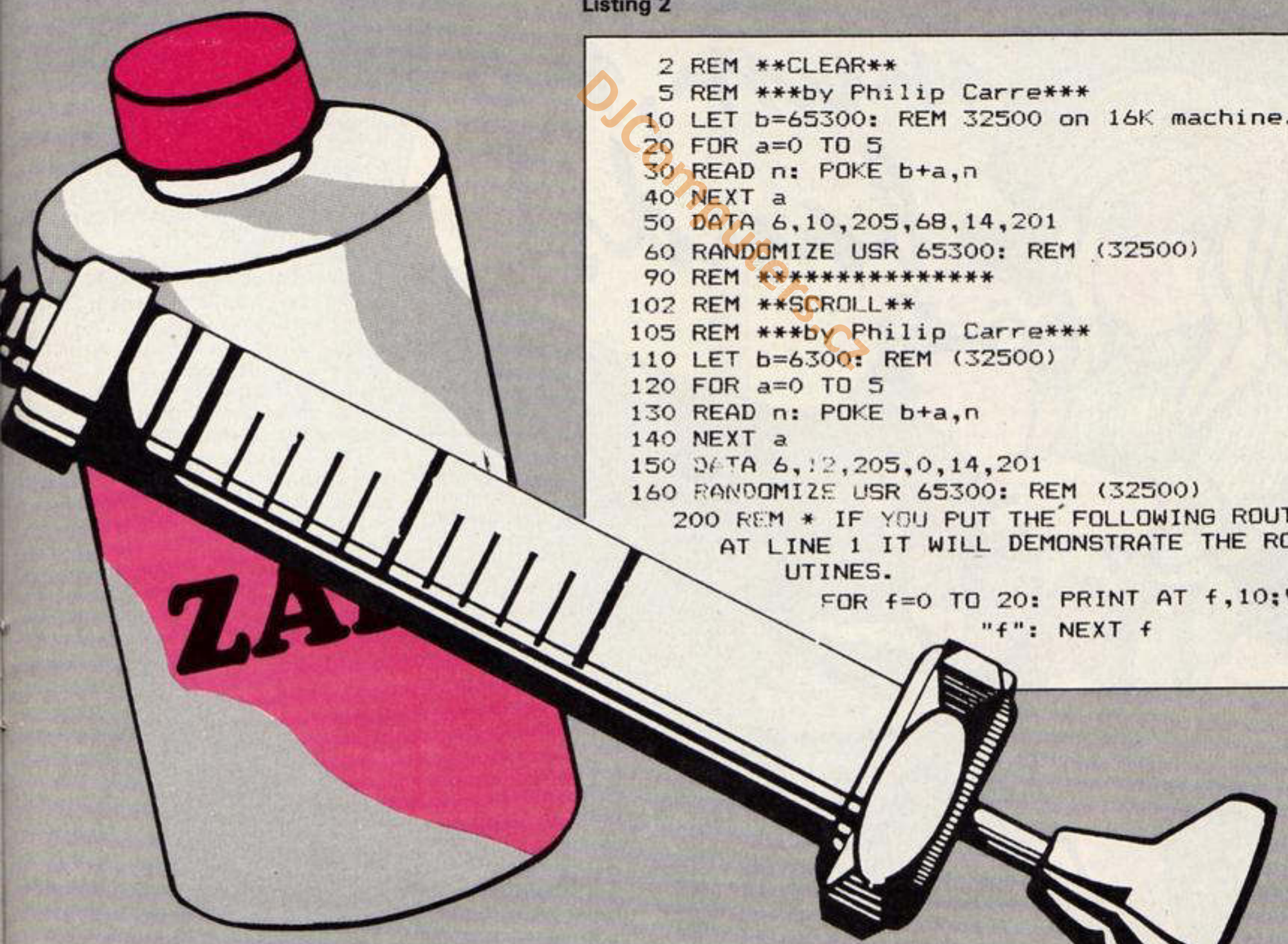
In this routine also, the second DATA number sets the number of lines to be scrolled up the screen, from the bottom up. Numbers in brackets are for 16K machines.

As you can see, these very powerful ROM routines may be accessed using very little machine code.

If you want to learn machine code, it's a lot easier if you use an assembler, because it calculates addresses and jumps for you. With an assembler and a good

Listing 1

```
  5 REM *** by Philip Carre ***
 10 BORDER 2: PAPER 5: BRIGHT 1: CLS
 20 FOR f=USR "A" TO (USR "B"+7): READ n: POKE f,n: NEXT f: REM graphic A and
B
 30 BEEP .05,.05: RESTORE 120
 50 LET a=18: LET b=15
 60 PRINT AT a,b; INK 1;" AB ": REM graphic A and B with 1 space before and af
t
er.
 70 LET b=b+(IN 65278=251 AND b<28)-(IN 65278=253 AND b>0)
 80 IF INKEY$="0" THEN  GO SUB 100
 90 GO TO 60
100 PLOT 8*b+15,(14-a)*8: DRAW  INK 7;0,-(-127): FOR f=60 TO 64: BEEP .01,f: NE
XT f: PLOT  OVER 1;8*b+15,(14-a)*8: DRAW  OVER 1;0,-(-127): OVER 0
110 RETURN
120 DATA 1,3,134,134,254,134,130,3,0,128,194,194,254,194,130,128
130 REM * LINE 70 SHOULD BE  REMOVED FROM THIS LISTING AND PRINTED SEPERATLY AS
 IN THE ARTICLE.
140 REM * REMOVE THE LINE NUMBER FROM THE FOLLOWING LINE AND PRINT AS IN THE AR
TICLE.
150 LET b=b+(INKEY$="x" AND b<28)-(INKEY$="z" AND b>0)
160 REM * THE MACHINE CODE ROUTINES FOLLOW ON TAPE. JUST TYPE  IN  LOAD ""
```



Listing 2

```
  2 REM **CLEAR**
  5 REM ***by Philip Carre***
 10 LET b=65300: REM 32500 on 16K machine.
 20 FOR a=0 TO 5
 30 READ n: POKE b+a,n
 40 NEXT a
 50 DATA 6,10,205,68,14,201
 60 RANDOMIZE USR 65300: REM (32500)
 90 REM ****************
102 REM **SCROLL**
105 REM ***by Philip Carre***
110 LET b=6300: REM (32500)
120 FOR a=0 TO 5
130 READ n: POKE b+a,n
140 NEXT a
150 DATA 6,12,205,0,14,201
160 RANDOMIZE USR 65300: REM (32500)
  200 REM * IF YOU PUT THE FOLLOWING ROUTINE
     AT LINE 1 IT WILL DEMONSTRATE THE RO
     UTINES.
         FOR f=0 TO 20: PRINT AT f,10;VAL
             "f": NEXT f
```


beginners' book on Z80 programming (make sure it mentions the Spectrum in the title) you could find learning machine code a lot easier than you thought.

If the idea of learning machine code doesn't appeal to you, at least your BASIC progams will be a lot faster if you use the routines in this article.

Control your computer by throwing your voice. With these novel programs by Peter Sweasey you can play games without even touching the keyboard

As we progress in the eighties, fact continues to catch up with science fiction. A feature of many futuristic tales is the human communicating with a computer without using a keyboard. For a few years computers for the disabled have already been able to respond to sound, and now your spectrum will be able to do this too, using no extra hardware apart from a cassette recorder and a microphone.

It would be complex and unreliable to program the Spectrum to differentiate between sounds. Therefore these three programs simply "listen" for a sound and only have one control. You may think this would make things cumbersome, but that isn't necessarily the case.

To use the system one simple statement is needed. IN 31000 will scan the EAR port for sound (it will also scan some of the keys but this isn't relevant here). When you want your program to "listen" for a sound, type in a statement like

IF IN 31000 255 THEN...

When no sound is given 255 is the value. Changing the BORDER and PAPER colours sometimes affects most Spectrums, so this statement has been expanded to counteract the result of 191 being given, so:

IF IN 31000 255 AND NOT IN 31000 = 191 THEN ...

As most tape recorders have built-in microphones it is quite simple to speak to your Spectrum. Press Record and the pause button and connect the EAR socket to the computer. Now adjust the volume level: if the microphone is too sensitive it will pick up the tiniest vibrations (even the computer's own sounds) and if it isn't sensitive enough you will find yourself bellowing into it! If you use an external microphone don't hold it too close to the computer. And please note that not all tape recorders will work this way. Computer cassette recorders may well be the worst, and you may find you need to try several different models.

In order to test whether your tape recorder is suited and if you have the correct volume level, type in Listing one. Once RUN, the screen should remain blank yellow until a sound is found, when it will turn red and fire an arrow from left to right, before returning to yellow.

The major use of home computers is, perhaps unfortunately, for games. To demonstrate the capabilities of this system type in Hi-Lo, which could alternatively be called Say Your Cards Right! It's an adaptation of the well-known TV game show in which you are presented with four cards. At the top of the screen are two boxes, one with Hi and other with Lo, and an arrow points to one of the boxes. This arrow swings between the two boxes; when it points to the box you want, speak. You begin the game with £100 and score £100 for each correct guess. You can win a bonus of £200 if you guess correctly four cards in a row. If your guess is wrong, the game restarts. You can break the bank if your luck holds.

There are other games possibilities for the voice-sensitive micro. A menu-driven adventure is possible but it would be slow, unless it was simple, in which case it would be boring.

It would be more interesting to use it as a fire command in arcade games. This would simplify games like Scramble, in which six keys are needed. Speech could be used to pick things up, in Atic Atac, for example. Alternatively you could use the speech facility in menu-driven games designers like Melbourne Houses HURG.

The program Quiz demonstrates a more serious side of speech control. Once RUN, it will start to ask you questions, displaying three possible answers. A flashing cursor will move slowly past each answer. When it passes the one you believe to be correct, speak. There are 10 simple general knowledge questions and it is simple to change the program by altering the DATA statements.

Such a program is ideal for people unfamiliar with the keyboard. It could serve as an introduction to the computer for young schoolchildren or computer illiterate adults. The system is already at use with disabled people.

One word of caution: don't speak for too long in either program, as your computer will rush on without you!

These programs may be one small step towards a future in which keyboards are redundant. Control of computers will be entirely by voice — what a boon for us lazy humans!

● **Listing 1 Hi-Lo**

```
  1 REM          Hi-Lo
  2 REM By Peter Sweasey
  3 FOR v=0 TO 7: READ q: POKE USR "a"+v,q: NEXT v: DATA 136,68,34,17,136,68,34
,17
  4 FOR v=0 TO 7: READ q: POKE USR "b"+v,q: NEXT v: DATA 24,60,126,255,255,126,
60,24
  5 LET mo=100: LET ba=7000
 10 BORDER 0: PAPER 4: CLS
 15 PRINT AT 0,0; PAPER 3; INK 7;"
 20 PRINT AT 0,13; PAPER 6; INK 0; FLASH 1;"HiLo"
 35 INK 1: PAPER 7: PRINT AT 2,0;"
 40 PRINT "£ "; FLASH 1; INK 2;"Hi"; INK 1; FLASH 0;"
;
 FLASH 1; INK 2;"Lo"; FLASH 0; INK 1;"
 45 PRINT "
 50 FOR w=2 TO 5: PRINT AT w,6;"                    ": NEXT w
 55 FOR a=6 TO 16: PRINT AT a,0; PAPER 7; INK 4;"

 8": NEXT a
 57 PRINT AT 21,0; INK 0;"Money:£";mo;"   Bank:£";ba;"  "
 60 FOR a=7 TO 15: PRINT AT a,1; PAPER 7; INK 2;"          ";AT a,9;"          ";AT a
,18;
"          ";AT a,26;"          ": NEXT a
 70 DIM k(4): LET k(1)=INT (RND*13)+1
 71 LET k(2)=INT (RND*13)+1: IF k(2)=k(1) THEN  GO TO 71
 72 LET k(3)=INT (RND*13)+1: IF k(3)=k(2) OR k(3)=k(1) THEN  GO TO 72
 73 LET k(4)=INT (RND*13)+1: IF k(4)=k(3) OR k(4)=k(2) OR k(4)=k(1) THEN  GO TO
 73
100 FOR f=1 TO 4
101 GO SUB 9000
103 IF f=1 THEN  GO TO 109
104 IF hilo=1 AND k(f)>k(f-1) THEN  PRINT AT 18,0; PAPER 6; INK 0; FLASH 1;"Wel
l Done!The card was higher'  ": BEEP .3,40: BEEP .3,36: BEEP .3,37 BEEP .4,41: B
EEP .45,40: LET ba=ba-100: LET mo=mo+100: GO SUB 3000: GO TO 100
105 IF hilo=0 AND k(f)<k(f-1) THEN  PRINT AT 18,0; PAPER 6; INK 0; FLASH 1;"Wel
l Done'The card was lower'   ": BEEP .3,40: BEEP .3,35: BEEP .3,37: BEEP .4,43:
BEEP .45,40: LET ba=ba-100: LET mo=mo+100: GO SUB 3000: GO TO 100
106 IF hilo=1 AND k(f)<k(f-1) THEN  PRINT AT 18,0; INK 1; FLASH 1;"Oh dear!The
card was lower'     ": BEEP .5,0: BEEP .05,3: BEEP .5,1: BEEP .05,2: BEEP .03,10
: BEEP .5,1: BEEP .5,0: LET ba=7000: LET mo=100: GO SUB 3000: GO TO 202
107 IF hilo=0 AND k(f)>k(f-1) THEN  PRINT AT 18,0; INK 1; FLASH 1;"Oh dear'The
card was higher'    ": BEEP .2,0: BEEP .4,3: BEEP .5,1: BEEP .5,2: BEEP .6,1: BE
EP .5,0: LET ba=7000: LET mo=100: GO SUB 3000: GO TO 202
108 BEEP .5,0: PRINT AT 21,0; INK 0;"Money:£";mo;"   Bank:£";ba;"  ": GO SUB
200: IF f=4 THEN  GO TO 155
109 FOR r=1 TO 70
110 PRINT AT 3,7; FLASH 1; PAPER 6; INK 1;"<<<<<<<<<"
115 IF IN 31000<>255 AND NOT IN 31000<191 THEN  FOR a=1 TO 6: BEEP .05,a*10: NE
XT a: LET hilo=1: GO TO 150
120 NEXT r
121 PRINT AT 3,7;"          "
125 FOR r=1 TO 70
130 PRINT AT 3,17; FLASH 1; PAPER 6; INK 1;">>>>>>>>>"
133 IF IN 31000<255 AND NOT IN 31000=191 THEN  FOR a=1 TO 6: BEEP .05,a*5: NEXT
a: LET hilo=0: GO TO 150
135 NEXT r
136 PRINT AT 3,17;"          "
140 GO TO 109
150 PRINT AT 3,7;"                    ": NEXT f
155 GO SUB 200: GO TO 202
200 IF ba<=0 THEN  PAPER 7: BORDER 7: CLS : INK 0: PRINT AT 10,3;"YOU HAVE BROK
EN THE BANK!": STOP
201 RETURN
202 IF mo>100 THEN  LET mo=mo+200
205 GO TO 55
3000 PRINT AT 18,0; INK 7;"Say Anything to continue...     "
3005 IF IN 31000<255 AND NOT IN 31000=191 THEN  GO TO 3010
3006 GO TO 3005
3010 PRINT AT 18,0;"                                ": RETURN
9999 STOP
9000 IF f=1 THEN  LET ka=1
9001 IF f=2 THEN  LET ka=9
9002 IF f=3 THEN  LET ka=18
9003 IF f=4 THEN  LET ka=26
9004 FOR a=6 TO 16: PRINT AT a,ka; PAPER 7;"      ": NEXT a: BEEP .05,6
9005 PAPER 7: INK 2: IF k(f)>1 AND k(f)<11 THEN  PRINT AT 6,ka-1;k(f);"  ";AT 16
,
ka+4;k(f);"  ": IF k(f)=10 THEN  PRINT AT 6,ka-1;"10 ";AT 16,ka+3;"10 "; INK 4;"
  "
"
9006 IF k(f)=1 THEN  PRINT AT 11,ka;"ACE ": GO TO 9020
9007 IF k(f)=2 THEN  PRINT AT 9,ka+1;"  ";AT 13,ka+3;"  ": GO TO 9020
9008 IF k(f)=3 THEN  PRINT AT 8,ka+1;"  ";AT 11,ka+2;"  ";AT 14,ka+3;"  ": GO TO 9
020
9009 IF k(f)=4 THEN  PRINT AT 9,ka;"  ";AT 13,ka;"  ": GO TO 9020
9010 IF k(f)=5 THEN  PRINT AT 9,ka;"  ";AT 11,ka+2;"  ";AT 13,ka;"  ": GO
 TO 9020
9011 IF k(f)=6 THEN  PRINT AT 9,ka;"  ";AT 11,ka;"  ";AT 13,ka;"  ";
GO
 TO 9020
9012 IF k(f)=7 THEN  PRINT AT 9,ka;"  ";AT 11,ka;"  ";AT 13,ka;"  ";
 GO
 TO 9020
9013 IF k(f)=8 THEN  PRINT AT 8,ka;"  ";AT 10,ka;"  ";AT 12,ka;"  ";A
T
14,ka;"  ": GO TO 9020
9014 IF k(f)=9 THEN  PRINT AT 8,ka;"  ";AT 10,ka;"  ";AT 11,ka+2;"  ";AT
12
,ka;"  ";AT 14,ka;"  ": GO TO 9020
9015 IF k(f)=10 THEN  PRINT AT 8,ka;"  ";AT 10,ka+1;"  ";AT 12,ka+1;"  "
;AT
 14,ka;"  ": GO TO 9020
9016 IF k(f)=11 THEN  PRINT AT 11,ka-1;"Jack "
9017 IF k(f)=12 THEN  PRINT AT 11,ka-1;"Queen "
9018 IF k(f)=13 THEN  PRINT AT 11,ka-1;"King "
9020 PAPER 4: RETURN
```

Hi-Lo
How it works

3-4 UDGs
5 set up score variables
10-30 set up screen
35-50 draw Hi and Lo boxes
55-60 Print cards face down
70-73 work out values of four cards
100-108 check if player is still allowed to play
109-140 scan IN 31000 to see if player has spoken
150 finish main loop (F)
200 check to see if you have broken the bank, end game if you have
201 award bonus for correctly guessing four cards
205 start next sequence of four cards
3000-3010 pause routine, waiting for you to speak
900-9020 print card when called

Hi-Lo
Variables

mo how much money you have
ba amount of money in bank
K(1-4) value of each of four cards
hilo did you choose higher or lower? Set at 0 for lower, 1 for higher
Ka position horizontally on screen of card last dealt
v,w,a,f,r,f, FOR...NEXT loops

Quiz
How it works

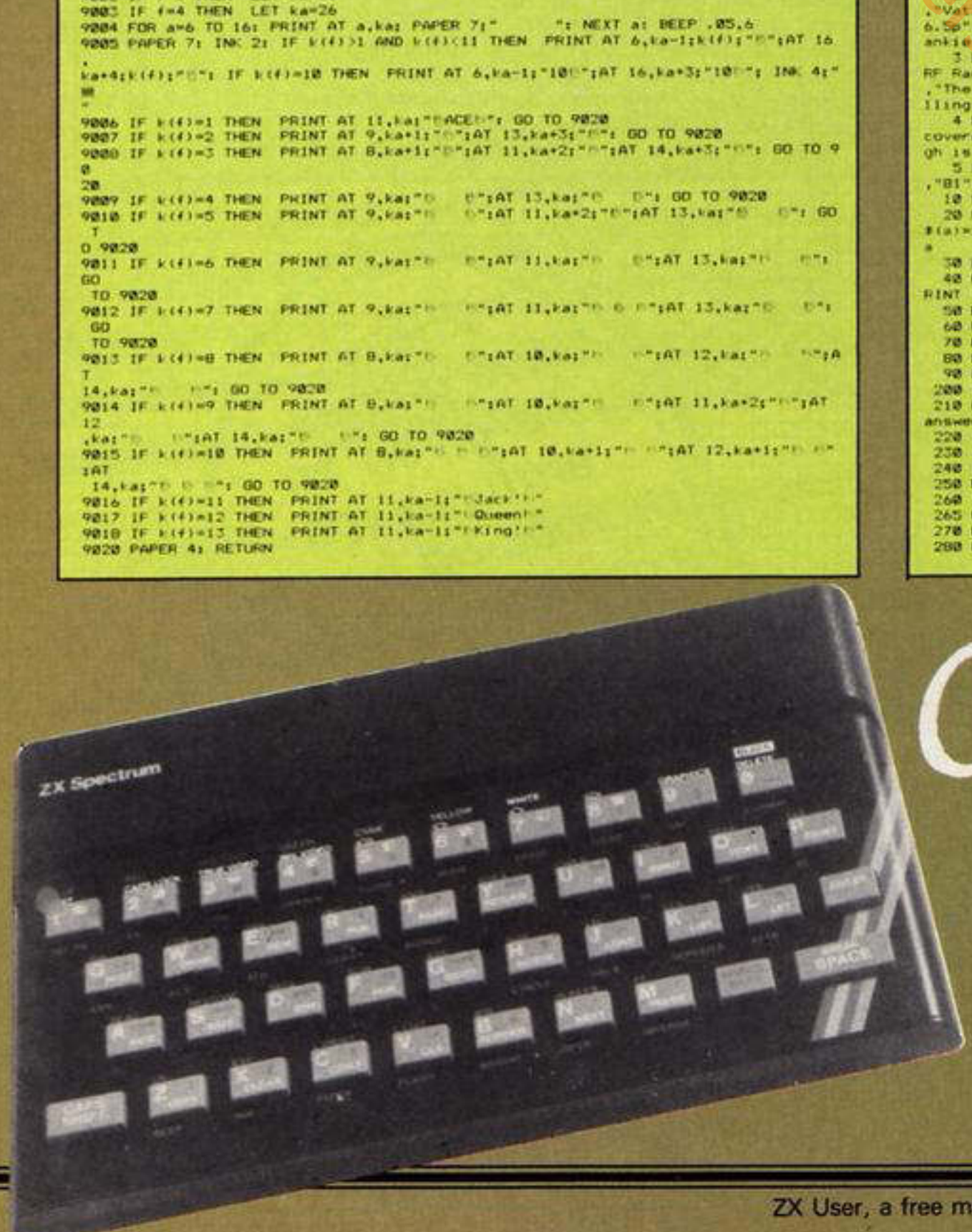
2-4 data for first none questions
5 data for last questions and screen positions of all 10 answers
10 DIMs variables
20 give value to all variables
30 set up screen
40 start main FOR...NEXT loop, print question and three possible answers
50-80 scan for voice at each answer
90 finish main FOR...NEXT loop, print final score
200-280 decide if answer is right or wrong, print appropriate responses

Quiz
Variables

score
q$(10,61) questions
a$ (10,29), b$(10,29), c$(10,29) answers
h(10) screen position of all 10 correct answers
a,b,c,d, FOR...NEXT loops

● **Listing 2 Quiz**

```
  1 REM Quiz by Peter Sweasey
  2 LET score=0: DATA "1:What is the smalllest country in the world?","Andorra"
,"Vatican city","Monaco","2:What is the cost of a first  class stamp?","16p","1
6.5p","17p","3:Who had a number one with  ""The Reflex""?","Culture Club","Fr
ankie Goes To Hollywood","Duran Duran"
  3 DATA "4:What is the world's tallest  tower?","Sears Tower","C.N Tower","WK
RF Radio Tower","5:Which film won the oscar for best motion picture this year?"
,"The Right Stuff","Educating Rita","Terms Of Endearment","6:What is the best se
lling home computer in the world?","ZX Spectrum","ZX81","Vic 20"
  4 DATA "7:How many signs of the zodiac are there?","10","22","12","8:Who dis
covered penicillin?","Alexander Fleming","Robert Koch","Joseph Lister","9:How hi
gh is Everest?","19,285ft (5878m)","29,028ft (8848m)","25,490ft (7772m)"
  5 DATA "10:How many letters are there inthe world's longest word?","94","186"
,"81",6,4,8,6,8,4,8,4,6,4
 10 DIM q$(10,61): DIM a$(10,29): DIM b$(10,29): DIM c$(10,29): DIM h(10)
 20 FOR a=1 TO 10: READ z$: LET q$(a)=z$: READ z$: LET a$(a)=z$: READ z$: LET b
$(a)=z$: READ z$: LET c$(a)=z$: NEXT a: FOR a=1 TO 10: READ z: LET h(a)=z: NEXT
 a
 30 BORDER 0: PAPER 0: CLS : INK 6
 40 FOR b=1 TO 10: CLS : PRINT AT 0,0;q$(b): PRINT AT 4,0;"> ";a$(b): PRINT : P
RINT "> ";b$(b): PRINT : PRINT "> ";c$(b)
 50 FOR c=4 TO 0 STEP 2: PRINT AT c,0; FLASH 1;">"
 60 FOR d=1 TO 100: IF IN 31000<255 AND NOT IN 31000=191 THEN  GO TO 200
 70 NEXT d: PRINT AT c,0;">"
 80 NEXT c: GO TO 50
 90 BEEP .4,0: NEXT b: CLS : PRINT AT 10,8;"FINAL SCORE:";score: STOP
200 IF c=h(b) THEN  GO TO 270
210 PRINT AT 10,0;"WRONG!": BEEP .5,0: BEEP .05,10: PRINT AT 12,0;"The correct
answer was:-"
220 IF h(b)=4 THEN  PRINT a$(b)
230 IF h(b)=6 THEN  PRINT b$(b)
240 IF h(b)=8 THEN  PRINT c$(b)
250 PRINT : PRINT "Your score is:";score: PRINT : PRINT "Speak to continue..."
260 IF IN 31000<255 AND NOT IN 31000=191 THEN  GO TO 90
265 GO TO 260
270 LET score=score+1: PRINT AT 10,0;"CORRECT!": BEEP .5,10: BEEP .05,30
280 GO TO 250
```



Control your Spectrum

Smash Hits

Worse Things Happen at Sea 48K £5.95

Silversoft, London Hse, 271-273 King St, London W6 9LZ

You're a robot, frantically trying to keep a rusty ship afloat. Each compartment has a pump and one patch, may spring several leaks and pump handles are in short supply. Standing in water saps energy and can kill; you can only recharge in compartment one and you have three lives. I'm still on my second crossing but gather you have to steer the ship while cooling the engines.

Water level, course, engine temperature and energy charge are displayed top-screen, plus animation of the ship sailing between two cliffs and a floor plan, showing your position and the water rising. Scoring: cargo value minus spoilage, repairs and replacement robots.

Super graphics, good sound. This game inspires panic and is physically exhausting. It needs a cool head and cunning rather than fast fingers. Very addictive. No inlay instructions were supplied with pre-production tape but they are necessary because on-screen ones are cryptic. It finally dawned on me there were other doors apart from the obvious ones.

This is the best game I've seen for ages. **D.C.**

instructions	60%
playability	90%
graphics	90%
value for money	90%

★ ★ ★ ★

Front Loop 48K £3

NTD Software, PO Box 543, London SW6 5DS

The screen displays a number of compartments, each containing apples. Fighting a time limit, you have to collect all the apples before moving to the next screen. The apples sometimes turn rotten and you must find a blue banana to restore them.

On later levels, some compartments have no entrances and blasting in penalises your time allowance. Guards appear in increasing numbers, and contact with them, rotten apples or blue squares loses one of five lives. Press the panic button to freeze guards — but pay the penalty. There are 10 screens, six skill levels, and you can use keyboard or joystick.

Graphics aren't great — both you and the guards are bodyless forward-facing heads. Inlay instructions are clear, which is useful because on-screen instructions take no for an answer without any keypress. The game doesn't crash but plays improperly with Microdrive attached. The panic button seems to move guards elsewhere rather than freeze them and it's unfair that they materialise at your start point.

I found the game amusing but suspect 60 screens of apples may produce colic. **D.C.**

instructions	80%
playability	60%
graphics	55%
value for money	90%

★ ★ ★

Empires 48K £19.95

Imperial Software, Imperial Hse, 153, Churchill Rd, Poole, Dorset

This is a three-player strategy game in which feature a human, reptile and robot. You must dispatch scouts, mine planets, build cargo and military ships, ally or fight other races. Your aim is to rule the galaxy. The package contains umpire tape, three individual player tapes, four instruction sets, four galaxy grid maps and data cards.

Each galactic year, players enter moves on their programs, saving and sending data to the umpire, who processes and returns updated data for next move. Excellent text and graphics although human blue gives difficulty.

An absorbing game which I yearn to play, but the loading time is seven minutes. The program news at the end of each move and sometimes on loading, forces another seven-minute wait. **D.C.**

instructions	90%
ease of use	50%
display	90%
value for money	75%

★ ★ ★ ★

Sabre Wulf £9.95

Ultimate Play the Game, The Green, Ashby de la Zouch, Leics LE6 5JU

When a firm like Ultimate Play the Game is quiet for a while something is in the wind. The ads excite interest by simply giving the name and price of the game.

Sabre Wulf is very similar to Atic Atac but more playable and makes better use of colour. Previous Ultimate games had virtually no instructions: this comes in a larger than average box with a glossy sheet explaining a little in riddle form.

You are an intrepid explorer lost in a jungle. Your task (I think) is to find the four sections of a talisman and defeat the wulf. The game has arcade and adventure elements. You kill spiders, snakes and other nasties using a sword. You chase and are chased by rhinos and wildebeest. The adventure is in mapping the jungle (a job for two) and finding out the effect on you of secretions from local fauna. Yellow flowers stun you, red make you invincible, purple reverse your movements and so on. These effects are not permanent.

The action is fast, smooth and detailed to a cartoon standard. The price reflects demand and the work involved. This is a must for all Spectrum games players.

instructions	100%
playability	100%
graphics	100%
value for money	100%

★ ★ ★ ★ ★

Crazy Crane £5.50

Voyger Software, Unit 31, Wirral Business Centre, Corsey Lane, Birkenhead, Merseyside

The scene is a harbour in shark-invested waters. You are a computerised magnetic crane on the pier, a short distance from the harbour wall. Ships appear from behind the wall and pass the pier. Before they reach you, you must extend your magnetic grab, lift off their cargo and land it safely on the pier. If missed, the cargo falls into the water at the pier, losing one of three lives. Unload five lots of cargo and you reach the next level. The cargo alternates between helicopters, containers, animal cages and aliens. Ships come fast and furious, sometimes in pairs.

After level four, missiles attack. What happens thereafter is a mystery to me. Your score is based on cargo unloaded plus 100 times level bonus and presumably number of levels is limited only by ability to survive. Control is by keyboard and Kempston-type joystick.

Good graphics, but Microdrives are not new so there's no excuse for program-crash because they are attached. Crazy Crane required intense concentration and fast fingers. One for the 'my score is highest' brigade. I might tire of lack of variety in screen and cargo after a while but it's enjoyable for now. **D.C.**

instructions	90%
playability	75%
graphics	85%
value for money	90%

★ ★ ★ ★

These five games are given an enthusiastic thumbs-up by our experienced panel of reviewers. Read on to find why they're so popular